

**Kovács Henriette**

# **OKTATÁSI ÓRATERVEZETEK**

ÓRA IDŐKERETE, MENETE, CÉLOK, DIDAKTIKAI FELADATOK,  
NEVELÉSI-OKTATÁSI STRATÉGIA  
(MÓDSZEREK, TANULÓI MUNKAFORMÁK, ESZKÖZÖK)



Kiadó:  
Magyar Kultúra Emlékívek Kiadó

Főszerkesztő:  
Kollár Ferenc

**Kovács Henriette**  
**OKTATÁSI ÓRATERVEZETEK**  
ÓRA IDŐKERETE, MENETE, CÉLOK, DIDAKTIKAI FELADATOK,  
NEVELÉSI-OKTATÁSI STRATÉGIA  
(MÓDSZEREK, TANULÓI MUNKAFORMÁK, ESZKÖZÖK)

ISBN:  
978-615-5746-30-7

Minden jog fenntartva.  
A kiadvány egyetlen részlete sem sokszorosítható  
semmiféle formában a kiadó engedélye nélkül.

## TARTALOMJEGYZÉK

Bevezetés .....	7
Óratervezet - Miért hasznosak a tömbök? .....	9
Az óra cél- és feladatrendszere.....	10
Fejlesztendő készségek, képességek .....	10
Fejlesztendő attitűd .....	11
Az óra didaktikai feladatai .....	12
Az óra menete.....	14
Feladatok.....	30
Reflexió.....	31
Óratervezet - A String osztály és a tömbök. Túlindexelés. ....	32
Az óra cél- és feladatrendszere.....	33
Fejlesztendő készségek, képességek .....	33
Fejlesztendő attitűd .....	34
Az óra didaktikai feladatai .....	34
Az óra menete.....	36
Feladatok.....	46
Reflexió.....	47
Óratervezet - Tömb elemeinek alapértelmezett értékei. Tömb feltöltése.....	48
Az óra cél- és feladatrendszere.....	49
Fejlesztendő készségek, képességek .....	50
Fejlesztendő attitűd .....	50
Az óra didaktikai feladatai .....	51
Az óra menete.....	53

Feladatok.....	63
Reflexió.....	64
Óratervezet - Tömb elemeinek feltöltése billentyűzetről, kiírása vesszővel elválasztva. ....	66
Az óra cél- és feladatrendszere.....	67
Fejlesztendő készségek, képességek .....	67
Fejlesztendő attitűd .....	67
Az óra didaktikai feladatai .....	68
Az óra menete.....	70
Feladatok.....	81
Reflexió.....	82
Óratervezet - Tömb elemeinek kiírása vesszővel elválasztva. Inicializáló blokk. ....	83
Az óra cél- és feladatrendszere.....	84
Fejlesztendő készségek, képességek .....	85
Fejlesztendő attitűd .....	85
Az óra didaktikai feladatai .....	86
Az óra menete.....	88
Feladatok.....	100
Reflexió.....	101
Óratervezet - Véletlenszámok generálása. Tömb elemeinek feltöltése véletlenszámokkal. ....	102
Az óra cél- és feladatrendszere.....	103
Fejlesztendő készségek, képességek .....	104
Fejlesztendő attitűd .....	104
Az óra didaktikai feladatai .....	105
Az óra menete.....	107
Feladatok.....	121
Házi feladat .....	122
Reflexió.....	124

Óratervezet - Alapvető adatgyűjtési és feldolgozási algoritmusok.....	126
Az óra cél- és feladatrendszere.....	127
Fejlesztendő készségek, képességek .....	127
Fejlesztendő attitűd .....	128
Az óra didaktikai feladatai .....	129
Az óra menete.....	131
Feladatok.....	143
Reflexió.....	144
Óratervezet - Maximumkiválasztás. Tömbök elemein végzett műveletek.....	146
Az óra cél- és feladatrendszere.....	147
Fejlesztendő készségek, képességek .....	147
Fejlesztendő attitűd .....	148
Az óra didaktikai feladatai .....	149
Az óra menete.....	151
Feladatok.....	165
Reflexió.....	166
Óratervezet - Összetett elemtípusú tömbök. Tömbelemek cseréje. ....	168
Az óra cél- és feladatrendszere.....	169
Fejlesztendő készségek, képességek .....	170
Fejlesztendő attitűd .....	170
Az óra didaktikai feladatai .....	171
Az óra menete.....	173
Feladatok.....	191
Reflexió.....	192
Óratervezet - Tömb elemeinek megfordítása. Gyakorlás. ....	193
Az óra cél- és feladatrendszere.....	194
Fejlesztendő készségek, képességek .....	194

Fejlesztendő attitűd .....	194
Az óra didaktikai feladatai .....	196
Az óra menete.....	198
Feladat.....	208
Reflexió.....	215
Foglalkozás tervezete – Az ismétlés - iteráció. Ciklusok fajtái, átalakíthatóságuk, alkalmazásuk. ....	216
A foglalkozás cél- és feladatrendszere .....	217
Fejlesztendő készségek, képességek .....	217
Fejlesztendő attitűd .....	218
Az óra didaktikai feladatai .....	219
A foglalkozás menete .....	221
Feladatok.....	236

## BEVEZETÉS

*A tanár szerepe kulcsfontosságú a pedagógiai, oktatási folyamatban, mert a tanuló fejlődését közvetlenül befolyásolja. A tanári munka során fejleszteni kell azokat a kompetenciákat, amelyek által a tanulók élet- és munkaképessé válnak, valamint nagyfokú támogatást kell nyújtani a tehetséges, a hátrányos helyzetű és a fejlődési zavarokkal küzdő tanulóknak, hogy képességeikhez mérténen valósíthassák meg céljaikat.*

*Nagyon fontos a szakmai fejlődésért az elkötelezettség és szakmai felelősségvállalás. Munkánk során elengedhetetlen a szaktárgyi tudás naprakészen tartása. Az informatika területén pedig különösen fontos az önképzés és a folyamatos továbbképzés.*

*A szakmai tartalom mellett a módszertani megújulás is nagyon lényeges. Az informatikus hallgatók szinte egész életükben a véget nem érő (tovább)tanulásra lesznek kényszerítve, így fontos, hogy mind a tanulási, mind pedig az értékelési folyamatokban már olyan eszközöket és módszereket alkalmazzunk, amelyek motiválják őket, felkeltik érdeklődésüket.*

*A pedagógiai munka releváns része a pedagógiai eszköztár folyamatos fejlesztése és megújítása is. A generációváltás, a megváltozott világ mindig új kihívások elé állítja a pedagógust. A társadalmi változások arra kényszerítenek bennünket, hogy próbáljunk meg olyan új módszereket és eszközöket alkalmazni, amelyek*

*a tanított generációknak segítik a tananyag befogadását. A gyorsan változó informatika területén ez még fontosabb, mint más területeken. A pedagógiai eszköztár fejlesztése elsősorban a tanulói visszajelzések, visszacsatolások alapján történik. Általános szabályok nincsenek egy-egy új módszer, nevelési elv használatára, hiszen ami az egyik tanulócsoportban működik, az még nem garantálja, hogy egy másikban is működőképes lesz.*

*A tanári tervező munka legkisebb egysége az órák, foglalkozások tervezete, mely tartalmazza az óra/foglalkozás általános adatait, tananyagát, cél- és feladatrendszerét, a fejlesztendő készségeket, képességeket, a fejlesztendő attitűdöt, az óra didaktikai feladatait, a tantárgyi kapcsolatokat, a felhasznált forrásokat; egy táblázatban az óra/foglalkozás időkeretét, menetét, célját, didaktikai feladatait, a didaktikai feladatok megvalósításához a nevelési-oktatási stratégiát, azaz az alkalmazott módszereket, tanulói munkaformákat, eszközöket; a táblaképet, a feladatokat és az első tíz óra esetén az órához tartozó reflexiókat. Ez a tanár személyes tervezete, mely különböző formákat ölthet.*

*A könyvben található tizenegy tervezet formailag egységes, melyek megalkotásakor az vezérelt, hogy az órák/foglalkozások menete átlátható, egyértelmű, könnyen követhető, letisztult, ezáltal könnyen tanítható legyen. Ennek eléréséhez a vizualitást hívtam segítségül.*

*A táblázatban az óra/foglalkozás menete szélesebb oszlopba került, hiszen itt található az oktatott tananyag, a konkrét feladatok, a megoldások, a feltett kérdések és a várható válaszok. Abban az esetben, ha nem tervezetet, hanem vázlatot írunk, akkor nem kell feltüntetni a várható tanulói válaszokat. A célok, didaktikai feladatok külön oszlopban, az óra menete mellett találhatók, a nevelési-oktatási stratégiát pedig, azaz az oktatási módszereket, a tanulói munkaformákat és az eszközöket három oszlop helyett egy oszlopban helyeztem el, megkülönböztetésükre halvány háttérszínezést alkalmaztam. Az oktatási módszereket a piros, a tanulói munkaformákat a kék, az eszközöket a sárga szín jelöli. A megjegyzéseket sem külön oszlopba, hanem a nevelési-oktatási stratégia oszlopába helyeztem el, háttérszín nélkül.*

*Tíz tervezet óratervezet, melyek tartalmilag összefüggnek, az első hat óratervezet egy tanítási napon hat tanítási órán, a többi négy*

*szintén egy napon, négy tanítási órán valósult meg. A tizenegyedik tervezet egy foglalkozás tervezete, hiszen a szakképzésről szóló 2019. évi LXXX. törvény értelmében a szakképzésben nem tanórákat, hanem foglalkozásokat tartunk.*

*Az egész életen át tartó tanulással Life Long Learning (LLL) folyamatosan képezzük magunkat, korszerűsítsük és fejlesszük pedagógiai és szakmai tudásunkat, így személyes példamutatással tudjuk alakítani a tanulók értékrendjét, pozitív tulajdonságait, a tanuláshoz és társaikhoz, valamint a későbbiekben a munkához és a társadalomhoz való hozzáállásukat.*

**Kovács Henriette**  
**szakmai és köznevelési szakértő**

*Budapest, 2022.*

## ÓRATERVEZET - MIÉRT HASZNOSAK A TÖMBÖK?

**A pedagógus neve:** Kovács Henriette

**A szakmacsoport száma és megnevezése:** 7. Informatika

**A szakképesítés azonosító száma:** 54 213 05

**A szakképesítés megnevezése:** Szoftverfejlesztő

**Iskolai rendszerű szakképzés** (a szakképzési évfolyamok száma: 2)

**Szakmai követelménymodul:** 10817-12 Hálózatok, programozás és adatbázis-kezelés

**Tantárgy:** Adatbázis- és szoftverfejlesztés gyakorlat

**Programozási nyelv:** Java

**Fejlesztői környezet:** NetBeans IDE 7.3

**Osztály - csoport:** I/13. S - SZ2

**Az óra tananyaga:** Miért hasznosak a tömbök?

## Az óra cél- és feladatrendszere

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonómiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt; *rávezetni a tanulókat a tömb adatszerkezet szükségességére, megismertetni és létrehozni ezt a gyors és hatékony listát, a tömböt*, mindezzel hozzájárulva a tanulók logikus gondolkodásának, problémamegoldásának, programozói szemléletének kialakításához. Kialakítani a tanulóknál az informatikai ismereteik folyamatos bővítésének és megújításának igényét.
- Korszerű alkalmazói készség kialakítása: meglévő, - Internet segítségével - új alkalmazások kiválasztása a feladat tervezéséhez (Microsoft Office Excel, Structorizer), legyenek képesek arra, hogy mindezeket célszerűen használják.

- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás: a NetBeans fejlesztői környezet, a felhasználó tevékenységére azonnal reagál, így lehetőséget teremt az egyéni ütemű tanulásra, a pontos, kitartó, fegyelmezett munkára, a lemaradókkal való különleges foglalkozásra.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, tömböt létrehozni, és elemeit kiírni.*
- *Tudjon tömb elemeit tabulátorokkal elválasztva egymás mellett megjeleníteni.*
- *Legyen képes a tömbök elemeit elérni.*

## Fejlesztendő készségek, képességek

- Absztrakt gondolkodás, algoritmikus gondolkodás, alkalmazásválasztás, alkalmazói képesség, alkotó képesség, diagnosztizálás,

döntéskéesség, eszközválasztás, hallott szakmai szöveg értése, hibakeresés, logikus gondolkodás, megfigyelés, meggyőzőképesség, olvasott szakmai szöveg megértése angol nyelven, önállóság, önellenőrzés, precizitás, pontosság, problémamegoldás, programkód írása, programválasztás, struktogram elemzése, struktogram készítése, szakmai nyelvű beszédkésztség, szintaktika követése, visszalemlékezés.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait. Tudjon adatokat megkeresni, elérni a hálózati szolgáltatások alkalmazásával.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Legyen képes elemi és összetett adatok megkülönböztetésére, kezelésére, használatára.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.

- Legyen képes alkalmazást (szoftvert) futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanuló társa is fel tudja használni.
- Legyen képes eligazodni és tudjon keresni a Java hivatalos dokumentációjában.

### **Az óra didaktikai feladatai**

- A figyelem felkeltése
- A jelenség elemzése
- A megoldás sokoldalú elemzése
- A tanulók informálása a tanulási célról
- A tények elemzése
- Aktivizálás
- Alkalmazás
- Általánosítás
- Az új ismerteket megalapozó tények biztosítása
- Ellenőrzés
- Értékelés
- Figyelem felkeltése
- Fogalomalkotás
- Absztrakció
- Általánosítás
- Fokozatosság
- Motiváció
- Szemléletesség
- Ismétlés
- Visszacsatolás
- Új ismeret közlése
- Tudományosság
- Megerősítés
- Részösszefoglalás
- Rendszerezés
- Információ gyűjtése
- Gyakorlás
- Következtetés
- Problémafelvetés
- Rögzítés

**Tantárgyi kapcsolatok:** Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek

**Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):**

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyiné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

<http://docs.oracle.com/javase/7/docs/api/>


Dátum: éééé. hh. nn.

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 2'	<p>Számítógép bekapcsolása, házi feladat megnyitása</p> <pre> public class EgyValtozonev {      public static void main(String[] args) {         /*          HÁZI feladat: Tárolja el egy öttagú család tagjainak életkorát!          Számítsa ki a család átlagéletkorát és a monitorra írassa ki          két tizedesjegyre kerekítve!         */         int imre = 82;         int sara = 79;         int zsofi = 42;         int zoltan = 51;         int zsolt = 33;          double atlagEletkor = (imre + sara + zsofi + zoltan + zsolt) / 5.0;         System.out.println("A család átlagéletkora: "             + Math rint(atlagEletkor * 100) / 100);     } } </pre> <p>Képzeljék el, hogy a házi feladatban nem egy öttagú család tagjai, hanem egy 18 lakásos társasház lakói szerepelnek. Mi okozná a nehézséget?</p> <p><i>Válasz: A nehézséget az okozná, hogy sok változót kellene deklarálni.</i></p>	<p>Ellenőrzés Önellőrzés Diagnosztizálás Hibakeresés</p> <p>Precizitás, pontosság Szintaktika követése</p> <p>A figyelem felkeltése Motiváció</p>	<p>Számítógép Házi feladat</p> <p>A házi feladatok elektronikus úton, határidőre történő beadása; reflektálás részben elektronikusan, részben órán.</p> <p>Számítógép Projektor Vetítővászon</p> <p>Frontális munka</p> <p>Megbeszélés (beszélgetés)</p> <p>NetBeans IDE 7.3 Integrált Fejlesztői Környezet használata</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
2 – 3'	<p>Hogy tudnánk megkönnyíteni a dolgunkat?</p> <p>Jó lenne, ha lenne olyan adatszerkezet, mely egy adott változónévvel több különálló adatot tud tárolni a megadott típusból.</p> <p>Pl.: Képzeljék el, hogy egyforma dobozokat rakunk le egymás mellé és a dobozokba beletesszük a számokat.</p>  <p>Mindegy, hogy egész számokat vagy valós számokat helyezünk el a dobozokban?</p> <p><i>Válasz: Nem mindegy, hogy egész számokat vagy valós számokat teszünk a dobozokba.</i></p>	<p>Problémamegoldás</p> <p>Szemléletesség</p> <p>Logikus gondolkodás</p> <p>Fokozatosság</p> <p>Absztrakt gondolkodás</p> <p>Az új ismerteket megalapozó tények biztosítása</p>	<p>Szemléltetés</p> <p>Projektor</p> <p>Saját készítésű kép CINEMA 4D programmal</p> <p>Kooperatív oktatási módszer</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
3 – 4'	<p>Mit szeretnénk még elhelyezni benne?</p> <p><i>Válasz: Szeretnénk még elhelyezni benne pl. szöveget, logikai értéket.</i></p> <p>Ezen igényeink megvalósítására Java-ban a tömb alkalmas.</p>	<p>Absztrakt gondolkodás</p> <p>Új ismeret közlése</p>	<p>Frontális munka</p> <p>Elbeszélés</p>
4 – 7'	<p>Tömb: Fix elemszámú, azonos típusú elemekből álló összetett adatszerkezet.</p> <p>Használatában az a nagyszerű, hogy nem kell az elemeknek egyedi neveket kitalálni, hanem egy gyűjtőnevet adunk az egész csoportnak.</p> <p>Hogyan deklarálunk egy változót? Mondjon egy példát!</p> <p><i>Válasz: Úgy deklarálunk egy változót, hogy megadjuk a típusát és a nevét!</i></p> <p>Pl.: <code>int a;</code></p> <p>A tömb referencia típusú változó, melyet szintén <b>deklarálnunk</b> kell.</p> <ol style="list-style-type: none"> <li>1. elemek típusa</li> <li>2. tömbképző operátor: szögletes zárójelpár</li> <li>3. tömb azonosítója</li> </ol> <p>↙ ide is tehetjük!</p> <p><code>&lt;elemtípus&gt;[] &lt;tömbAzonosító&gt;;</code></p> <p><code>&lt;elemtípus&gt; &lt;tömbAzonosító&gt;[];</code></p> <p>Pl.: <code>int[] iArray;</code></p>	<p>Fogalomalkotás, absztrakció, általánosítás</p> <p>Rögzítés</p> <p>Visszaemlékezés Visszacsatolás Rögzítés</p> <p>Alkalmazás Fogalomalkotás</p> <p>Megfigyelés Rögzítés</p>	<p>Elbeszélés</p> <p>Magyarázat</p> <p>Megbeszélés</p> <p>Tábla</p> <p>Projektor</p> <p>Tábla</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
7 – 9'	<p>Pl.: <code>int[] iArray1;</code>  <code>int iArray2[];</code></p> <p>Hogyan tudunk egyszerre több változót deklarálni?</p> <p><i>Válasz: Úgy tudunk egyszerre több változó deklarálni, hogy megadom a típust és vesszővel elválasztva a változók neveit.</i></p> <p>Feladat: Döntse el, melyik tömb a következők közül?</p> <p><code>int[] iArray1, iArray2;</code>  <code>int iArray3[], iArray4;</code></p>	<p>Megerősítés</p> <p>Visszaemlékezés Visszacsatolás</p> <p>Döntésképeség Aktivizálás Gyakorlás Alkalmazás</p>	<p>Számítógép Projektor Vetítővászon</p> <p>Frontális munka</p> <p>Kooperatív oktatási módszer</p> <p>Önálló munka</p>
9 – 10'	<p>Emlékezzünk vissza, mi történik a memóriában, ha deklarálunk egy primitív típusú változót!</p> <p><i>Válasz: A fordító az 'a' nevet egy meghatározott tárolóterülethez rendeli.</i></p> <p>a <span style="border: 1px solid black; display: inline-block; width: 50px; height: 20px; vertical-align: middle;"></span></p> <p>Nézzük, mi történik a tömböknél!</p> <p>Milyen adatszerkezet a tömb?</p> <p><i>Válasz: A tömb összetett adatszerkezet.</i></p>	<p>Visszaemlékezés Ismétlés</p> <p>Megerősítés</p>	<p>Frontális munka</p> <p>Megbeszélés</p> <p>Tábla</p> <p>Szemléltetés</p>


IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
10 – 12'	<p>Az így deklarált változó <u>képes</u> egy megadott elemtípusú tömbre <u>mutatni</u>. A változóban az objektum memóriabeli címét tároljuk.</p> <p>Deklaráció hatására nem jön létre objektum!</p> <p>Nézzük, mi történik a memóriában, amikor az iArray nevű int típusú elemekből álló tömböt deklaráljuk!</p> <div style="text-align: center;">  </div> <p>A deklarálással csak a referenciának foglalunk helyet, ezen a ponton még nincs tömb!</p>	<div>Hallott szakmai szöveg értéke</div> <div>Új ismeret közlése Tudományosság</div> <div>Szemléletesség</div> <div>Megerősítés</div>	<div>Elbeszélés</div> <div>Szemléltetés</div> <div>Tábla</div>
12 – 13'	<p>A fejlesztői környezetben a Gyakorlat13 nevű projektben hozzon létre egy gyakorlat13 csomagot és ebben egy Gyakorlat13_1 nevű osztályt!</p>	<div>Önállóság Eszközválasztás Aktivizálás</div>	<div>NetBeans IDE 7.3</div> <div>Egyéni munka</div>
13 – 14'	<p>Feladat:</p> <p>Deklaráljon egy iArray nevű, int típusú elemekből álló tömböt, a kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben!</p> <p>Figyeljen a kód strukturáltságára!</p>	<div>Alkotó képesség Programkód írás</div>	<div>Egyéni munka</div>

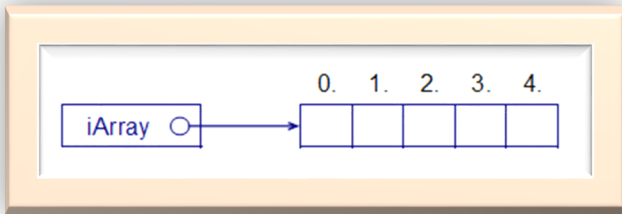
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
14 – 15'	<p>A kivetített kódrészlet alapján ellenőrizze munkáját!</p> <pre> public class Gyakorlat13_0 {      public static void main(String[] args) {          // 1. feladat          // tömb deklarálása         int[] iArray; </pre>	<p>Szintaktika követése Precizitás, pontosság Önellenőrzés</p>	<p>Szemléltetés</p> <p>Projektor</p>
15 – 17'	<p>Melyik operátorral hozunk létre objektumokat? (Gondoljon pl. a billentyűzet-ről való bevitelre!)</p> <p><i>Válasz: A new operátorral hozunk létre objektumokat.</i></p> <p>A tömböt külön <b>létre</b> kell <b>hoznunk!</b></p> <p>Mit gondolnak, melyik operátorral tudunk tömböt létrehozni?</p> <p><i>Válasz: A new operátorral tudunk tömböt létrehozni.</i></p> <p>A tömböt a new operátorral hozhatjuk létre a program futása közben.</p> <ol style="list-style-type: none"> <li>1. new operátor</li> <li>2. elemek típusa</li> <li>3. tömb mérete (elemek száma): nem negatív, int típusú kifejezés.</li> </ol> <pre> new &lt;elemtípus&gt;[&lt;méret&gt;]; </pre>	<p>Visszacsatolás</p> <p>Logikus gondolkodás Fokozatosság Alkalmazói képesség</p> <p>Új ismeret közlése</p>	<p>Megbeszélés Elbeszélés</p> <p>Frontális munka</p> <p>Projektor</p>

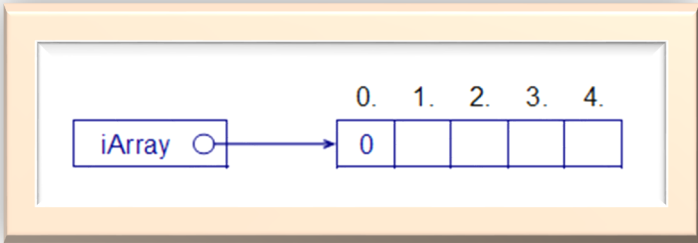
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
17 – 18'	<p>A tömb méretét mibe kell tenni?</p> <p><i>Válasz: A tömb méretét szögletes zárójelbe kell tenni.</i></p> <p>Feladat: Hogyan tud létrehozni egy 5 elemű, int típusú tömböt? Fel tudja használni az előbbi deklarációt? Mindenki a fejlesztői környezetben próbálkozzon, ha hibát jelez a NetBeans IDE, próbálja megtalálni a hibát! Ha nem megy, segítek. A kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben!</p>	<p>Megfigyelés</p> <p>Alkalmazás Hibakeresés</p>	<p>Önálló munka</p> <p>Fejlesztői környezet</p>
18 – 19'	<p>A kivetített kódrészlet alapján ellenőrizze munkáját!</p> <pre>// tömb létrehozása iArray = new int[5];</pre>	<p>Szintaktika követése Precizitás, pontosság Önellenőrzés</p>	<p>Szemléltetés</p> <p>Projektor NetBeans IDE 7.3</p>
19 – 20'	<p>Nézzük meg, mi történik a memóriában az értékadás során, primitív és összetett típusú változó esetén!</p> <pre>int a = 13;  int[] iArray = new int[5];</pre>	<p>A tanulók informálása a tanulási célról</p> <p>Visszacsatolás</p>	<p>Tábla</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
20 – 22'	<p>Próbálja megfogalmazni a különbséget!</p> <pre>int a = 13;</pre> <div data-bbox="577 480 983 756" data-label="Diagram"> </div> <p>Láthatjuk, hogy az érték (13) közvetlenül a változó tárolóterületére íródik be.</p> <pre>int[] iArray = new int[5];</pre> <div data-bbox="479 983 1079 1171" data-label="Diagram"> </div> <p>A tömb adatai külön tárolóterületen helyezkednek el.</p>	<div data-bbox="1332 316 1637 521"> Szakmai nyelvű beszédkésztség  Aktivizálás  Szemléletesség  Visszacsatolás </div> <div data-bbox="1332 590 1637 746"> Az új ismereteket megalapozó tények biztosítása  A tények elemzése </div> <div data-bbox="1332 815 1637 861"> Általánosítás </div> <div data-bbox="1332 930 1637 976"> Tudományosság </div> <div data-bbox="1332 1045 1637 1091"> Rendszerezés </div>	<div data-bbox="1711 367 2089 456"> Megbeszélés  Kooperatív oktatási módszer </div> <div data-bbox="1711 523 2089 569"> Tábla </div> <div data-bbox="1711 687 2089 734"> Frontális munka </div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK						
22 – 25'	<p>Hány eleme lehet egy tömbnek?</p> <p>Maximum <code>Integer.MAX_VALUE</code> eleme lehet a tömbnek.</p> <p>Nézzünk utána a Java hivatalos dokumentációjában, hogy mit is jelent ez?</p> <p>Hol találom meg a dokumentációt?</p> <p><i>Válasz: A dokumentációt a böngészőben találom meg.</i></p> <p>Mit írjunk be?</p> <p><i>Válasz: Integer Java szavakat írjuk be.</i></p> <p>Melyik oldalt keresem?</p> <p><i>Válasz: Azt az oldalt keresem, amelyben bent van a docs, oracle szavak és a 7.</i></p> <p><a href="http://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html">http://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html</a></p> <div><div>Field Summary</div><div>Fields</div><table><thead><tr><th>Modifier and Type</th><th>Field and Description</th></tr></thead><tbody><tr><td>static int</td><td><code>MAX_VALUE</code> A constant holding the maximum value an <code>int</code> can have, <math>2^{31}-1</math>.</td></tr><tr><td>static int</td><td><code>MIN_VALUE</code> A constant holding the minimum value an <code>int</code> can have, <math>-2^{31}</math>.</td></tr></tbody></table></div>	Modifier and Type	Field and Description	static int	<code>MAX_VALUE</code> A constant holding the maximum value an <code>int</code> can have, $2^{31}-1$ .	static int	<code>MIN_VALUE</code> A constant holding the minimum value an <code>int</code> can have, $-2^{31}$ .	<p>Figyelem felkeltése</p> <p>Programválasztás</p> <p>Információ gyűjtése</p> <p>Olvasott szakmai szöveg megértése angol nyelven</p>	<p>Irányított, önálló munka</p> <p>Számítógép Internet</p> <p>Kooperatív oktatási módszer</p> <p><a href="http://docs.oracle.com/javase/7/docs/api/">http://docs.oracle.com/javase/7/docs/api/</a></p>
Modifier and Type	Field and Description								
static int	<code>MAX_VALUE</code> A constant holding the maximum value an <code>int</code> can have, $2^{31}-1$ .								
static int	<code>MIN_VALUE</code> A constant holding the minimum value an <code>int</code> can have, $-2^{31}$ .								

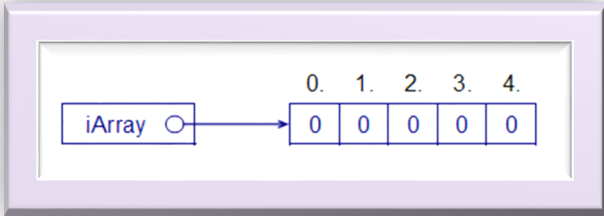
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
25 – 26'	<p>Mivel tudnám kiszámolni ezt az értéket?</p> <p><i>Válasz: Számítógép számológépével.</i></p> 	Alkalmazásválasztás	<div>Irányított, önálló munka</div> <div>Számológép</div>
26 – 28	<p>Tehát:</p> <p>A tömb elemei külön tárolóterületen helyezkednek el. Láthattuk, hogy milliárdos nagyságrendű lehet a tömb elemeinek a száma.</p> <p>Vajon vannak-e a létrehozott iArray tömbnek értékei? Mi nem töltöttük fel értékekkel, a kérdés tehát az, hogy vesznek-e fel alapértelmezett értékeket?</p> <p>Hogy tudnánk meggyőződni róla?</p> <p><i>Válasz: Ha látjuk, hogy vannak benne értékek.</i></p> <p>Hogyan láthatjuk?</p> <p><i>Válasz: Ha ki tudjuk írni az értékeket, akkor biztosak lehetünk benne, hogy vannak benne értékek.</i></p>	<div>Megerősítés Részösszefoglalás</div> <div>Logikus gondolkodás</div> <div>A figyelem felkeltése Problémafelvetés</div>	<div>Elbeszélés Megbeszélés</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
28 – 31'	<p>Ahhoz, hogy ki tudjuk írni a tömb elemeit, el kell őket érni. A tömb egyes elemeire indexekkel hivatkozunk.</p>  <p>Az elemek tehát indexelhetők, de vigyázzunk, mert az index minden esetben 0-ról induló egész szám.</p> <p>A tömbképző operátor nemcsak a tömb létrehozásakor, hanem a tömb egyes elemeinek elérésekor is használatos.</p> <p>Mit gondolnak, hogy hivatkozunk az iArray tömb első elemére? Ügyeljen az indexre!</p> <p><i>Válasz: iArray[0]</i></p> <p>Feladat: Írassa ki az iArray tömb első elemét a fejlesztői környezetben! A kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben!</p> <p>Ha hibát jelez a NetBeans IDE, próbálja megtalálni a hibát!</p>	<p>Az új ismereteket megalapozó tények biztosítása Fogalomalkotás</p> <p>Szemléletesség</p> <p>Új ismeret közlése Figyelem felkeltése</p> <p>Fokozatosság</p> <p>Aktivizálás Figyelem felkeltése</p> <p>Alkalmazás Precizitás, pontosság Diagnosztizálás</p>	<p>Elbeszélés</p> <p>Szemléltetés</p> <p>Frontális munka</p> <p>Tábla</p> <p>Fejlesztői környezet</p> <p>Önálló munka</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
31 – 32'	<p>A kivetített kódrészlet alapján ellenőrizze munkáját! Mit tapasztal a futtatás során?</p> <pre data-bbox="320 459 1216 585">// tömb első elemének kiírása System.out.println("Az iArray tömb első eleme: " + iArray[0]);</pre> <p><i>Válasz: A tömb első eleme alapértelmezett értéket vett fel, ez a nulla.</i></p>  <p>Mire következtetnek ebből?</p> <p><i>Válasz: A tömb többi eleme is az alapértelmezés szerinti nulla értéket veszi fel.</i></p>	<p>Szintaktika követése Precizitás, pontosság Önellenőrzés</p> <p>Megfigyelés</p> <p>Információt gyűjt A jelenség elemzése</p> <p>Következtetés</p> <p>Logikus gondolkodás</p>	<p>Szemléltetés</p> <p>Projektor</p> <p>Önálló munka</p> <p>Megbeszélés</p> <p>Tábla</p> <p>Kooperatív oktatási módszer</p>

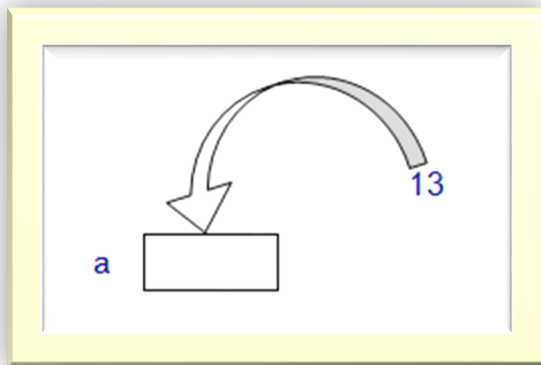
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
32 – 33'	<p>Győződjünk meg róla, hogy sejtésünk igaz-e!</p> <p>Hány elemet kellene még kiíratnunk?</p> <p><i>Válasz: Még négy elemet kellene kiíratnunk.</i></p> <p>Hogyan tudnánk ezt az ismétlő tevékenységet egyszerűbben lekódolni? Gondoljon bele, ha 100 elemű lenne a tömb!</p> <p><i>Válasz: Egyszerűbben iterációval, méghozzá, mivel ismerjük az ismétlések számát, ezért léptetési ciklussal tudjuk lekódolni.</i></p>	<p>Megfigyelés</p> <p>Figyelem felkeltése Visszacsatolás Ismétlés Fokozatosság</p> <p>Problémamegoldás</p>	<p>Kooperatív oktatási módszer</p> <p>Frontális munka</p>
33 – 36'	<p>Feladat: Írassa ki a tömb elemeit egymás mellé, tabulátorral elválasztva, használva a már megismert léptetési ciklust! Készítsük el a kiíratás tervét, válaszszuk ki a hozzá megfelelő alkalmazást!</p> <div data-bbox="528 973 1120 1308" data-label="Diagram"> </div>	<p>Eszközválasztás Alkalmazás Algoritmikus gondolkodás Struktogram készítése Precizitás, pontosság Alkotó képesség</p> <p>Struktogram készítése vagy</p> <p>Struktogram elemzése</p>	<p>Structorizer vagy Microsoft Office Excel</p> <p>Irányított, önálló munka</p> <p>Ha kevés az idő a struktogramot megkapják hálózaton, elemezzük, és az alapján kódolnak.</p> <p>Beszélgetés Szemléltetés</p> <p>Projektor</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
36 – 40'	A terv alapján kódoljon!  Figyelje meg a futási eredményt!	Programkód írása Szintaktika követése Hibakeresés	Számítógép Fejlesztői környezet  Egyéni munka
40 – 41'	A kivetített kódrészlet alapján ellenőrizze munkáját!  Vesse össze a kétféle megoldást, nézze meg, mi a különbség a kettő között!	Szemléletesség	Szemléltetés  Projektor
41 – 42'	Melyik megoldást célszerű alkalmazni?  <pre>// tömb elemeinek kiíratása I.a) System.out.println("A tömb elemei: "); for (int i = 0; i &lt;= 4; i++) {     System.out.print(iArray[i] + "\t"); } System.out.println();  // tömb elemeinek kiíratása I.b) System.out.println("A tömb elemei: "); for (int i = 0; i &lt; 5; i++) {     System.out.print(iArray[i] + "\t"); } System.out.println();</pre>	Önellenőrzés  Megfigyelés  A megoldás sokoldalú elemzése  Meggyőzőképesség	Projektor NetBeans IDE 7.3  Vita

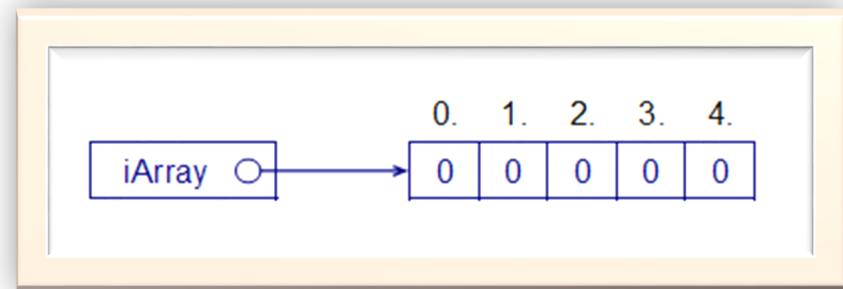
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>I.b)</p> <p>A tömb tulajdonságait figyelembe véve az 5 a tömb elemeinek a száma.</p> <p>Egy karakterrel kevesebbet kell gépelni ☺</p>	Döntésképeség	A következő órán a length bevezetése után még jobban megerősítést kap és tudatosan választásunk oka.
42 – 43'	<p>Mit állapíthatunk meg tehát a futási eredmény alapján?</p> <p>Az újonnan létrehozott tömb elemei alapértelmezés szerinti értékeket vesznek fel.</p> 	Általánosítás	<p>Szemplétetés</p> <p>Tábla</p> <p>A KÖVETKEZŐ ÓRA MEGTARTÁSÁRA KÖZVETLENÜL EZ UTÁN AZ ÓRA UTÁN KERÜL SOR, EZÉRT NINCS HÁZI FELADAT.</p>
43 – 44'	Rendszerezés	Rendszerezés	Kooperatív oktatási módszer
44 – 45'	Tanulók munkájának értékelése	Értékelés	Megbeszélés

# TÁBLAKÉP

```
int a = 13;
```



```
int[] iArray = new int[5];
```



## Feladatok

### 1. feladat:

Döntse el, melyik tömb a következők közül?

```
int[] iArray1, iArray2;  
int iArray3[], iArray4;
```

### 2. feladat:

Deklaráljon egy iArray nevű, int típusú elemekből álló tömböt, a kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben!

Figyeljen a kód strukturáltságára!

### 3. feladat:

Hogyan tud létrehozni egy 5 elemű, int típusú tömböt? Fel tudja használni az előbbi deklarációt? Mindenki a fejlesztői környezetben próbálkozzon, ha hibát jelez a NetBeans IDE, próbálja megtalálni a hibát! A kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben!

### 4. feladat:

Írassa ki az iArray tömb első elemét a fejlesztői környezetben! A kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben! Ha hibát jelez a NetBeans IDE, próbálja megtalálni a hibát!

### 5. feladat:

Írassa ki a tömb elemeit egymás mellé, tabulátorral elválasztva, használva a már megismert léptetési ciklust! Készítse el a kiíratás tervét, válassza ki a hozzá megfelelő alkalmazást! A terv alapján kódoljon! Figyelje meg a futási eredményt!

## Reflexió

Új ismereteket feldolgozó óra.

A házi feladat nagyon jól előkészítette az új anyag bevezetését, a tömb adatszerkezet fogalmának kialakítását, és a figyelem felkeltését. Nagymértékben elősegítette a motiválást a CINEMA 4D programmal készített modell, melynek még az elkészítése felől is kérdezősködtek a diákok. Sikertelenül felkelteni érdeklődésüket új programok iránt, illetve így személyes példát mutattam az egész életen át tartó tanulás jelentőségére.

Nehéz elképzelni a tömb memóriefoglalását, ennek megértését ábrákkal tettem szemléletessé, de természetesen ennek rögzülése hosszú folyamat. Nagy hangsúlyt fektettem az elmondottak ismétlésére, mindezzel hozzájárulva az új fogalmak megerősítéséhez és rögzítéséhez.

Fontosnak tartom a dokumentáció folyamatos tanulmányozását, használatát és így egy tanári közlés helyett a diákok saját maguk tapasztalhatják meg a tényeket, ezzel elősegítve az olvasott angol nyelvű szakmai szöveg megértését.

A tömb kiíratását nem pusztán feladatként jelöltem meg, hanem ezáltal saját maguk észlelheték a diákok, hogy milyen alapértelmezett

értékeket vesznek fel egy int típusú tömb elemei. Az észrevételüket folyamatosan rögzítettem a táblán.

A struktogramot sikerült elkészíteni, de lehet, hogy az óra végi összefoglalás így nem volt olyan teljeskörű, de ez nem probléma, mert a következő órát, ami közvetlenül ez után jött, ezzel kezdtem.

Többféle megoldási alternatívákra hívtam fel kódozásnál a figyelmüket, valamint arra, érdemes mindig átgondolni, hogy melyik a hatékonyabb megoldás. Az, hogy miért is esett a választás a b) megoldásra, azt a következő órán értették meg igazán a length konstans bevezetésével.

A folyamatos visszacsatolás és a fokozatosság elvének betartását követve a diákok szinte saját maguk jöttek rá és mondták ki az új ismereteket, így elmondhatom, hogy a logikus gondolkodás fejlesztése végig kísérte az órát. Természetesen az új ismereteket a tudományosság elvét követve mindig újra megfogalmaztam, mindezzel hozzájárulva a hallott szakmai szöveg megértésének fejlesztéséhez, a szakmai nyelvű beszédkésztség kialakításához és az anyag megerősítéséhez.

## ÓRATERVEZET - A STRING OSZTÁLY ÉS A TÖMBÖK. TÚLINDEXELÉS.

**A pedagógus neve:** Kovács Henriette

**A szakmacsoport száma és megnevezése:** 7. Informatika

**A szakképesítés azonosító száma:** 54 213 05

**A szakképesítés megnevezése:** Szoftverfejlesztő

**Iskolai rendszerű szakképzés** (a szakképzési évfolyamok száma: 2)

**Szakmai követelménymodul:** 10817-12 Hálózatok, programozás és adatbázis-kezelés

**Tantárgy:** Adatbázis- és szoftverfejlesztés gyakorlat

**Programozási nyelv:** Java

**Fejlesztői környezet:** NetBeans IDE 7.3

**Osztály - csoport:** I/13. S - SZ2

**Az óra tananyaga:** A String osztály és a tömbök. Túlindexelés.

## **Az óra cél- és feladatrendszere**

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonómiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt.
- Korszerű alkalmazói készség kialakítása: legyenek képesek arra, hogy a szoftvereket célszerűen használják.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.

- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, létrehozni.*
- *Legyen képes a tömbök elemeit elérni.*
- *Tudjon tömböt feltölteni, tömb elemeit megjeleníteni.*
- *Lássa a hasonlóságot String és tömb indexelésében.*
- *Tudja mi a különbség a length() metódus és a length konstans között.*
- *Legyen képes felismerni, diagnosztizálni a túlindexelés jelenségét, ezt követően legyen képes azt javítani.*

## **Fejlesztendő készségek, képességek**

- Alkalmazás, feladatértelmezés, fogalomalkotás, kommunikációs képesség, logikus gondolkodás, önellenőrzés, precizitás, pontosság, struktogram készítése, programkód írása, visszaemlékezés, diagnosztizálás, hibakeresés.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanuló társa is fel tudja használni.

### **Az óra didaktikai feladatai**

- Aktivizálás
- Ismétlés
- Megerősítés
- Fokozatosság
- A tanulók informálása a tanulási célokról
- Szemléletesség
- Visszacsatolás
- Összehasonlítás
- Következtetés
- A tények sokoldalú elemzése
- Új ismeret közlése
- Rögzítés
- Ellenőrzés
- A futási eredmény elemzése
- Gyakorlás

**Tantárgyi kapcsolatok:** Adatbázis- és szoftverfejlesztés elmélet, Matematika, Hálózati ismeretek I., Informatikai alapismeretek

**Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):**

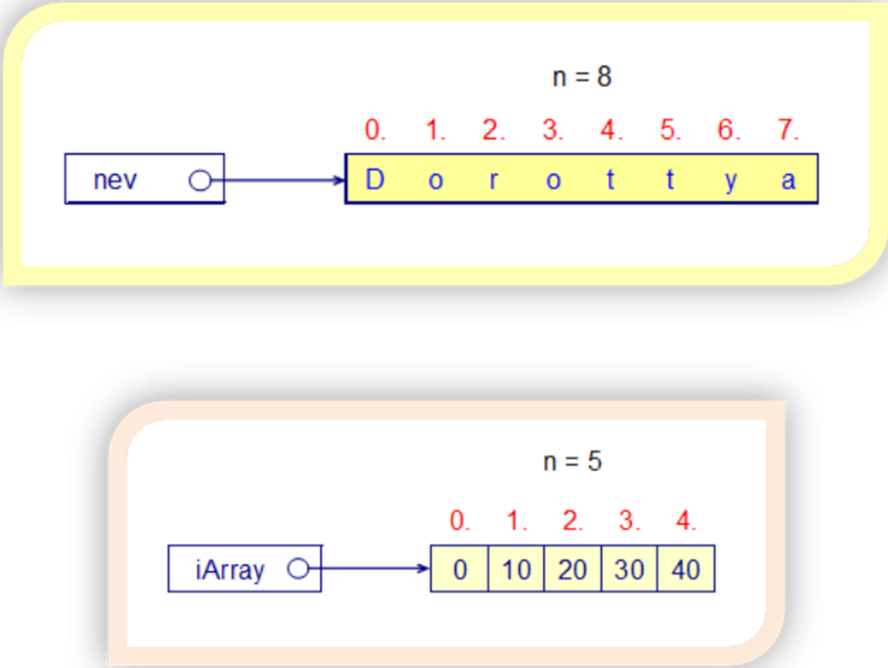
Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyiné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

<http://docs.oracle.com/javase/7/docs/api/>

Dátum: éééé. hh. nn.

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 5'	<p>Elmúlt órán tanultak felelevenítése:</p> <p>Mi tette szükségessé a tömb adatszerkezet bevezetését?</p> <p>Hogyan deklarálunk egy tömböt?</p> <p>A tömbképző operátort szintaktikailag hol kell elhelyezni?</p> <p>Deklaráció hatására jön létre objektum?</p> <p>Melyik operátorral hozunk létre tömböt?</p> <p>A tömb méretét mibe kell tenni?</p> <p>A memóriában hol helyezkednek el a tömb adatelemei?</p> <p>Hány eleme lehet egy tömbnek?</p> <p>Egy int típusú tömb elemei vesznek-e fel alapértelmezett értékeket? Miket?</p> <p>(Az elkövetkezőkben meg fogjuk vizsgálni azt is, hogy más elemtípusú tömb elemei milyen alapértelmezett értékeket vesznek fel.)</p> <p>Melyik vezérlési szerkezetet használtuk fel a tömb elemeinek kiíratására?</p> <p>Hogyan hivatkozunk a tömb elemeire?</p>	<p>Visszaemlékezés Kommunikációs képesség</p> <p>Aktivizálás Ismétlés Megerősítés Fokozatosság</p> <p>A tanulók informálása a tanulási célokról</p>	<p>Ez az óra közvetlenül kapcsolódik az előző órához.</p> <p>Beszélgetés</p> <p>Frontális munka</p> <p>Projektor</p>

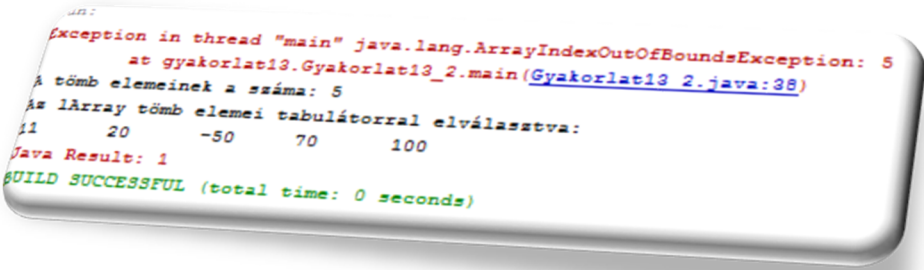
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK																
5 – 7'	<p>Hol találkoztunk már az index fogalmával?</p> <p>A String objektum szövege unicode karakterek sorozata, a karakterek sorszámozottak: ha a szöveg n karaktert tartalmaz, vagyis a szöveg hossza n, akkor a legelső karakter sorszáma, indexe 0, az utolsóé pedig n-1. Tehát hasonlóan, mint a tömböknél.</p> <div><div>n = 8</div><table><tr><td>0.</td><td>1.</td><td>2.</td><td>3.</td><td>4.</td><td>5.</td><td>6.</td><td>7.</td></tr><tr><td>D</td><td>o</td><td>r</td><td>o</td><td>t</td><td>t</td><td>y</td><td>a</td></tr></table></div> <p>Hasonlítsuk össze egy String objektum és egy tömb objektum memóiafoglalását!</p> <p>Diktálják, milyen kódot írjak az ábrák fölé?</p> <pre>String nev = new String("Dorottya");</pre> <pre>int[] iArray = new int[5];</pre>	0.	1.	2.	3.	4.	5.	6.	7.	D	o	r	o	t	t	y	a	<div>Visszaemlékezés</div> <div>Szemléletesség Visszacsatolás Megerősítés</div> <div>Logikus gondolkodás</div> <div>Összehasonlítás</div>	<div>Kooperatív oktatási módszer Elbeszélés</div> <div>Frontális munka</div> <div>Tábla</div> <p>A diákok mondják, hogy mit írjak az ábrák fölé, melyeket felrajzolok a táblára.</p> <div>Tábla</div>
0.	1.	2.	3.	4.	5.	6.	7.												
D	o	r	o	t	t	y	a												

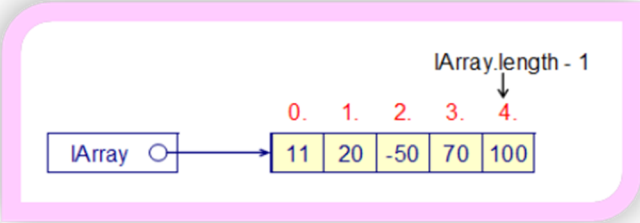
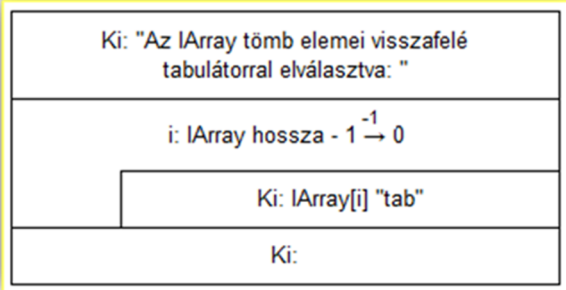
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
7 – 9'	 <p>The diagram illustrates memory storage for a string and an array. The top part shows a variable 'nev' pointing to a memory block containing the string 'Dorottya'. The indices 0 through 7 are shown above the characters. The bottom part shows a variable 'iArray' pointing to an array containing the values 0, 10, 20, 30, and 40. The indices 0 through 4 are shown above the values.</p>	<p>Szemléletesség Összehasonlítás</p>	<p>Frontális munka Kooperatív oktatási módszer</p>
9 – 11'	<p>Hogyan tudjuk lekérdezni egy szöveg hosszát?</p> <p>A String osztály <u>length()</u> metódusa visszaadja a szöveg hosszát. A visszatérési érték int típusú.</p> <p>Mit gondolnak vajon le tudom kérdezni a tömb hosszát is?</p>	<p>Visszaemlékezés Visszacsatolás Megerősítés</p> <p>Következtetés</p>	<p>Megbeszélés</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>Minden tömb objektumnak van egy <u>length konstansa</u>, amely megadja a tömb hosszát.</p> <p><code>&lt;tömbAzonosító&gt;.length</code></p> <p>A length értéke a tömb létrehozásától állandó, nem lehet megváltoztatni.</p> <p>A tömb indexei 0 és length-1 közöttiek.</p>	<p>A tények sokoldalú elemzése → új ismeret közlése</p> <p>Fokozatosság</p>	<p>Elbeszélés</p>
11 – 23'	<p>Feladat:</p> <p>1) Hozzon létre egy lArray nevű, long elemtípusú, 5 elemű tömböt! Készítsen a feladathoz struktogramot!</p> <p>a) Nézze meg, hogy a length konstans valóban visszaadja-e a tömb hosszát!</p> <p>b) Töltse fel a tömb elemeit a következő számokkal: 11, 20, -50, 70, 100!</p> <p>c) Írassa ki a tömb elemeit a length konstans felhasználásával, tabulátorokkal elválasztva!</p> <p>A feladatot a Gyakorlat13 projekt gyakorlat13 csomagjának, Gyakorlat13_2 néven létrehozott osztályába készítse el!</p>	<p>Feladatértelmezés Alkalmazás Precizitás, pontosság Struktogram készítése Programkód írása</p> <p>Aktivizálás Rögzítés Megerősítés Ellenőrzés</p> <p>Gyakorlás</p>	<p>Egyéni munka</p> <p>Számítógép Structorizer Java 1.5 NetBeans IDE 7.3</p> <p>Közben segítek annak, aki igényli és követem a munkájukat.</p> <p>Lehetséges, hogy túlindexelési problémába futnak. → Következő feladat</p> <p>Magyarázat</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK											
23 – 25'	<div><table><tr><td>tömb (lArray, 5)</td></tr><tr><td>n = lArray hossza</td></tr><tr><td>Ki: "A tömb elemeinek a száma: " n</td></tr><tr><td>iArray[0] = 11</td></tr><tr><td>iArray[1] = 20</td></tr><tr><td>iArray[2] = -50</td></tr><tr><td>iArray[3] = 70</td></tr><tr><td>iArray[4] = 100</td></tr><tr><td>Ki: "Az lArray tömb elemei tabulátorral elválasztva: "</td></tr><tr><td>i: 0 → lArray hossza - 1</td></tr><tr><td><div>Ki: lArray[i] "tab"</div></td></tr></table></div> <p>A tömb utolsó elemének indexe a tömb hossza-1. A kódolásnál, ha csak a kisebb relációjelet használja, akkor utána a tömb hosszát kell csak beírnia, ellenben ha a kisebb-egyenlő relációjelet használja, akkor utána a tömb hossza-1-et kell beírnia, úgymint a struktogramban.</p>	tömb (lArray, 5)	n = lArray hossza	Ki: "A tömb elemeinek a száma: " n	iArray[0] = 11	iArray[1] = 20	iArray[2] = -50	iArray[3] = 70	iArray[4] = 100	Ki: "Az lArray tömb elemei tabulátorral elválasztva: "	i: 0 → lArray hossza - 1	<div>Ki: lArray[i] "tab"</div>	<div>Önellenőrzés</div>	<div>Microsoft Office Excel 2007</div> <div>Projektor</div> <div>Magyarázat</div>
tömb (lArray, 5)														
n = lArray hossza														
Ki: "A tömb elemeinek a száma: " n														
iArray[0] = 11														
iArray[1] = 20														
iArray[2] = -50														
iArray[3] = 70														
iArray[4] = 100														
Ki: "Az lArray tömb elemei tabulátorral elválasztva: "														
i: 0 → lArray hossza - 1														
<div>Ki: lArray[i] "tab"</div>														

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
25 – 27'	 <pre> 5 package gyakorlat13; 6 /** 7  * 8  * @author Kovács Henriette 9  */ 10 public class Gyakorlat13_2 { 11 12     public static void main(String[] args) { 13         // 1. feladat 14 15         // tömb deklarálása és létrehozása 16         long[] lArray = new long[5]; 17 18         // tömb elemeinek a száma 19         int n = lArray.length; 20         System.out.println("A tömb elemeinek a száma: " + n); 21 22         // tömb elemeinek a feltöltése 23         lArray[0] = 11; 24         lArray[1] = 20; 25         lArray[2] = -50; 26         lArray[3] = 70; 27         lArray[4] = 100; 28 29         // tömb elemeinek kiírása I.C) 30         System.out.println("Az lArray tömb elemei tabulátorral elválasztva:"); 31         for (int i = 0; i &lt; lArray.length; i++) { 32             System.out.print(lArray[i] + "\t"); 33         } 34         System.out.println(); 35     } 36 } </pre>	<div data-bbox="1335 440 1637 531">Diagnosztizálás Hibakeresés</div> <div data-bbox="1335 603 1637 652">Önellenőrzés</div>	<div data-bbox="1711 576 2089 620">Projektor</div>
27 – 29'	<p>Feladat:</p> <p>2) Nézzük meg, mi történik, ha a tömb egy nem létező elemére hivatkozunk! Pl. az lArray tömb 5., nem létező elemét próbáljuk meg feltölteni -13-mal! Figyeljük meg kapunk-e és honnan erről valamilyen hibaüzenetet!</p>	<div data-bbox="1335 1090 1637 1134">Feladatértelmezés</div>	<div data-bbox="1711 1090 2089 1134">Egyéni munka</div> <div data-bbox="1711 1145 2089 1275">Számítógép Java 1.5 NetBeans IDE 7.3</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
29 – 31'	<p>A NetBeans nem jelez semmilyen problémát! Futtatás során azonban hibába futunk!</p>  <p>A Java a programvégrehajtás alatt figyelmeztetést küld, ha egy hivatkozás egy tömb érvénytelen elemére mutat, és a Java-értelmező, az interpreter megszakítja működését az <u>ArrayIndexOutOfBoundsException</u> hibaüzenettel.</p> <p>Kommentezze ki a sort!</p>	<p>A futási eredmény elemzése</p> <p>Fogalomalkotás</p>	<p>Frontális munka</p> <p>Kooperatív oktatási módszer</p> <p>Magyarázat</p>
31 – 37'	<p>Feladat:</p> <p>3.) Írassa ki az lArray tömb elemeit fordított sorrendben!</p> <p>Készítsen a kiíratáshoz struktogramot!</p> <p>Ha nem megy, segítségül a hálózaton található egy kép Visszafele néven.</p>	<p>Feladatértelmezés Logikus gondolkodás Alkalmazás Gyakorlás</p>	<p>Egyéni munka</p>

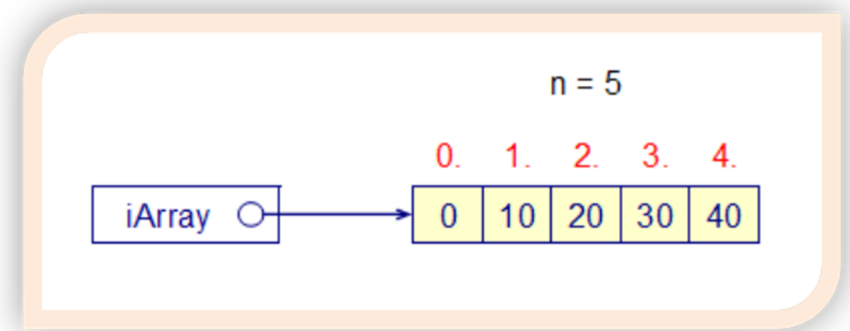
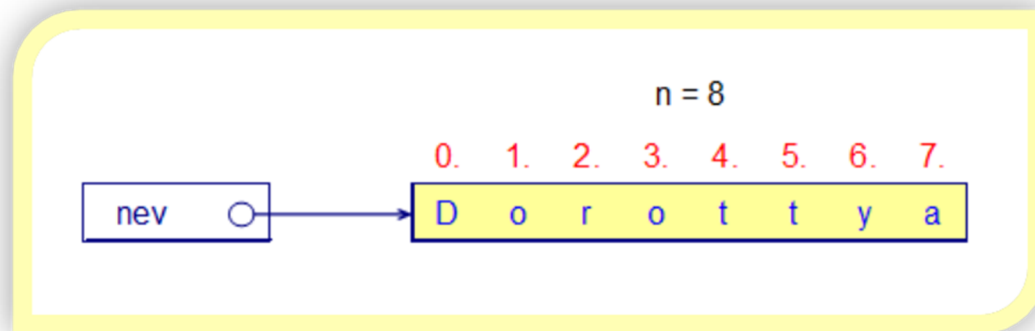
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
37 – 38'		<div>Szemléletesség</div> <div>Rögzítés</div>	<div>Szemléltetés</div> <div>Visszafele.png</div>
38 – 39'		<div>Ellenőrzés</div> <div>Önellenőrzés</div>	<div>Microsoft Office Excel 2007</div> <div>Projektor</div>
39 – 40'	<pre>// 3.feladat System.out.println("Az lArray tömb elemei visszafelé " + "tabulátorral elválasztva:"); for (int i = lArray.length - 1; i &gt;= 0; i--) {     System.out.print(lArray[i] + "\t"); } System.out.println();</pre>	<div>Ellenőrzés</div> <div>Diagnosztizálás</div> <div>Hibakeresés</div> <div>Önellenőrzés</div>	<div>Projektor</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
40 – 44'	Rendszerezés: 1. Indexelés 2. Memóriafooglalás 3. length 4. ArrayIndexOutOfBoundsException	Rendszerezés	Kooperatív oktatási módszer A KÖVETKEZŐ ÓRA MEGTARTÁSÁRA KÖZVETLENÜL EZ UTÁN AZ ÓRA UTÁN KERÜL SOR, EZÉRT NINCS HÁZI FELADAT.
44 – 45'	Tanulók munkájának értékelése	Értékelés	Megbeszélés

# TÁBLAKÉP

```
String nev = new String("Dorottya");
```

```
int[] iArray = new int[5];
```



## Feladatok

1.) Hozzon létre egy lArray nevű, long elemtípusú, 5 elemű tömböt!  
Készítsen a feladathoz struktogramot!

- a) Nézze meg, hogy a length konstans valóban visszaadja-e a tömb hosszát!
- b) Töltse fel a tömb elemeit a következő számokkal: 11, 20, -50, 70, 100!
- c) Írassa ki a tömb elemeit a length konstans felhasználásával, tabulátorral elválasztva!

2.) Nézzük meg, mi történik, ha a tömb egy nem létező elemére hivatkozunk! Pl. az lArray tömb 5., nem létező elemét próbáljuk meg feltölteni -13-mal! Figyeljük meg kapunk-e és honnan erről valamilyen hibaüzenetet!

3.) Írassa ki az lArray tömb elemeit fordított sorrendben!

Készítsen a kiíratáshoz struktogramot!

Ha nem megy, segítségül a hálózaton található egy kép Vissza-fele néven.

## Reflexió

Új ismereteket feldolgozó óra

Úgy érzem nagyon hasznos volt az a választás, hogy összehasonlítottam több szempontból is (indexelés, memórafoglalás, hossz) a tömböket és a String osztály objektumait. Ezzel elősegítettem a Stringről tanultak elmélyítését és a tömb ismeretanyagának rögzítését. A tények sokoldalú elemzésével rávilágítottam a hasonlóságokra és a különbségekre, mely a fogalmak rendszerezését nagyban elősegítette.

A for ciklusban a length konstans használatával a precizitás és pontosság kompetenciákat fejlesztettem, de ennek elmélyítését a továbbiakban fontos feladatommak tartom, mivel nehézséget okozott az  $i \leq \text{lArray.length} - 1$  és az  $i < \text{lArray.length}$  azonosságának megértése.

A struktogram elkészítésénél az új jelölések készségszinten természetesen még nem fejlődtek ki, ebben sokat segítetttem, sőt a struktogramokat részben közösen készítettük el.

Az `ArrayIndexOutOfBoundsException` hibaüzenet megjelenése meglepetésemre nem okozott nehézséget senkinek, sőt a kifejezést még meg is mosolyogták, amikor kimondtam. Örömmel fogadták, hogy ha esetleg ilyen típusú hibát ejtenek, az interpreter jelez nekik. Megértették, hogy azért kell ismerniük az Exception-öket, mert így érteni fogják, hogy mi a hiba oka.

A tömb elemeinek fordított sorrendben történő kiírása két fontos célt is szolgált, egyrészt a léptetési ciklus megerősítését, valamint az indexek helyes használatát. Sejtésem beigazolódtott, egy diáknak problémát okozott és hasznos volt a hálózatra feltett kép, mely a szemléltetés és vizualitás révén segítséget nyújtott neki.

## **ÓRATERVEZET - TÖMB ELEMEINEK ALAPÉRTELMEZETT ÉRTÉKEI. TÖMB FELTÖLTÉSE.**

**A pedagógus neve:** Kovács Henriette

**A szakmacsoport száma és megnevezése:** 7. Informatika

**A szakképesítés azonosító száma:** 54 213 05

**A szakképesítés megnevezése:** Szoftverfejlesztő

**Iskolai rendszerű szakképzés** (a szakképzési évfolyamok száma: 2)

**Szakmai követelménymodul:** 10817-12 Hálózatok, programozás és adatbázis-kezelés

**Tantárgy:** Adatbázis- és szoftverfejlesztés gyakorlat

**Programozási nyelv:** Java

**Fejlesztői környezet:** NetBeans IDE 7.3

**Osztály - csoport:** I/13. S - SZ2

**Az óra tananyaga:** Tömb elemeinek alapértelmezett értékei. Tömb feltöltése.

## Az óra cél- és feladatrendszere

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonómiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt.
- Korszerű alkalmazói készség kialakítása: legyenek képesek arra, hogy mindezeket célszerűen használják.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, létrehozni.*
- *Legyen képes a tömbök elemeit elérni.*
- *Tudjon tömböt szabály alapján léptetési ciklussal feltölteni.*
- *Tudjon tömböt megjeleníteni, elemeit szóközzel vagy tabulátorral elkülöníteni.*
- *Tudja primitív és összetett elemtípusú tömbök elemeinek alapértelmezett értékeit meghatározni.*

### **Fejlesztendő készségek, képességek**

- Algoritmus megvalósítása, algoritmus módosítása, diagnosztizálás, hibakeresés, feladatértelmezés, logikus gondolkodás, önállóság, önellenőrzés, precizitás, pontosság, programkód írása, programkód átalakítása, javítása, programkód módosítása, szintaktika követése, visszaemlékezés, kommunikációs képesség, olvasott szakmai szöveg megértése, struktogram átalakítása, struktogram készítése, összefüggések felismerése.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.

- Legyen képes elemi és összetett adatok megkülönböztetésére, kezelésére, használatára.
- Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.

### **Az óra didaktikai feladatai**

- A figyelem felkeltése
- A szükséges előzetes ismeretek felidézése
- A tanulók informálása a tanulási célokról
- A tények elemzése
- Absztrakciók
- Aktivizálás
- Általánosítások
- Az új ismerteket megalapozó tények biztosítása
- Ellenőrzés
- Értékelés
- Fogalomalkotás
- Fokozatosság
- Gyakorlás
- Információ keresése, elemzése.
- Ismétlés
- Következtetés
- Megerősítés
- Rendszerezés
- Új ismeret közlése

**Tantárgyi kapcsolatok:** Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek, Biológia

**Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):**

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyiné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

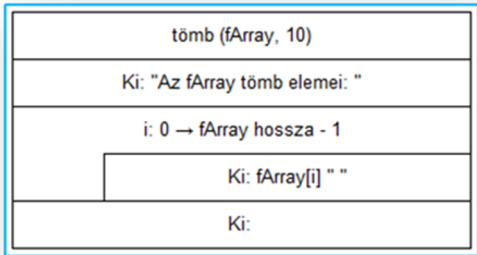
<http://docs.oracle.com/javase/7/docs/api/>

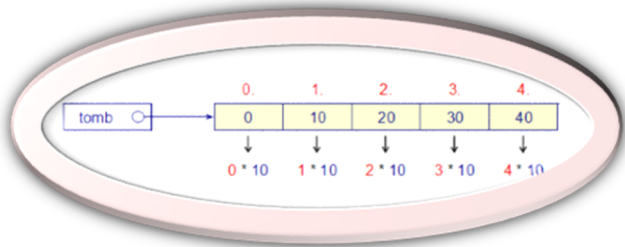
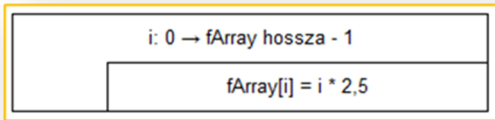
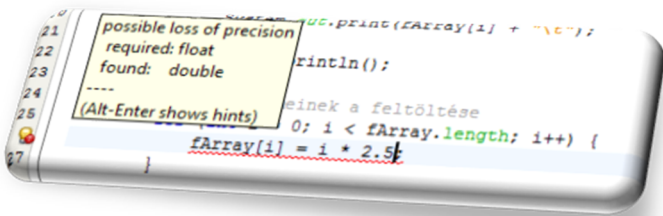
<http://unicode-table.com/en/>

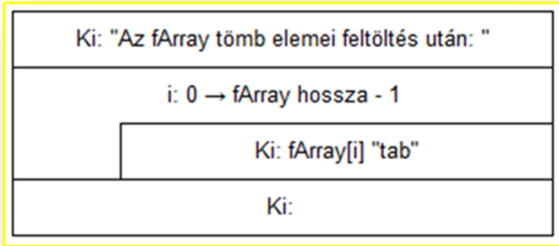
Dátum: éééé. hh. nn.

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 5'	<p>Tanórát előkészítő kérdések:</p> <ul style="list-style-type: none"> <li>– A length szó kapcsán mi jut eszébe?</li> <li>– Tömbök esetében hol célszerű használni a length konstanst?</li> <li>– Milyen alapértelmezett értékeket vesznek fel az int, long elemtípusú tömb elemei?</li> <li>– Tudna-e olyan elemtípust mondani, ahol ugyanez az érték jelenne meg?</li> <li>– Milyen primitív típusok vannak még a Java-ban?</li> <li>– A mai órán megtagasztaljuk, hogy valós, logikai, karakter és összetett típus esetén mik az alapértelmezett értékei a tömb elemeinek.</li> <li>– Hogyan töltöttük fel a tömb elemeit?</li> </ul> <p><i>Válasz: egyesével.</i></p> <ul style="list-style-type: none"> <li>– A mai órán azt is megtanuljuk, hogyan tudnánk sokkal hatékonyabban feltölteni a tömböket.</li> </ul> <p>Gyakorlat13/gyakorlat13: Gyakorlat13_3 osztály létrehozása</p>	<p>Visszaemlékezés Kommunikációs képesség</p> <p>Aktivizálás Ismétlés Megerősítés Fokozatosság</p> <p>A tanulók informálása a tanulási célokról</p>	<p>Ez az óra közvetlenül kapcsolódik az előző órához.</p> <p>Beszélgetés</p> <p>Frontális munka</p> <p>Projektor</p> <p>Java 1.5 NetBeans IDE 7.3</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
5 – 8'	<p>A feladat előkészítése:</p> <ul style="list-style-type: none"> <li>✓ Felhívom a diákok figyelmét arra, hogy ha a tömb elemei és az indexek között valamilyen összefüggést, szabályt találnak, akkor azt alkalmazva egy léptetési ciklus segítségével fel tudják tölteni a tömböt. Ennek megértésére feltettem egy képet <i>Feltoltes_ciklussal</i> néven a hálózatra.</li> <li>✓ Átismétlem, hogy mi a literál és milyen típusú az egész és valós literál.</li> <li>✓ Az ismétlődő kódrészleteket a megfelelő változtatással másolják!</li> </ul>	<p>Visszaemlékezés</p> <p>Figyelem felkeltése Aktivizálás Szükséges előzetes ismeretek felidézése</p>	<p>Együttműködő oktatási módszer Szemléltetés</p> <p>Frontálismunka</p> <p>Projektor Hálózat</p>
8 – 9'	<p>Feladat: Minden feladathoz készítsen struktogramot!</p> <p>1.) Hozzon létre egy fArray nevű, float elemtípusú tömböt, melynek 10 eleme van!</p> <p>a) Írassa ki a tömb elemeinek értékét egymás mellé szóközzel elválasztva! Milyen következtetéseket von le belőle? Írja be megjegyzésbe!</p> <p>b) Töltse fel a tömb elemeit az alábbi értékekkel!</p> <p>0.0   2.5   5.0   7.5   10.0   12.5   15.0   17.5   20.0   22.5</p> <p>c) Írassa ki a tömb elemeit tabulátorral elválasztva!</p> <p>Ha a fejlesztői környezet felhívja figyelmét egy lehetséges pontosságvesztésre, gondoljon arra, hogy egy valós literál milyen típusú!</p>	<p>Feladatértelmezés Logikus gondolkodás Összefüggések felismerése</p>	<p>Projektor</p> <p>Megbeszélés</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
9 – 13	<p>1.a) feladat tervezése</p> 	<p>Algoritmus tervezése Struktogram készítése Visszaemlékezés</p> <p>Visszacsatolás Gyakorlás</p>	<p>Egyéni munka</p> <p>Figyelemmel kísérem a diákok munkáját, ha szükséges, vagy igénylik segítséget nyújtok nekik.</p> <p>Számítógép Structorizer</p>
13 – 17'	<p>1.a) feladat kódolása</p> <pre> 5 package gyakorlat13; 6 7 /** 8  * 9  * @author Kovács Henriette 10 */ 11 public class Gyakorlat13_3 { 12     public static void main(String[] args) { 13         // 1. feladat 14 15         float[] fArray = new float[10]; 16 17         // tömb elemeinek kiírása 18         System.out.println("Az fArray tömb elemei: "); 19         for (int i = 0; i &lt; fArray.length; i++) { 20             System.out.print(fArray[i] + " "); 21         } 22         System.out.println(); </pre> <p>A float elemtípusú tömb elemeinek alapértelmezett értéke: 0,0. Ez a double elemtípusú tömb elemeire is igaz, tehát megállapíthatjuk, hogy a valós elemtípusú tömb elemeinek alapértelmezett értéke: 0,0.</p>	<p>Precizitás, pontosság Szintaktika követése Programkód írása Algoritmus megvalósítása</p> <p>Gyakorlás Ellenőrzés Következtetés Általánosítás Új ismeret közlése</p>	<p>Egyéni munka</p> <p>Számítógép Java 1.5 NetBeans IDE 7.3</p> <p>Megbeszélés</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
17 – 19'	<p>Feltoltes_ciklussal.png – Szabály felismerése az index és a tömbelem értéke között:</p> 	<p>Szabály felismerése</p>	<p>Feltoltes_ciklussal.png</p> <p>Szemléltetés</p> <p>Megbeszélés</p>
19 – 21'	<p>1.b) feladat tervezése</p> 	<p>Struktogram átalakítása, készítése</p> <p>Önállóság</p>	<p>Egyéni munka</p> <p>Structorizer</p>
21 – 23'	<p>NetBeaans figyelmeztet a lehetséges pontosságvesztésre: possible loss of precision</p> 	<p>Olvasott szakmai angol szöveg értése</p> <p>Diagnosztizálás, hibakeresés</p>	<p>Együttműködő oktatási módszer</p> <p>Projektor</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
23 – 24'	1.b) feladat kódolása <pre> 23 24 // tömb elemeinek a feltöltése 25 for (int i = 0; i &lt; fArray.length; i++) { 26     fArray[i] = i * 2.5F; 27 } 28 </pre>	Programkód átalakítása, javítása Szintaktika követése	Egyéni munka Számítógép Java 1.5 NetBeans IDE 7.3
24 – 27'	1.c) feladat tervezése 	Algoritmus módosítása	Egyéni munka Structorizer
27 – 29'	1.c) feladat kódolása <pre> 28 29 // tömb elemeinek kiírása feltöltés után 30 System.out.println("Az fArray tömb elemei " 31     + "feltöltés után: "); 32 for (int i = 0; i &lt; fArray.length; i++) { 33     System.out.print(fArray[i] + "\t"); 34 } 35 System.out.println(); 36 } 37 </pre>	Programkód módosítása Precizitás, pontosság Szintaktika követése Önellenőrzés	Egyéni munka Számítógép Java 1.5 NetBeans IDE 7.3 Projektor

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
29 – 30'	A 2., 3., 4. feladat nem véletlenszerűen, tudatosan felosztásra kerül a 3 diák között.	Fokozatosság Precizitás, pontosság	Differenciálás
30 – 35'	<pre>// 2. feladat  boolean[] bArray = new boolean[4];  // bArray tömb elemeinek kiíratása System.out.println("A bArray tömb elemei: "); for (int i = 0; i &lt; bArray.length; i++) {     System.out.print(bArray[i] + "\t"); } System.out.println();  // bArray tömb elemeinek a feltöltése bArray[0] = true; bArray[1] = false; bArray[2] = true; bArray[3] = false;  // bArray tömb elemeinek kiíratása feltöltés után System.out.println("A bArray tömb elemei "     + "feltöltés után:"); for (int i = 0; i &lt; bArray.length; i++) {     System.out.print(bArray[i] + "\t"); } System.out.println();</pre>	<div>Programkód írása</div> <div>Az új ismerteket meg- alapozó tények biztosí- tása A tények elemzése</div> <div>Ellenőrzés</div>	<p>Struktogramot nem kérek – eddigi struktogramok és kódok felhasználása</p> <div>Egyéni munka</div> <div>NetBeans IDE 7.3</div> <p>A megoldást a hálózatra a bead mappába rakatom fel.</p> <p>Kivetítem</p> <div>projektor</div> <p>segítségével.</p>
35 – 36'	Egy boolean elemtípusú tömb elemeinek alapértelmezés szerinti értéke: <u>false</u>	Fogalomalkotás, kö- vetkeztetés, absztrak- ció, általánosítás.	Tanulói elbeszélés

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
30 – 35'	<pre>// 3. feladat  char[] cArray = new char[4];  // cArray tömb elemeinek kiírása System.out.println("A cArray tömb elemei: "); for (int i = 0; i &lt; cArray.length; i++) {     System.out.print(cArray[i] + "mi van köztük"); } System.out.println();  // cArray tömb elemeinek a feltöltése cArray[0] = 'K'; cArray[1] = 'H'; cArray[2] = 'I'; cArray[3] = 'S';  // cArray tömb elemeinek kiírása feltöltés után System.out.println("A cArray tömb elemei "     + "feltöltés után"); for (int i = 0; i &lt; cArray.length; i++) {     System.out.print(cArray[i] + ". "); } System.out.println();</pre>	<div>Programkód írása</div> <div>Az új ismerteket meg- alapozó tények biztosí- tása A tények elemzése</div> <div>Ellenőrzés</div>	<div>Egyéni munka</div> <div>NetBeans IDE 7.3</div> <p>A megoldást a hálózatra a bead mappába rakatom fel.</p> <p>Kivetítem</p> <div>projektor</div> <p>segítségével.</p> <p>A helyes megoldások felkerülnek a hálózatra, így mindenkinek meglesz a „másik munkája”, melyeket be tudnak másolni a saját kódjukba.</p>
36 – 37'	<p>Egy char elemtípusú tömb elemeinek alapértelmezés szerinti értéke: <u>\u0000</u></p> <p>Nézzünk utána a böngészőben, hogy melyik kódú unicode karakter ez.</p> <p>PL.:</p> <p><a href="http://unicode-table.com/en/">http://unicode-table.com/en/</a></p>	<div>Fogalomalkotás, kö- vetkeztetés, absztrak- ció, általánosítás.</div>	<div>Tanulói elbeszélés</div> <div>Internet</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
37 – 39'	 <p>A Unicode a különböző írásrendszerek egységes kódolását leíró nemzetközi szabvány.</p> <p>A \u0000 a 0-s kódú unicode karakter.</p>	<div data-bbox="1335 209 1635 256">CÉLOK ÉS FELADATOK</div> <div data-bbox="1335 507 1635 588">Információ keresése, elemzése</div> <div data-bbox="1335 999 1635 1088">Megerősítés Új ismeret közlése</div>	<div data-bbox="1711 443 2089 488">Egyéni munka</div> <div data-bbox="1711 563 2089 652">Megbeszélés Magyarázat</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
30 – 35'	<pre>// 4. feladat  String[] sArray = new String[4];  // sArray tömb elemeinek kiíratása System.out.println("Az sArray tömb elemei: "); for (int i = 0; i &lt; sArray.length; i++) {     System.out.print(sArray[i] + "\t"); } System.out.println();  sArray[0] = "alma"; sArray[1] = "körte"; sArray[2] = "szilva"; sArray[3] = "eper";  // sArray tömb elemeinek kiíratása feltöltés után System.out.println("Az sArray tömb elemei "     + "feltöltés után: "); for (int i = 0; i &lt; sArray.length; i++) {     System.out.print(sArray[i] + "\t"); } System.out.println();</pre>	<div>Programkód írása</div> <div>Az új ismerteket meg- alapozó tények biztosí- tása A tények elemzése</div> <div>Ellenőrzés</div>	<div>Egyéni munka</div> <div>NetBeans IDE 7.3</div> <p>A megoldást a hálózatra a bead mappába rakatom fel.</p> <p>Kivetítem</p> <div>Projektor</div> <p>segítségével.</p> <p>Egészséges táplálkozás fontosságá- nak hangsúlyozása.</p> <p>Ha egy diák sokkal hamarabb vé- gez, mint a másik, akkor a másik két feladatból választ, hogy melyi- ket oldja meg.</p>
39 – 41'	<p>Egy String elemtípusú tömb elemeinek alapértelmezés szerinti értéke: <u>null</u></p> <p><u>Referenciák esetén ez az érték null.</u></p> <p>A következő órán megmutatom, mi történik a memóriában, ha a tömb összetett elemtípusú.</p>	<div>Fogalomalkotás, kö- vetkeztetés, absztrak- ció, általánosítás.</div> <div>Figyelem felkeltése</div>	<div>Tanulói elbeszélés</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
41 – 44'	<p>Rendszerezés:</p> <ol style="list-style-type: none"> <li>1. Egyszerű elemtípusú tömbök elemeinek alapértelmezett értékei</li> <li>2. Összetett elemtípusú tömbök elemeinek alapértelmezett értékei</li> <li>3. Tömbök feltöltési lehetőségei: <ul style="list-style-type: none"> <li>kódban egyesével</li> <li>szabály alapján</li> </ul> </li> </ol> <p>Mit gondolnak, hogy lehetne még feltölteni a tömböket?</p> <p>felhasználó által (a következő órán megnézzük)</p> <ol style="list-style-type: none"> <li>4. Tömbök kiírása <ul style="list-style-type: none"> <li>szóközzel, tabulátorral elválasztva</li> </ul> </li> </ol> <p>Hogy lenne a tömbök kiírása még szebb?</p> <p>vesszővel elválasztva</p>	<p>Rendszerezés</p> <p>Figyelem felkeltése</p>	<p>Együttműködő oktatási módszer</p> <p>A KÖVETKEZŐ ÓRA MEGTARTÁSÁRA KÖZVETLENÜL EZ UTÁN AZ ÓRA UTÁN KERÜL SOR, EZÉRT NINCS HÁZI FELADAT.</p>
44 – 45'	Tanulók munkájának értékelése	Értékelés	<p>Megbeszélés</p> <p>Egész órát végig kíséri az értékelés.</p>

## Feladatok

- 1.) Hozzon létre egy fArray nevű, float elemtípusú tömböt, melynek 10 eleme van! Minden feladathoz készítsen struktogramot!
  - a) Írassa ki a tömb elemeinek értékét egymás mellé szóközőkkel elválasztva! Milyen következtetéseket von le belőle? Írja be megjegyzésbe!
  - b) Töltse fel a tömb elemeit az alábbi értékekkel!

0.0	2.5	5.0	7.5	10.0	12.5	15.0	17.5
	20.0	22.5					
  - c) Írassa ki a tömb elemeit tabulátorral elválasztva!

Ha a fejlesztői környezet felhívja figyelmét egy lehetséges pontosságvesztésre, gondoljon arra, hogy egy valós literál milyen típusú!
- 2.) Hozzon létre egy bArray nevű, boolean elemtípusú tömböt, melynek 4 eleme van!
  - a) Írassa ki a tömb elemeinek értékét egymás mellé szóközőkkel elválasztva! Milyen következtetéseket von le belőle? Írja be megjegyzésbe!
  - b) Töltse fel a tömb elemeit tetszőleges logikai értékekkel!
  - c) Írassa ki a tömb elemeit tabulátorral elválasztva!
- 3.) Hozzon létre egy cArray nevű, char elemtípusú tömböt, melynek 4 eleme van!
  - a) Írassa ki a tömb elemeinek értékét egymás mellé szóközőkkel elválasztva! Milyen következtetéseket von le belőle? Írja be megjegyzésbe!
  - b) Töltse fel a tömb elemeit tetszőleges karakterekkel!
  - c) Írassa ki a tömb elemeit tabulátorral elválasztva!
- 4.) Hozzon létre egy sArray nevű, String elemtípusú tömböt, melynek 4 eleme van!
  - a) Írassa ki a tömb elemeinek értékét egymás mellé szóközőkkel elválasztva! Milyen következtetéseket von le belőle? Írja be megjegyzésbe!
  - b) Töltse fel a tömb elemeit egészséges élelmiszerekkel!
  - c) Írassa ki a tömb elemeit tabulátorral elválasztva!

## Reflexió

Ezen óra elemzése során a feladatsor mentén fogok haladni, mely formailag szinte ugyanazon feladatokat tartalmazza. Ez azonban csak a látszat. Minden feladat hordoz új ismeret megszerzésére irányuló törekvéseket, túlhaladva az informatikán, keresvén a kapcsolatot más tantárgyakkal, ami ilyen elvont, absztrakt ismeret megalapozása során nagyon fontos feladat, de ugyanakkor kihívás.

Beigazolódott, hogy nagyon jó választásnak bizonyult az 1. feladat. Összeállításánál több gondolat is vezérelt. Ezek a következők voltak: át szerettem volna ismételni gyakorlati feladaton keresztül a tömb létrehozását, feltöltését és megjelenítését; meg szerettem volna mutatni, hogy tömb elemeinek mik az alapértelmezett értékei valós elemtípus esetén; illetve azt, hogy egy tömb feltöltése nemcsak szekvenciákkal lehetséges. Először double elemtípusú tömbben gondolkoztam, aztán átírtam a feladatot float elemtípusúra, egyrészt azért, mert így lehetőségem volt átismételni literálok adattípusait, másrészt belefutottak a possible loss of precision üzenetbe. Valóban segítségül szolgált a hálózatra feltett Feltoltes\_ciklussal.png kép, főleg az egyik diáknak, aki a matematikával nincs olyan jó barátságban. ☺ A második kiíratásnál

- 1.c) feladat - azért változtattam a fehér szóközökön, hogy megerősítsem bennük azt a tényt, ha átvesznek valamilyen kódrészletet, mindig győződjenek meg arról, hogy az megfelel-e a specifikációnak.

A következtetés leírása gondot okozott, a megfogalmazás maga nem volt probléma, de ezt megjegyzésbe beírni nehezükre esett. Vagyis a kommentek írását a továbbiakban is következetesen kérni fogom.

A 2-4. feladat alkalmat nyújtott a differenciálásra és nagyon kézenfekvő volt. Láttam világosan, hogy arra nem lesz idő, hogy megtapasztalják azt, hogy logikai, karakter és összetett elmtípus esetén mik az alapértelmezett értékek és mivel hárman voltak így nehézségi sorrendben kiosztottam a feladatokat. Lehet, hogy megkérdezhettem volna őket, ki melyiket szeretné, és ugyanazt választják, amit én szántam volna nekik. Így még motiváltabbak lettek volna, de ez magában foglalta volna azt a lehetőséget, hogy nem tudták volna feltölteni a tömböt és nem lesz sikerélményük.

A tetszőleges elemekkel való feltöltés szintén nagyon tudatos volt részemről, mert nem igazán kreatívak, így ezt a képességet mindenképpen fejleszteni szerettem volna. Az utolsó feladatban egy kis megkötés azért volt, az egészséges élelmiszerekből lehetett csak választani. Ez irányban nagyon fogékony ez a csoport is, sokat

szoktam beszélgetni velük erről és úgy érzem próbálják betartani amiket elmagyarázok nekik ezzel kapcsolatban.

Feladataim a fokozatosság elvét követve jó szolgálatot tettek mind az új ismeretek elsajátításában, mind a gyakorlásban, rögzítésben.

## **ÓRATERVEZET - TÖMB ELEMEINEK FELTÖLTÉSE BILLENTYŰZETRŐL, KIÍRATÁSA VESSZŐVEL ELVÁLASZTVA.**

**A pedagógus neve:** Kovács Henriette

**A szakmacsoport száma és megnevezése:** 7. Informatika

**A szakképesítés azonosító száma:** 54 213 05

**A szakképesítés megnevezése:** Szoftverfejlesztő

**Iskolai rendszerű szakképzés** (a szakképzési évfolyamok száma: 2)

**Szakmai követelménymodul:** 10817-12 Hálózatok, programozás és adatbázis-kezelés

**Tantárgy:** Adatbázis- és szoftverfejlesztés gyakorlat

**Programozási nyelv:** Java

**Fejlesztői környezet:** NetBeans IDE 7.3

**Osztály - csoport:** I/13. S - SZ2

**Az óra tananyaga:** Tömb elemeinek feltöltése billentyűzetről, kiíratása vesszővel elválasztva.

### **Az óra cél- és feladatrendszere**

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonómiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt. Kialakítani a tanulóknál az informatikai ismereteik folyamatos bővítésének és megújításának igényét.
- Korszerű alkalmazói készség kialakítása: meglévő, - Internet segítségével - új alkalmazások kiválasztása a feladat tervezéséhez.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.

- Alkotó munkára nevelés.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, létrehozni.*
- *Legyen képes a tömbök elemeit absztrakt módon elérni.*
- *Tudjon tömböt billentyűzetről feltölteni.*
- *Tudjon tömbelemeket vesszővel elválasztva megjeleníteni.*
- *Legyen képes algoritmust alkotni, megvalósítani, struktogramot készíteni, programkódot írni.*
- *Legyen képes a diagnosztizálásra, hibakeresésre.*

### **Fejlesztendő készségek, képességek**

- Visszaemlékezés, kommunikációs képesség, feladatértelmezés, alkalmazásválasztás, algoritmus alkotása, algoritmus tervezése, algoritmus készítése, struktogram készítése, precizitás, pontosság, önellenőrzés, diagnosztizálás, programkód írása, szintaktika követése, absztrakt gondolkodás, logikus gondolkodás, algoritmikus gondolkodás, rendszerezés, figyelem felkeltése.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.

- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanuló társa is fel tudja használni.

#### **Az óra didaktikai feladatai**

- Motiválás
- A figyelem felkeltése
- A tanulók informálása a tanulási célokról.
- Aktivizálás
- Ismétlés
- A szükséges előzetes ismeretek felidézése a feladat megoldásához.
- Fokozatosság
- Figyelem felhívása
- Megerősítés
- Feladat kitűzése
- Gyakorlás
- Visszacsatolás
- Általánosítás
- Szemléletesség
- Rögzítés

**Tantárgyi kapcsolatok:** Adatbázis- és szoftverfejlesztés elmélet, Angol, Hálózati ismeretek I., Informatikai alapismeretek

**Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):**

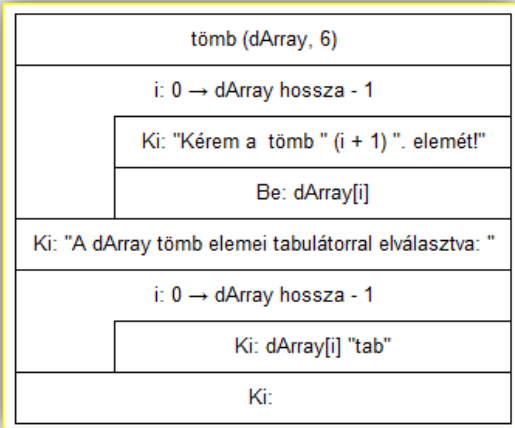
Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyiné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

<http://docs.oracle.com/javase/7/docs/api/>

Dátum: éééé. hh. nn.

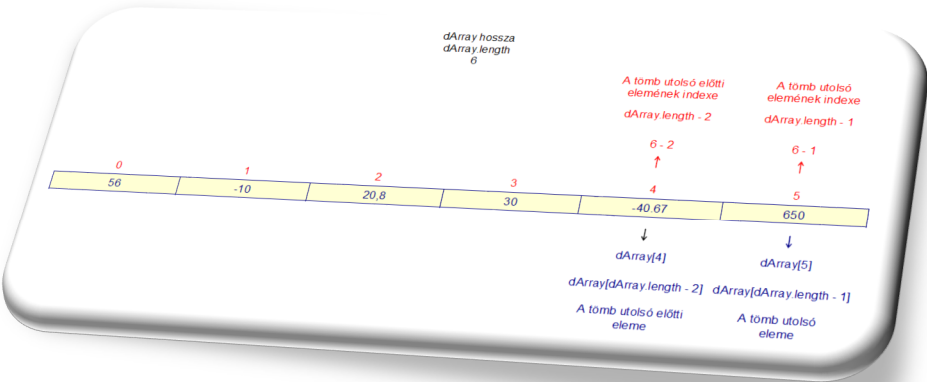
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 4'	<p>Az órát előkészítő kérdések:</p> <p>1. Hogyan tudunk létrehozni tömböket? new operátorral <i>meg fogunk ismerkedni még egy lehetőséggel, az úgynevezett inicializáló blokkal</i></p> <p>2. Hogyan tudunk feltölteni tömböket? egyesével, szekvenciával szabály alapján, vezérlési szerkezettel, léptetési ciklussal <i>felhasználó által (ezen az órán)</i> <i>véletlenszerűen</i></p> <p>3. Hogyan végeztük tömbök kiíratását? szóközzel, tabulátorral elválasztva <i>vesszővel elválasztva (ezen az órán)</i></p> <p>Hozzanak létre a Gyakorlat13 projekt gyakorlat13 csomagjában egy Gyakorlat13_4 osztályt!</p>	<p>Visszaemlékezés Kommunikációs képesség</p> <p>Motiválás A figyelem felkeltése A tanulók informálása a tanulási célokról.</p> <p>Aktivizálás Ismétlés Fokozatosság</p>	<p>Ez az óra közvetlenül kapcsolódik az előző órához.</p> <p>Beszélgetés</p> <p>Frontális munka</p> <p>Projektor NetBeans IDE 7.3 Java 1.5</p>

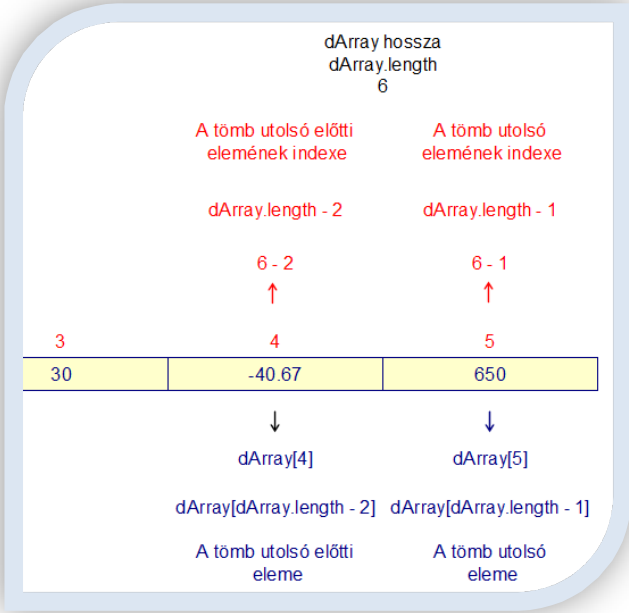
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
4 – 5'	<p>Feladat kiadása:</p> <p>Olvassa el figyelmesen a feladatot!</p> <p>1.) A feladathoz készítsen struktogramot!</p> <p>Hozzon létre egy dArray nevű, double elemtípusú tömböt, melynek 6 eleme van!</p> <p>a) Töltse fel a tömböt a felhasználó által megadott értékekkel!</p> <p>b) Írassa ki a tömb elemeinek értékét egymás mellé tabulátorral elválasztva!</p>	Feladatértelmezés	<p>Projektor</p> <p>Hálózat</p> <p>Egyéni munka</p>
5 – 6'	<p>Ne feledkezzen meg kiírni a felhasználónak, hogy írjon be számot! Azt is közölje mindig a felhasználóval, hogy hányadik elemet kéri be!</p> <p>A felhasználó számára első elem a programban milyen indexű?</p> <p><i>Válasz: 0</i></p> <p>Hogy hivatkozunk a tömb egy elemére?</p> <p><i>Válasz: A tömb neve és szögletes zárójelben az elem indexe.</i></p>	<p>Figyelem felhívása</p> <p>Megerősítés</p>	<p>Frontális munka</p> <p>Elbeszélés</p> <p>Megbeszélés</p>
6 – 13'	Készítse el a struktogramot, használja hozzá a legoptimálisabb programot!	<p>Feladat kitűzése</p> <p>Alkalmazásválasztás</p>	<p>Microsoft Office Excel 2007</p> <p>Structorizer</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
6 – 13'		<p>Algoritmus alkotása Struktogram készítés Precizitás, pontosság</p> <p>Aktivizálás Gyakorlás</p>	<p>Egyéni munka</p> <p>Figyelemmel követem munkájukat.</p>
13 – 15'	<p>Kérem, ellenőrizték a kivetített struktogram alapján, hogy jól dolgoztak-e!</p> <p>Ha nem tudják eldönteni jó-e amit elkészítettek, szóljanak, segítek.</p> <p>Az esetleges hibákat javítsák!</p>	<p>Önellenőrzés Diagnosztizálás</p>	<p>Projektor</p>
15 – 16'	<p>Kódolás előkészítése:</p> <p>Mit kell tennünk, hogy a felhasználó adatokat tudjon bevinni a programba a billentyűzeten keresztül?</p> <p><i>Válasz: Billentyűzetadatok beolvasására a <u>Scanner</u> osztály szolgál. Létre kell hoznunk egy Scanner típusú objektumot. Az osztálynak tudnia kell, hogy honnan jönnek az adatok. Billentyűzet esetén ez a <u>System.in</u>.</i></p>	<p>Ismétlés</p>	<p>Megbeszélés</p>

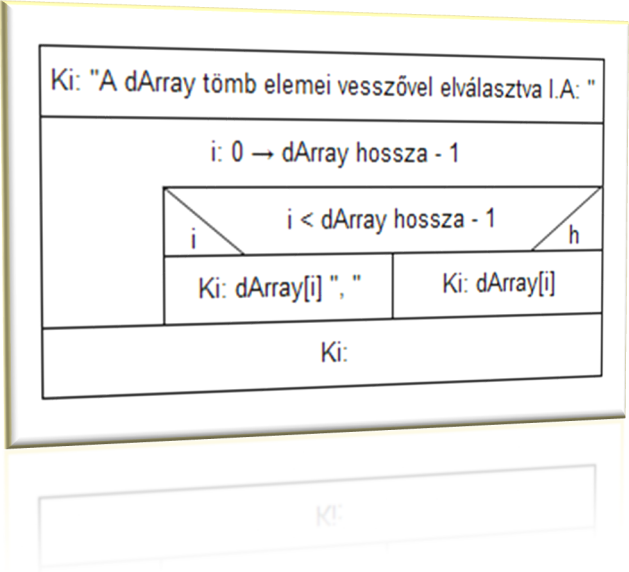
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
15 – 16'	<p>Mi a Java alapértelmezett csomagja? Melyik csomagban van a Scanner osztály?</p> <p><i>Válasz: A lang az alapértelmezett csomag, Scanner osztály a util csomagban van, ezért importálni kell.</i></p> <p>Melyik metódus szolgál double típusú számok beolvasására?</p> <p><i>Válasz: A <u>nextDouble()</u> metódus szolgál double típusú számok beolvasására.</i></p> <p>Számok beolvasása esetén mire kell figyelnünk?</p> <p><i>Válasz: A megfelelő metódus nem olvassa ki az Entert, ezért a <u>nextLine()</u> metódus segítségével erről gondoskodnunk kell.</i></p>	<p>A szükséges előzetes ismeretek felidézése a feladat megoldásához.</p>	<p>Beszélgetés</p>
16 – 17'	<p>Kódoljon a struktogram alapján!</p> <pre> package gyakorlat13;  import java.util.Scanner;  /**  *  * @author Kovács Henriette  */ public class Gyakorlat13_4 {     public static void main(String[] args) {          // 1. feladat          double[] dArray = new double[6];          Scanner billentyuzet = new Scanner(System.in); </pre>	<p>Programkód írása</p> <p>Precizitás, pontosság</p>	<p>Egyéni munka</p> <p>Számítógép Java 1.5 NetBeans IDE 7.3</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
17 – 22'	<pre>// dArray tömb elemeinek a feltöltése billentyűzetről for (int i = 0; i &lt; dArray.length; i++) {     System.out.print("Kérem a tömb " + (i + 1) + ". elemét! ");     dArray[i] = billentyuzet.nextDouble();     billentyuzet.nextLine(); }  // dArray tömb elemeinek kiírása tabulátorral elválasztva System.out.println("A dArray tömb elemei tabulátorral elválasztva:"); for (int i = 0; i &lt; dArray.length; i++) {     System.out.print(dArray[i] + "\t"); } System.out.println();</pre>	Szintaktika követése	<div>Egyéni munka</div> <div>Számítógép Java 1.5 NetBeans IDE 7.3</div>
22 – 23'	<p>Kérem, ellenőrizték a kivetített kód alapján, hogy jól dolgoztak-e!</p> <p>Az esetleges hibákat javítsák!</p>	Önellenőrzés Diagnosztizálás	Projektor
23 – 24'	<p>2.) Feladat előkészítése I.</p> <p>Gondoljuk át a tömb elemeinek indexelését a length konstans felhasználásával! Példaként vegyünk egy 6 elemű tömböt.</p> <p>Mi a tömb utolsó elemének indexe?</p> <p><i>Válasz: A tömb utolsó elemének indexe 5.</i></p> <p>Utolsó előtti elemének indexe? 4</p> <p>Hogyan hivatkozunk ezen indexű elemekre, ha a tömb neve mondjuk dArray?</p> <p><i>Válasz: Utolsó elemre: dArray[5]</i></p> <p><i>Utolsó előtti elemre: dArray[4]</i></p>	<div>Visszaemlékezés</div> <div>Visszacsatolás</div>	<div>Megbeszélés</div> <div>Frontális munka</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
24 – 28'	<p>Hogyan határozhatjuk meg a length konstanssal ezen tömb hosszát?</p> <p>Válasz: <code>dArray.length</code></p> <p>Nézzük meg, ha általánosságban szeretnénk megfogalmazni, nem konkrét értékekkel, akkor mit mondhatunk az indexekről?</p> <p>Válasz: A tömb utolsó elemének indexe <code>dArray.length - 1</code></p> <p>A tömb utolsó előtti elemének indexe <code>dArray.length - 2</code></p> <p>Hogyan hivatkozunk ezen indexű elemekre, ha a tömb neve mondjuk <code>dArray</code>?</p> <p>Válasz: Utolsó elemre: <code>dArray[dArray.length - 1]</code></p> <p>Utolsó előtti elemre: <code>dArray[dArray.length - 2]</code></p>	<p>Absztrakt gondolkodás</p> <p>Általánosítás</p>	<p>Megbeszélés</p> <p>Frontális munka</p>
	<p>Segít a megértésben a hálózatra feltett kép, elemezzük!</p> 	<p>Szemléletesség</p>	<p>TombHossza_Index.png</p> <p>Szemléltetés</p> <p>Magyarázat</p> <p>Frontális munka</p> <p>Hálózat Projektor</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>A kép egy részletét, ami kinagyítja a lényegét, szintén feltettem a hálózatra, le tudják tölteni!</p> 	<p>Rögzítés Megerősítés</p>	<p>TombHossza_Index_resze.png</p> <p>Elbeszélés</p> <p>Frontális munka</p> <p>Hálózat</p> <p>Projektor</p>
28 – 29'	<p>Feladat kiadása: Olvassa el figyelmesen a feladatot!</p> <p>2.) A feladathoz készítsen struktogramot!</p> <p>Írassa ki a dArray tömb elemeit vesszővel elválasztva úgy, hogy az utolsó elem után ne legyen vessző! Keressen többféle megoldást!</p>	<p>Feladatértelmezés</p>	<p>Projektor Hálózat</p> <p>Egyéni munka</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
29 – 32'	<p>2.) Feladat további előkészítése II.</p> <p>Próbálja megfogalmazni lépésről lépésre, hogy mit tesz!</p> <p><i>Válasz: Kiíratom az első elemet, ha nem az utolsó elem vesszőt teszek, kiíratom a következő elemet, ha nem az utolsó elem, vesszőt teszek, ... és így tovább ... kiíratom a következő elemet, ez az utolsó elem, nem teszek vesszőt.</i></p> <p>Mikor teszek vesszőt?</p> <p><i>Válasz: Mindig teszek vesszőt, ha nem az utolsó elemnél vagyok.</i></p> <p>Milyen elemindexnél teszek vesszőt?</p> <p><i>Válasz: Akkor teszek vesszőt, ha az elemindex kisebb, mint az utolsó elem indexe.</i></p> <p>Mitől függ, hogy teszek-e vesszőt?</p> <p><i>Válasz: Egy feltételtől függ, méghozzá attól, hogy az elemindex kisebb-e, mint az utolsó elem indexe.</i></p> <p>Milyen vezérlési szerkezetet használok ennek megvalósítására?</p> <p><i>Válasz: Elágazást.</i></p> <p>Hányszor kell megvizsgálnom ezt a feltételt?</p> <p><i>Válasz: Ahány eleme van a tömbnek, annyiszor kell megvizsgálnom a feltételt.</i></p> <p>Milyen vezérlési szerkezetet használok ennek megvalósítására?</p> <p><i>Válasz: Léptetéses ciklust fogok használni.</i></p>	<p>Logikus gondolkodás</p> <p>Fokozatosság</p> <p>Algoritmikus gondolkodás</p> <p>Algoritmus tervezése</p> <p>Algoritmus alkotása</p>	<p>Megbeszélés</p> <p>Frontális munka</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
32 – 35'	Készítsék el a struktogramot, használják hozzá a leoptimálisabb programot!	Struktogram készítés Precizitás, pontosság Aktivizálás	Microsoft Office Excel 2007 Structorizer
		Algoritmus készítése Struktogram készítése	Egyéni munka
35 – 36'	Kérem, ellenőrizzék a kivetített struktogram alapján, hogy jól dolgoztak-e! Az esetleges hibákat javítsák!	Önellenőrzés Diagnosztizálás	Projektor
	A struktogram alapján kódoljanak!	Feladat kitűzése	

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
36 – 41'	<pre>// 2. feladat  // dArray tömb elemeinek kiíratása vesszővel elválasztva I.A System.out.println("A dArray tömb elemei vesszővel elválasztva I.A:");  for (int i = 0; i &lt; dArray.length; ++i) {     if (i &lt; dArray.length - 1) {         System.out.print(dArray[i] + ", ");     }     else {         System.out.print(dArray[i]);     } } System.out.println();</pre>	<div>Programkód írása</div> <div>Precizitás, pontosság</div> <div>Szintaktika követése</div>	<div>Számítógép Java 1.5 NetBeans IDE 7.3</div>
41 – 42'	<p>Kérem, ellenőrizték a kivetített kód alapján, hogy jól dolgoztak-e!</p> <p>Az esetleges hibákat javítsák!</p>	<div>Önellenőrzés Diagnosztizálás</div>	<div>Projektor</div>
42 – 44'	<p>Mivel gyarapítottuk a tudásunkat a mai órán ☺?</p> <p>Válasz: Fel tudunk tölteni tömböket a billentyűzetről és ki tudjuk írni a tömb elemeit vesszővel elválasztva.</p> <p>Ezzel az utolsó feladattal fogjuk folytatni a következő órát!</p>	<div>Rendszerezés</div> <div>Figyelem felkeltése</div>	<div>Együttműködő oktatási módszer</div> <p>A KÖVETKEZŐ ÓRA MEGTARTÁSÁRA KÖZVETLENÜL EZ UTÁN AZ ÓRA UTÁN KERÜL SOR, EZÉRT NINCS HÁZI FELADAT.</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
44 – 45'	Tanulók munkájának értékelése	Értékelés	Megbeszélés Egész órát végig kíséri az értékelés.

## Feladatok

1.) A feladathoz készítsen struktogramot!

Hozzon létre egy dArray nevű, double elemtípusú tömböt, melynek 6 eleme van!

- a) Töltse fel a tömböt a felhasználó által megadott értékekkel!
- b) Írassa ki a tömb elemeinek értékét egymás mellé tabulátorral elválasztva!

2.) A feladathoz készítsen struktogramot!

Írassa ki a dArray tömb elemeit vesszővel elválasztva úgy, hogy az utolsó elem után ne legyen vessző!

Keressen többféle megoldást!

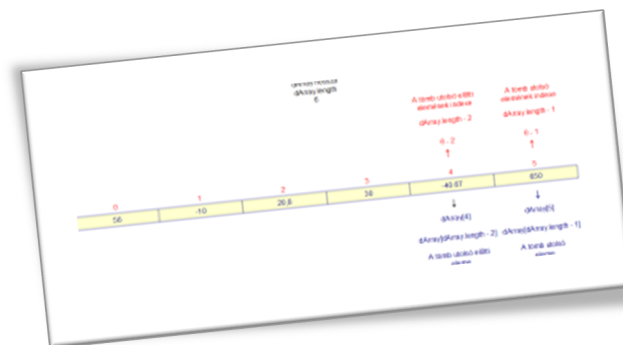
## Reflexió

A diákok mind a két feladathoz kellően motiváltak voltak, anélkül, hogy bármilyen erőfeszítést tettem volna ennek érdekében, bár megjegyzem, hogy szünetben való beszélgetésünk alkalmával felmerült már ennek igénye részükről. Egyszóval mind a billentyűzetről való feltöltés, mind a tömbelemek vesszővel elválasztva történő megjelenése érdekelte őket. Még az algoritmikus gondolkodásuk is szárnyakat kapott, holott ez azért nehézségekbe szokott ütközni. Tulajdonképpen rájöttek az egyik lehetséges algoritmusra és pont arra, amelyiket erre az órára beterveztam, de ha nem így lett volna, akkor pusztán felcserélődtek volna a lehetséges sorrendek. A pontos megfogalmazásban segítenem kellett, arra mindig különös gondot fordítok, hogy világosan, precízen és pontosan fogalmazzanak (egyik diáknál ennek organikus oka van, vele a kommunikáció nagyon nehézkes élő szó formájában), a lényeg, hogy erre törekszem, ezáltal bennük is próbálom kialakítani az ez iránti igényt.

Nagyon fontos a programozás oktatása során az absztrakt gondolkodás fejlesztése, ezt a célt szolgálta a tömbelemek indexeinek, illetve a tömbelemek elérésének általános formában való megjelenítése, ami szükséges a következő megoldás során, úgymint:

<code>dArray.length - 1</code>	<code>dArray.length - 2</code>	<code>dArray[dArray.length - 1]</code>	<code>dArray[dArray.length - 2]</code>
--------------------------------	--------------------------------	--	--

Ez nehézségekbe ütközött és ezért is örültem nagyon, hogy készítettem egy képet, ami mindezt szemléletesen megmutatta nekik és ez sokat is segített.



TombHossza\_Index.png

Mindezzel a következő órát is előkészítettem. Ennek megértése nagyon fontos a tömbökkel, illetve az indexekkel való foglalkozás során.

## **ÓRATERVEZET - TÖMB ELEMEINEK KIÍRATÁSA VESSZŐVEL ELVÁLASZTVA. INICIALIZÁLÓ BLOKK.**

**A pedagógus neve:** Kovács Henriette

**A szakmacsoport száma és megnevezése:** 7. Informatika

**A szakképesítés azonosító száma:** 54 213 05

**A szakképesítés megnevezése:** Szoftverfejlesztő

**Iskolai rendszerű szakképzés** (a szakképzési évfolyamok száma: 2)

**Szakmai követelménymodul:** 10817-12 Hálózatok, programozás és adatbázis-kezelés

**Tantárgy:** Adatbázis- és szoftverfejlesztés gyakorlat

**Programozási nyelv:** Java

**Fejlesztői környezet:** NetBeans IDE 7.3

**Osztály - csoport:** I/13. S - SZ2

**Az óra tananyaga:** Tömb elemeinek kiíratása vesszővel elválasztva. Inicializáló blokk.

## Az óra cél- és feladatrendszere

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonomiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt. Kialakítani a tanulóknál az informatikai ismereteik folyamatos bővítésének és megújításának igényét.
- Korszerű alkalmazói készség kialakítása: meglévő, - Internet segítségével - új alkalmazások kiválasztása a feladat tervezéséhez és legyenek képesek arra, hogy mindezeket célszerűen használják.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.

- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Legyen képes az algoritmust megérteni, felismerni, elemezni, végrehajtani, alkotni, megvalósítani, módosítani, átalakítani, tervezni.*
- *Tudjon tömb típusú változót deklarálni, létrehozni, feltölteni.*
- *Legyen képes a tömbök elemeit elérni.*
- *Tudjon tömböt többféle módon és többféle algoritmussal megjeleníteni.*
- *Tudjon tömböt létrehozni inicializáló blokk segítségével.*
- *Ismerje a hatékonyság-vizsgálat jelentőségét.*
- *Tudja a feltételvizsgálatok, létrejött objektumok számát meghatározni.*

### **Fejlesztendő készségek, képességek**

- Visszaemlékezés, kommunikációs képesség, programkód átalakítása, módosítása, önellenőrzés, struktogram olvasása, megértése, struktogram elemzése, algoritmus megfogalmazása, algoritmikus gondolkodás, programkód írása, precizitás, pontosság, szintaktika követése, diagnosztizálás, hibakeresés, programkód olvasása, programkód megértése, analizálása, állománykezelés, döntésképesség, hatékonyság-vizsgálat.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.

- Ismerje fel és legyen képes különféle formákban megfogalmazni a tevékenységek algoritmizálható részleteit.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.

### **Az óra didaktikai feladatai**

- Motiválás
- A figyelem felkeltése
- A tanulók informálása a tanulási célokról.
- Aktivizálás
- Ismétlés
- Fokozatosság
- Feladat továbbfejlesztése
- Visszacsatolás
- Ismétlés
- Alkalmazás
- Megerősítés
- Feladatkitűzés
- Folyamat elemzése
- Következtetés, általánosítás
- A tények elemzése
- Új ismeret közlése
- Rögzítés
- Gyakorlás
- Rendszerezés

**Tantárgyi kapcsolatok:** Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek

**Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):**

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyne Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

<http://docs.oracle.com/javase/7/docs/api/>

Dátum: éééé. hh. nn.

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 3'	<p>Az órát bevezető kérdések:</p> <p>1. Hogyan tudunk létrehozni tömböket? new operátorral <i>meg fogunk ismerkedni még egy lehetőséggel, az úgynevezett inicializáló blokkal (mai órán)</i></p> <p>2. Hogyan tudunk feltölteni tömböket? egyesével, szekvenciával szabály alapján, vezérlési szerkezettel, léptetési ciklussal felhasználó által <i>véletlenszerűen</i></p> <p>3. Hogyan végeztük tömbök kiíratását? szóközzel, tabulátorral elválasztva vesszővel elválasztva <i>(folytatjuk)</i></p> <p>A Gyakorlat13 projekt gyakorlat13 csomagjában létrehozott Gyakorlat13_4 osztályban folytatjuk a kódolást!</p>	<p>Visszaemlékezés Kommunikációs képesség</p> <p>Motiválás A figyelem felkeltése A tanulók informálása a tanulási célokról.</p> <p>Aktivizálás Ismétlés Fokozatosság</p>	<p>Ez az óra közvetlenül kapcsolódik az előző órához. Az utolsó feladatot folytatjuk.</p> <p>Beszélgetés</p> <p>Frontális munka</p> <p>Projektor Számítógép NetBeans IDE 7.3 Java 1.5</p>

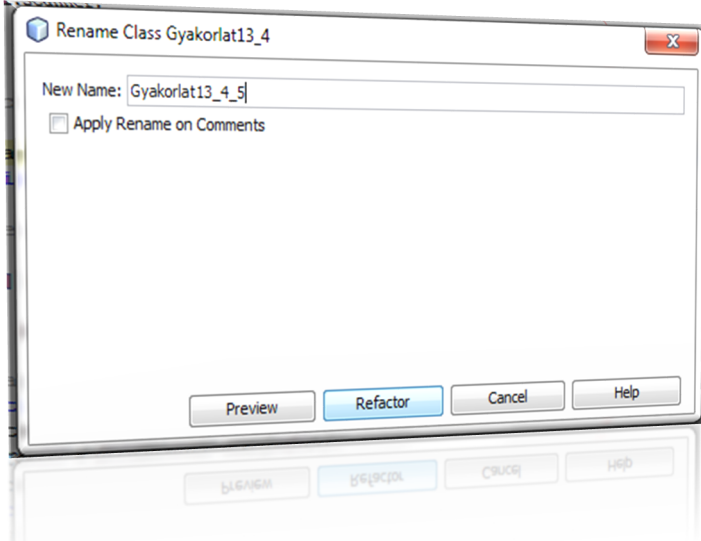
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
3 – 4'	<p>3.) Előző órai feladat folytatása.</p> <p>A feladathoz készítsen struktogramot!</p> <p>Írassa ki a dArray tömb elemeit vesszővel elválasztva úgy, hogy az utolsó elem után ne legyen vessző!</p> <p>Keressen többféle megoldást!</p>	<p>Visszacsatolás</p> <p>Feladat továbbfejlesztése</p>	<p>Projektor</p> <p>Hálózat</p>
4 – 5'	<p>Kódoljuk le egyszerűbben az elágazást! A kétágú elágazást mivel tudnánk helyettesíteni?</p> <p><i>Válasz: A feltételes kiértékelés segítségével egyszerűbb lenne a kód.</i></p> <p>☺ Mi a szintaktikája?</p> <p><i>Válasz: feltétel ? kif1 : kif2</i></p> <p>Ha a feltétel igaz, akkor kifejezés1, egyébként kifejezés2 kerül kiértékelésre.</p>	<p>Ismétlés</p> <p>Alkalmazás</p>	<p>Tábla</p> <p>Megbeszélés</p> <p>Frontális munka</p>
5 – 9'	<pre>// dArray tömb elemeinek kiíratása vesszővel elválasztva I.B System.out.println("A dArray tömb elemei vesszővel elválasztva I.B:"); for (int i = 0; i &lt; dArray.length; i++) {     System.out.print(i &lt; dArray.length - 1         ? dArray[i] + ", " : dArray[i]); } System.out.println();</pre>	<p>Programkód átalakítása, módosítása</p> <p>Megerősítés</p>	<p>Egyéni munka</p> <p>Számítógép</p> <p>NetBeans IDE 7.3</p> <p>Java 1.5</p>
9 – 10'	<p>A kivetített kód alapján ellenőrizze munkáját, ha szükséges javítsa a hibákat!</p> <p>.</p>	<p>Önellenőrzés</p> <p>Hibakeresés</p>	<p>Projektor</p>
10 – 13'	Hány feltételvizsgálat történik a kódban?		

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>6 elemű tömb esetén:</p> <div> <div> <div>0 &lt; 6</div> <div>0 &lt; 5</div> <div>1 &lt; 6</div> <div>1 &lt; 5</div> <div>2 &lt; 6</div> <div>2 &lt; 5</div> <div>3 &lt; 6</div> <div>3 &lt; 5</div> <div>4 &lt; 6</div> <div>4 &lt; 5</div> <div>5 &lt; 6</div> <div>5 &lt; 5</div> <div>6 &lt; 6</div> </div> <div> <div>6 * 2 + 1</div> </div> </div> <p>Válasz: <i>dArray.length * 2 + 1</i></p>	<div>Hatékonyság-vizsgálat</div> <div>Folyamat elemzése</div> <div>Következtetés, általánosítás</div>	<div>Magyarázat Szemléltetés</div> <div>Tábla</div> <div>Frontális munka</div>
13 – 15'	<p>Olvassa el a következő kódrészletet, elemezze, értelmezze!</p> <pre>// dArray tömb elemeinek kiíratása vesszővel elválasztva II. System.out.println("A dArray tömb elemei vesszővel elválasztva II."); String szoveg = ""; for(int i = 0; i &lt; dArray.length; i++){     szoveg += (i &lt; dArray.length - 1) ? dArray[i] + ", " : dArray[i]; } System.out.println(szoveg);</pre> <p>Hány feltételvizsgálat történik, hány String objektum keletkezik?</p>	<div>Programkód olvasása Programkód megértése, analizálása</div> <div>Hatékonyság-vizsgálat</div>	<div>Projektor Gyakorlat13_4_2_II.png</div> <div>Egyéni munka</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
15 – 16'	<p>Mi történik a kódban?</p> <p><i>Válasz: Egy üres Stringhez mindig hozzáfűzöm a tömb egy elemét és egy vesszőt, az utolsó elemnél csak az elemet.</i></p> <p><i>dArray.length * 2 + 1 feltételvizsgálat történik.</i></p> <p><i>dArray.length + 1 String objektum keletkezik.</i></p>	Kommunikációs készség	Tanulói elbeszélés
16 – 19'	Kódolja le String objektum felhasználásával a tömb vesszővel történő kiíratását!	Visszaemlékezés	Egyéni munka
19 – 20	<p>Elemezzék a következő struktogramot, és fogalmazzák meg szavakkal az algoritmust, hogyan kerülnek a tömb elemei közé a vesszők!</p> <div data-bbox="405 880 1115 1289" data-label="Diagram"> <pre> graph TD     A["Ki: "A dArray tömb elemei vesszővel elválasztva III.: ""]     B["i: 0 → dArray hossza - 2"]     C["Ki: dArray[i] ', '"]     D["Ki: dArray[dArray hossza - 1]"]     A --- B     B --- C     C --- D   </pre> </div> <p>Hány feltételvizsgálat történik?</p>	<p>Struktogram olvasása, megértése</p> <p>Struktogram elemzése</p> <p>Hatékonyság-vizsgálat</p>	<p>Projektor</p> <p>Hálózat Gyakorlat13_4_2_III_Struktogram. png</p> <p>Frontális munka</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK			
20 – 21'	<p>Emlékezzünk vissza, melyik elem indexe a <code>dArray.length – 2</code>?</p> <div><p>dArray hossza dArray.length 6</p><p>A tömb utolsó előtti elemének indexe      A tömb utolsó elemének indexe</p><p>dArray.length - 2      dArray.length - 1</p><p>6 - 2      6 - 1</p><p>↑      ↑</p><p>4      5</p><table><tr><td></td><td>-40.67</td><td>650</td></tr></table></div> <p>Válasz: A tömb utolsó előtti elemének indexe.</p>		-40.67	650	Visszacsatolás	<div>Szemléltetés</div> <div>Projektor</div> <div>Beszélgetés</div>
	-40.67	650				
21 – 23'	<p>Struktogram elemzése</p> <p>Válasz: Algoritmus: A tömb első elemétől kezdve végigmegyek a tömb elemein, de csak az utolsó előtti elemig. Mindig azt csinálom, hogy kiíratom az elemet és egy vesszőt.</p> <p>Aztán kiíratom a tömb utolsó elemét!</p> <p>Válasz: <code>dArray.length</code> feltételvizsgálat történik.</p>	<div>Algoritmikus gondolkodás</div> <div>Algoritmus megfogalmazása</div>	<div>Tanulói elbeszélés</div>			
	A struktogram alapján kódoljanak!	Feladatkitűzés				

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
23 – 26'	<pre>// dArray tömb elemeinek kiírása vesszővel elválasztva III. System.out.println("A dArray tömb elemei vesszővel elválasztva III.:"); for (int i = 0; i &lt; dArray.length - 1; ++i)     System.out.print(dArray[i] + ", "); System.out.println(dArray[dArray.length - 1]);</pre>	Programkód írása Precizitás, pontosság Szintaktika követése	Egyéni munka Számítógép NetBeans IDE 7.3 Java 1.5
26 – 27'	Ellenőrizze munkáját!	Önellenőrzés Diagnosztizálás, hibake- resés	Projektor
27 – 28'	A 3 kiírás közül melyikben történik a legkevesebb feltételvizsgálat? <i>Válasz: III-ban</i> Melyik kiíratást tartja a legkönnyebnek? ☺ Hatékonyság problémája.	A tények elemzése	Frontális munka Megbeszélés
28 – 29'	Óra elején már említettem, hogy a new operátor nélkül is létre lehet hozni méghozzá sokkal egyszerűbben tömböket. Nézzük meg, hogyan!	A figyelem felkeltése	
29 – 30'	Deklaráláskor a tömb elemeinek kezdeti értéket adhatunk úgynevezett inicializáló blokkal. Szintaktikája: <code>&lt;elemtípus&gt;[] &lt;tömbAzonosító&gt; = {&lt;érték0&gt;, &lt;érték1&gt;,...};</code> A kapcsos zárójelek által határolt blokkot inicializáló blokknak nevezzük. Ebben az esetben a tömb mérete pontosan akkora lesz, amennyi a felsorolt értékek száma. <pre>int[] iArray = {0, 20, 40, 60, 100};</pre>	Új ismeret közlése	Elbeszélés Tábla

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
30 – 31'	<p>Gyakorlat13_4 .java osztályt nevezze át Gyakorlat13_4_5.java osztályra!</p>  <p>Hozzon létre egy Gyakorlat13_5.java osztályt és a következő feladatot ebben végezze el!</p>	<p>Állománykezelés</p>	<p>Irányított, önálló munka</p> <p>Projektor</p> <p>NetBeans IDE 7.3 Számítógép</p>
30 – 31'	<p>Feladat:</p> <p>Hozzuk létre az iArray nevű int elemtípusú tömböt a táblára felírt elemekkel: 0, 20, 40, 60, 100! Írassa ki ez elemeket egymás mellé, vesszővel elválasztva!</p>	<p>Feladatkitűzés</p>	<p>Elbeszélés</p>

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
31 – 34'	<pre>package gyakorlat13;  /**  *  * @author Kovács Henriette  */ public class Gyakorlat13_5 {     public static void main(String[] args) {          // 1. feladat         // iArray tömb létrehozása inicializáló blokkal         int[] iArray = {0, 20, 40, 60, 100};          // az iArray tömb elemeinek kiírása         System.out.println("Az iArray tömb elemei vesszővel elválasztva:");         for (int i = 0; i &lt; iArray.length - 1; ++i)             System.out.print(iArray[i] + ", ");         System.out.println(iArray[iArray.length - 1]);     } }</pre>	<div>Rögzítés</div> <div>Döntésképesség</div>	<div>Projektor</div> <div>NetBeans IDE 7.3</div> <div>Magyarázat</div> <div>Frontális munka</div> <div>Egyéni munka</div>
34 – 38'	<p>Feladat</p> <p>2.) Az inicializáló blokk felhasználásával hozzon létre</p> <p>a) egy mgh nevű tömböt, mely az angol ábécé magánhangzóit tartalmazza!</p> <p>Írassa ki a tömb elemeit vesszővel elválasztva!</p> <p>b) egy szep nevű, logikai elemtípusú tömböt, mely a lehetséges logikai értékeket tartalmazza!</p> <p>Írassa ki a tömb elemeit vesszővel elválasztva!</p>	<div>Gyakorlás</div>	<div>Differenciálás</div> <p>ketten az a), egy diák a b) feladatot készíti el</p> <div>Egyéni munka</div> <div>NetBeans IDE 7.3</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
38 – 39'	<p>Ellenőrizték</p> <pre> char[] mgh = {'a', 'e', 'i', 'o', 'u'};  // az mgh tömb elemeinek kiírása System.out.println("Az mgh tömb elemei vesszővel elválasztva:"); for (int i = 0; i &lt; mgh.length - 1; ++i)     System.out.print(mgh[i] + ", "); System.out.println(mgh[mgh.length - 1]); </pre>	Önellenőrzés	<p>Egyéni munka</p> <p>Angol ábécé ismerete szükséges</p> <p>Projektor</p>
38 – 39'	<pre> boolean[] szep = {true, false};  // a szep tömb elemeinek kiírása System.out.println("A szep tömb elemei vesszővel elválasztva:"); for (int i = 0; i &lt; szep.length - 1; ++i)     System.out.print(szep[i] + ", "); System.out.println(szep[szep.length - 1]); </pre>	Önellenőrzés	<p>Egyéni munka</p> <p>Projektor</p>
39 – 43'	<p>Rendszerezés:</p> <p>Azon tény megállapítása, hogy nem elegendő egy megoldás megtalálása, mindig meg kell győződnünk róla, hogy az optimális-e, hatékony-e!</p> <p>Melyik vesszővel való kiíratással fog dolgozni az elkövetkezőkben?</p> <p>Ha a tömböt inicializáló blokkal hozom létre, még mi történik a létrehozáson kívül?</p>	Rendszerezés	<p>Vita</p> <p>A KÖVETKEZŐ ÓRA MEGTARTÁSÁRA KÖZVETLENÜL EZ UTÁN AZ ÓRA UTÁN KERÜL SOR, EZÉRT NINCS HÁZI FELADAT.</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	Válasz: Inicializáló blokkal való létrehozáskor a tömböt fel is töltöm elemekkel.		<div>Megbeszélés</div> <div>Frontális munka</div>
43 – 45'	Tanulók munkájának értékelése	Értékelés	<div>Megbeszélés</div> <p>Egész órát végig kíséri az értékelés.</p>

# TÁBLAKÉP 1.

Feltételes kiértékelés

`feltétel ? kif1 : kif2`

Feltételvizsgálatok száma I., II.:

0 < 6	}	$6 * 2 + 1$ <code>dArray.length * 2 + 1</code>
0 < 5		
1 < 6		
1 < 5		
2 < 6		
2 < 5		
3 < 6		
3 < 5		
4 < 6		
4 < 5		
5 < 6		
5 < 5		
6 < 6		

Feltételvizsgálatok száma III.:

0 < 5	}	6 <code>dArray.length</code>
1 < 5		
2 < 5		
3 < 5		
4 < 5		
5 < 5		

## TÁBLAKÉP 2.

Inicializáló blokk

```
<elemtípus>[] <tömbAzonosító> = {<érték0>, <érték1>,...};
```

```
int[] iArray = {0, 20, 40, 60, 100};
```

## Feladatok

1.) Előző órai feladat folytatása.

Írassa ki a dArray tömb elemeit vesszővel elválasztva úgy, hogy az utolsó elem után ne legyen vessző!

Keressen többféle megoldást!

2.) Az inicializáló blokk felhasználásával hozzon létre

a) egy mgh nevű tömböt, mely az angol ábécé magánhangzóit tartalmazza!

Írassa ki a tömb elemeit vesszővel elválasztva!

b) egy szep nevű, logikai elemtípusú tömböt, mely a lehetséges logikai értékeket tartalmazza!

## Reflexió

Ezt az órát úgy terveztem meg, hogy e tantárgy szempontjából az egyik legfontosabb informatikai kompetencia komponenssel, az algoritmikus gondolkodással és annak fejlesztésével foglalkoztam. Mindennapjainkban, mind a tanulásban, mind a munkában lépten-nyomon algoritmusokat hajtunk végre, mégis e tevékenységsorozatok megfogalmazása gondot okoz. Ezen probléma kialakulásához több tényező is hozzájárul, mint például azon egyszerű ok, hogy az emberek nem beszélgetnek egymással, keveset olvasnak, nem kommunikálnak, elmennek egymás mellett, magukba fordulnak. Ezt figyelembe véve, óráimon mindig törekszem arra, hogy diákjaimat beszéltessem, hogy kommunikáljunk, mindezzel hozzájárulva ahhoz, hogy szakmai nyelven is pontosan, precízen fogalmazzák meg észrevételeiket és így közvetett módon is fejlesszem algoritmikus gondolkodásukat.

A vesszővel való kiíratás feladatkörét tekintve, a különböző megközelítésű részfeladatok egy sor, az algoritmikus gondolkodás kompetencia szintjeit képező összetevőket fejleszt. Ezek: az algoritmus megértése, azaz megérteni, hogy mit kell tenni és megérteni, hogy miért azt kell tenni, az algoritmus felismerése, az algoritmus elemzése, az algoritmus végrehajtása, az algoritmus alkotása, az algoritmus megvalósítása, az algoritmus módosítása, átalakítása, az algoritmus tervezése. Láthatjuk,

hogy ez az első ránézésre banálisnak tűnő feladat is milyen fejlesztési lehetőségeket rejt magában. A diákok emellett élvezték, hogy mindig más típusú feladatokat kellett végrehajtaniuk, hol algoritmust kitalálni, hol megérteni, hol átalakítani, hol újabb lehetőségeket keresni, így érdeklődésüket az egész feladat során fenn tudtam tartani. Azt tapasztaltam, hogy az is nagyon jó motiváló erő számukra, hogy hol alkalmazói, hol fejlesztői környezetben dolgoznak.

Az óra végére egy új ismereteket feldolgozó egységet választottam, így sor került új ismeretek elsajátítására is. Sokat töprengtem, hogy vajon jó gondolat-e, jó ötlet-e az óra végén bevezetni egy új fogalmat. Az előző órai feladatot mindenképpen folyamatában szerettem volna, hogy lássák a diákok (hat óra egy napon, egymás után van megtartva minden héten), így a kérdés csak az volt, hogy az óra végére, vagy a következő óra elejére tegyem, és az órát egy gyakorló feladattal zárom. Ez az új anyag viszont annyira egyszerű, úgy éreztem inkább feldobná az órát, új erőt kapnának a diákok, és gondoltam kipróbálom. Megérzésem valóra vált, örültek az új tömbfeltöltési lehetőségnek és a differenciálás mindenképpen a feladat megoldására sarkallta őket.

Ennek az órának a felépítéséért tett lépések sorozata megérte a fáradozást, a diákok élveztek a változatosságot.

## **ÓRATERVEZET - VÉLETLENSZÁMOK GENERÁLÁSA. TÖMB ELEMEINEK FELTÖLTÉSE VÉLETLENSZÁMOKKAL.**

**A pedagógus neve:** Kovács Henriette

**A szakmacsoport száma és megnevezése:** 7. Informatika

**A szakképesítés azonosító száma:** 54 213 05

**A szakképesítés megnevezése:** Szoftverfejlesztő

**Iskolai rendszerű szakképzés** (a szakképzési évfolyamok száma: 2)

**Szakmai követelménymodul:** 10817-12 Hálózatok, programozás és adatbázis-kezelés

**Tantárgy:** Adatbázis- és szoftverfejlesztés gyakorlat

**Programozási nyelv:** Java

**Fejlesztői környezet:** NetBeans IDE 7.3

**Osztály - csoport:** I/13. S - SZ2

**Az óra tananyaga:** Véletlenszámok generálása. Tömb elemeinek feltöltése véletlenszámokkal.

## Az óra cél- és feladatrendszere

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonomiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt.
- Korszerű alkalmazói készség kialakítása.
- Az algoritmikus gondolkodás fejlesztése.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudja tömb elemszámát bekérni.*
- *Tudjon tömb típusú változót deklarálni, létrehozni, feltölteni.*
- *Legyen képes a tömbök elemeit elérni.*
- *Tudjon tömböt megjeleníteni.*
- *Ismerje az üres tömb és üres String fogalmakat.*
- *Tudja mi a különbség a Math osztály random() metódusa által adott és a Random osztály metódusai által generált számok között.*
- *Ismerje az álvéletlen szám és véletlenszám fogalmakat.*
- *Tudjon intervallumot bővíteni és eltolni.*
- *Tudjon véletlenszámot előállítani.*
- *Tudja tömb elemeit véletlenszámmal feltölteni.*
- *Tudja mikor és hogyan kell egy osztályt importálni.*
- *Tudjon tesztelni, diagnosztizálni.*
- *Tudja a futási eredményt értelmezni.*
- *Ismerje a következő kivételeket: ArrayIndexOutOfBoundsException, NegativeArraySizeException, InputMismatchException.*

### **Fejlesztendő készségek, képességek**

Visszaemlékezés, kommunikációs képesség, rendszerező képesség, szakmai nyelvű beszédképesség, feladatértelmezés, állománykezelés, hallott szakmai szöveg megértése, programkód írása, szintaktika követése, precizitás, pontosság, futási eredmény analizálása, algoritmikus gondolkodás, algoritmus tervezése, struktogram olvasása, értelmezése, önellenőrzés, tesztelés, diagnosztizálás, hibakeresés, olvasott angol nyelvű szakmai szöveg megértése, algoritmus alkotása.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.

- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.
- Legyen képes eligazodni és tudjon keresni a Java hivatalos dokumentációjában.

### **Az óra didaktikai feladatai**

- Ismétlés
- Rendszerezés
- Motiválás
- A figyelem felkeltése
- Tanulók informálása a tanulási célokról
- Feladatkitűzés
- Visszacsatolás
- Összehasonlítás
- Aktivizálás
- Fogalomalkotás
- Gyakorlás
- Következtetés
- Információ elemzése
- A tények elemzése
- Információ gyűjtése
- Új ismereteket megalapozó tények biztosítása
- Alkalmazás
- Fokozatosság
- Összefoglalás
- A házi feladat kijelölése
- Értékelés

**Tantárgyi kapcsolatok:** Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek

**Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):**

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyiné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

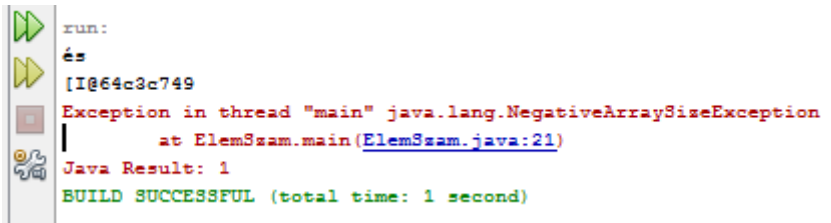
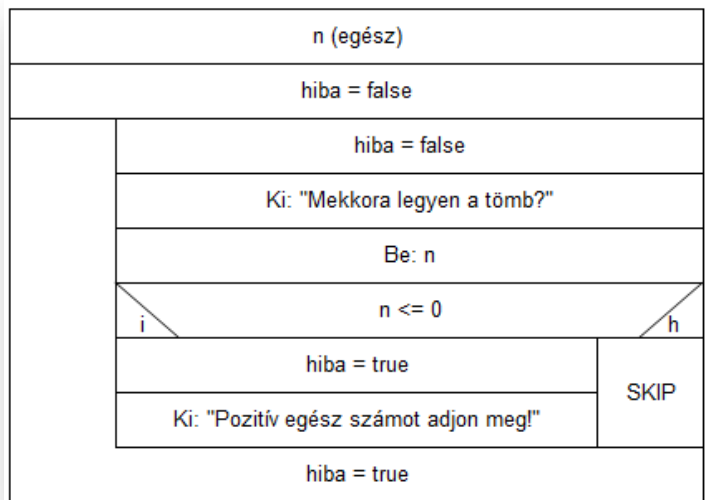
<http://docs.oracle.com/javase/7/docs/api/>

Dátum: éééé. hh. nn.

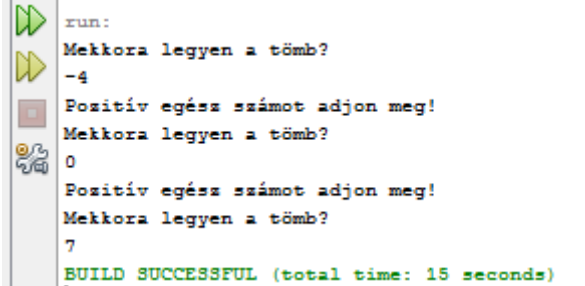
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 3'	<p>Bevezetés:</p> <p>Tömbök deklarálása</p> <p>Tömbök létrehozása</p> <p>Tömbök mérete → mai órán: 0 esetén mi jön létre és negatív szám esetén mi történik?</p> <p>Tömbök feltöltése → mai órán: véletlenszámokkal és nem álvéletlen számokkal → új osztály megismerése</p> <p>Tömbök kiírása</p> <p>Tömbelemek indexe</p> <p>Tömbelemek elérése</p> <p>ArrayIndexOutOfBoundsException</p>	<p>Visszaemlékezés</p> <p>Kommunikációs képesség</p> <p>Rendszerező képesség</p> <p>Szakmai nyelvű beszédképesség</p> <p>Ismétlés</p> <p>Rendszerezés</p> <p>Motiválás</p> <p>A figyelem felkeltése</p> <p>Tanulók informálása a tanulási célokról</p>	<p>Számítógép</p> <p>Bekapcsoltatom a gépeket.</p> <p>Megbeszélés</p> <p>Frontális munka</p>

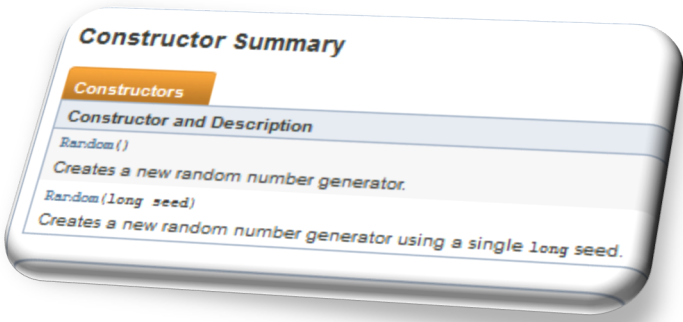
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
3 – 6'	<p>Feladat:</p> <p>1.) Írjon programot,</p> <p>a) melyben létrehoz egy egész számokból álló, a felhasználó által megadott elemszámú tömböt,</p> <p>b) feltölti a tömb elemeit olyan véletlenszámokkal, melyek a [-6, 16] zárt intervallumba esnek,</p> <p>c) kiírja a tömböt úgy, hogy elemei egymástól a ' ' karakterrel legyenek elválasztva, de az utolsó elem után ne legyen ' ' karakter!</p> <p>Létrehozni saját hálózati meghajtón:</p> <p>Gyakorlat13 projekt/gyakorlat13 csomag/Gyakorlat13_6.java</p> <p>Gyakorlat13 projekt/alapértelmezett csomag/ElemSzam.java</p> <p>Az a) feladat kapcsán:</p> <p>Felhasználó 0-t ad meg:</p> <p>Üres String</p> <p>Üres tömb is létezik Java-ban.</p>	<p>Feladatkitűzés</p> <p>Feladatértelmezés</p> <p>Állománykezelés</p> <p>Visszacsatolás Összehasonlítás</p>	<p>Hálózat Projektor</p> <p>Számítógép NetBeans IDE 7.3 Java 1.5</p> <p>Egyéni munka</p> <p>Magyarázat</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
6 – 9'	<p>Kód:</p> <pre> public class ElemSzam {     public static void main(String[] args) {          String str1 = new String(); // vagy         String str2 = ""; // üres String-ek         System.out.println(str1 + "és" + str2); // str1, str2 léteznek          int[] array = new int[0]; // üres tömb         System.out.println(array); // array tömb létezik, ki tudom íratni      } } </pre>	<p>Feladat kitűzése Hallott szakmai szöveg megértése</p> <p>Programkód írása Szintaktika követése Precizitás, pontosság</p> <p>Aktivizálás</p>	<p>Frontális munka</p> <p>Velem párhuzamosan mindenki önállóan dolgozik, a feladatokat élő szóban fogalmazom meg.</p> <p>Projektor Számítógép NetBeans IDE 7.3 Java 1.5</p>
9 – 10'	<p>Futási eredmény:</p> 	<p>Futási eredmény analízisa</p>	<p>Magyarázat</p>
10 – 11'	<p>Felhasználó negatív számot ad meg:</p> <p><b>NegativeArraySizeException</b> váltódik ki.</p> <p>Kód:</p> <pre> int[] array1 = new int[-8]; System.out.println(array1); </pre>	<p>Hallott szakmai szöveg megértése</p>	<p>Számítógép NetBeans IDE 7.3</p> <p>Frontális munka</p>

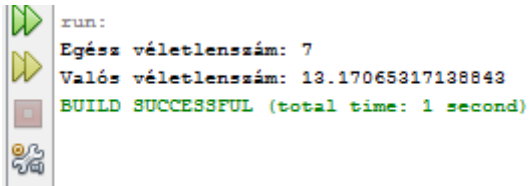
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
11 – 12'	<p>Futási eredmény:</p> 	<p>Futási eredmény analízálása</p> <p>Fogalomalkotás</p>	<p>Magyarázat</p>
12 – 15'	<p>Az a) feladat tervezése:</p> 	<p>Algoritmikus gondolkodás Algoritmus tervezése</p> <p>Visszacsatolás</p> <p>Struktogram olvasása, értelmezése</p> <p>Gyakorlás</p>	<p>Megbeszélés</p> <p>Frontális munka</p> <p>Struktogramot ez után megkapják és ez alapján kódolnak</p> <p>Hálózat 1a_Struktogram_06_or.png</p> <p>Egyéni munka</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
15 – 20'	<p>Az a) feladat kódja:</p> <pre> package gyakorlat13;  import java.util.Scanner;  public class Gyakorlat13_6 {     public static void main(String[] args) {          // 1.a) feladat         Scanner billentyuzet = new Scanner(System.in);          int n;         boolean hiba = false;          do{             hiba = false;             System.out.println("Mekkora legyen a tömb?");             n = billentyuzet.nextInt();             billentyuzet.nextLine();             if(n &lt;= 0) {                 hiba = true;                 System.out.println("Pozitív egész számot adjon meg!");             }         } while (hiba);      } } </pre>	<div>Programkód írása Szintaktika követése Precizitás, pontosság</div> <div>Önellenőrzés</div>	<div>Egyéni munka</div> <div>Számítógép NetBeans IDE 7.3 Java 1.5</div> <p>Követem munkájukat, ha szükséges segíték</p>
20 – 21'	<p>Az eddigi forráskód tesztelése:</p> <p>negatív egész számmal, 0-val és pozitív egész számmal</p> <p>valós szám esetén: <b>InputMismatchException</b> váltódik ki.</p>	<div>Tesztelés</div>	<div>Egyéni munka</div>

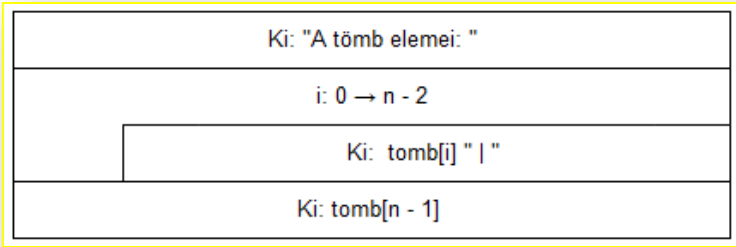
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
21 – 22'	 <pre> run: Mekkora legyen a tömb? -4 Positív egész számot adjon meg! Mekkora legyen a tömb? 0 Positív egész számot adjon meg! Mekkora legyen a tömb? 7 BUILD SUCCESSFUL (total time: 15 seconds) </pre>	<div>Diagnosztizálás, hiba-keresés</div>	<div>Egyéni munka</div>
22 – 23'	<p>Az b) feladat kapcsán:</p> <p>Eddig álvéletlen számokkal foglalkoztunk:</p> <p>Math osztály:</p> <p style="padding-left: 40px;">random() metódus: [0,0; 1,0[ intervallumban álvéletlen számot ad.</p> <p>De a Random osztály:</p> <p style="padding-left: 40px;">véletlenszámok előállítására szolgáló osztály</p> <p><a href="http://docs.oracle.com/javase/7/docs/api/java/util/Random.html">http://docs.oracle.com/javase/7/docs/api/java/util/Random.html</a></p> <p>util csomagban van → importálni kell</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px; background-color: #f9f9f9;"> <div>java.util</div> <div><b>Class Random</b></div> <div>java.lang.Object</div> <div>java.util.Random</div> </div>	<div>Fokozatosság</div> <div>Visszacsatolás Fogalomalkotás</div> <div>Összehasonlítás</div> <div>Információ gyűjtése Következtetés</div>	<div>Frontális munka</div> <div>Tanulói elbeszélés</div> <div>Elbeszélés, közlés</div> <div>Internet</div> <p><a href="http://docs.oracle.com/javase/7/docs/api/java/util/Random.html">http://docs.oracle.com/javase/7/docs/api/java/util/Random.html</a></p>

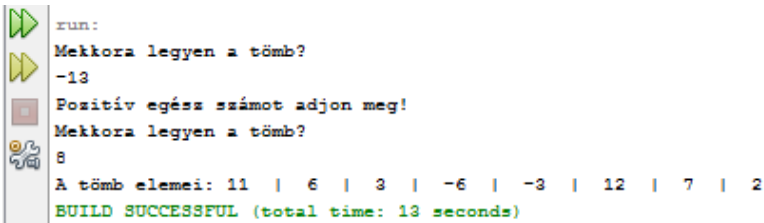
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
23 – 25'	<p>Random() konstruktor: készít egy új véletlenszám-generátort.</p> <p>Konstruktorok:</p>  <p>nextInt() metódus: A teljes int értéktartományban generál egy int típusú értéket.</p> <p>nextLong() metódus: A teljes long értéktartományban generál egy long típusú értéket.</p> <p>nextInt(int n) metódus: A [0; n[ intervallumban generál egy int típusú értéket.</p> <p>nextDouble() metódus: A [0,0; 1,0] intervallumban generál egy double típusú értéket.</p> <p>nextBoolean() metódus: Generál egy logikai értéket.</p>	<p>Információ elemzése</p> <p>A tények elemzése</p> <p>Információ gyűjtése</p> <p>Új ismereteket megalapozó tények biztosítása</p>	<p>Beszélgetés</p> <p>Frontális munka</p> <p>Internet</p> <p><a href="http://docs.oracle.com/javase/7/docs/api/java/util/Random.html">http://docs.oracle.com/javase/7/docs/api/java/util/Random.html</a></p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK														
	<p>Metódusok:</p> <div><div>Method Summary</div><div><div>Methods</div><table><thead><tr><th>Modifier and Type</th><th>Method and Description</th></tr></thead><tbody><tr><td>protected int</td><td><code>next(int bits)</code> Generates the next pseudorandom number.</td></tr><tr><td>boolean</td><td><code>nextBoolean()</code> Returns the next pseudorandom, uniformly distributed <code>boolean</code> value from this random number generator's sequence.</td></tr><tr><td>void</td><td><code>nextBytes(byte[] bytes)</code> Generates random bytes and places them into a user-supplied byte array.</td></tr><tr><td>double</td><td><code>nextDouble()</code> Returns the next pseudorandom, uniformly distributed <code>double</code> value between 0.0 and 1.0 from this random number generator's sequence.</td></tr><tr><td>int</td><td><code>nextInt()</code> Returns the next pseudorandom, uniformly distributed <code>int</code> value from this random number generator's sequence.</td></tr><tr><td>int</td><td><code>nextInt(int n)</code> Returns a pseudorandom, uniformly distributed <code>int</code> value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.</td></tr></tbody></table></div></div>	Modifier and Type	Method and Description	protected int	<code>next(int bits)</code> Generates the next pseudorandom number.	boolean	<code>nextBoolean()</code> Returns the next pseudorandom, uniformly distributed <code>boolean</code> value from this random number generator's sequence.	void	<code>nextBytes(byte[] bytes)</code> Generates random bytes and places them into a user-supplied byte array.	double	<code>nextDouble()</code> Returns the next pseudorandom, uniformly distributed <code>double</code> value between 0.0 and 1.0 from this random number generator's sequence.	int	<code>nextInt()</code> Returns the next pseudorandom, uniformly distributed <code>int</code> value from this random number generator's sequence.	int	<code>nextInt(int n)</code> Returns a pseudorandom, uniformly distributed <code>int</code> value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.	<div>Olvasott angol nyelvű szakmai szöveg megértése</div>	<div>Frontális munka</div> <div>Internet</div> <div><a href="http://docs.oracle.com/javase/7/docs/api/java/util/Random.html">http://docs.oracle.com/javase/7/docs/api/java/util/Random.html</a></div> <div>Beszélgetés</div>
Modifier and Type	Method and Description																
protected int	<code>next(int bits)</code> Generates the next pseudorandom number.																
boolean	<code>nextBoolean()</code> Returns the next pseudorandom, uniformly distributed <code>boolean</code> value from this random number generator's sequence.																
void	<code>nextBytes(byte[] bytes)</code> Generates random bytes and places them into a user-supplied byte array.																
double	<code>nextDouble()</code> Returns the next pseudorandom, uniformly distributed <code>double</code> value between 0.0 and 1.0 from this random number generator's sequence.																
int	<code>nextInt()</code> Returns the next pseudorandom, uniformly distributed <code>int</code> value from this random number generator's sequence.																
int	<code>nextInt(int n)</code> Returns a pseudorandom, uniformly distributed <code>int</code> value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.																
25 – 28'	<p>Létrehozni saját hálózati meghajtón:</p> <p>Gyakorlat13 projekt/alapértelmezett csomag/VeletlenSzam.java</p> <p>Random objektum létrehozása</p> <p>Random osztály importálása</p> <p><code>[-6;16]    16 - (-6) = 22</code></p> <p><code>nextInt(23) - 6</code></p> <p><code>nextDouble() * 22 - 6</code></p>	<div>Állománykezelés</div> <div>Alkalmazás</div>	<div>Hálózat</div> <div>Frontális munka</div> <div>Magyarázat</div> <div>Tábla</div>														

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
28 – 30'	<p>Kód:</p> <pre> import java.util.Random;  /**  *  * @author Kovács Henriette  */ public class VeletlenSzam {     public static void main(String[] args) {         Random generator = new Random();          // Egész véletlenszám generálása a [-6;16] intervallumban         int iszam = generator.nextInt(23) - 6;         System.out.println("Egész véletlenszám: " + iszam);          // Valós véletlenszám generálása a [-6;16] intervallumban         double dszam = generator.nextDouble() * 22 - 6;         System.out.println("Valós véletlenszám: " + dszam);     } } </pre>	<div>Programkód írása Szintaktika követése Precizitás, pontosság</div> <div>Önellenőrzés</div>	<div>Egyéni munka</div> <div>Számítógép NetBeans IDE 7.3 Java 1.5</div>
	<p>Futási eredmény:</p> 	<div>Diagnosztizálás</div>	<div>Egyéni munka</div> <div>Számítógép NetBeans IDE 7.3 Java 1.5</div>
	A b) feladat struktogramja:		

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
30 – 31'	<div data-bbox="383 405 1124 601" data-label="Diagram"> <pre> tomb (tomb, n) i: 0 → n - 1 tomb[i] = veletlenszam [-6..16] </pre> </div> <p data-bbox="275 711 1249 799">A véletlenszámokat nem kell eltárolni külön változóba, egyből értékül lehet adni a tömb elemeinek.</p>	<div data-bbox="1335 368 1637 413" data-label="Text"> <p>Algoritmus alkotása</p> </div>	<div data-bbox="1711 368 2089 413" data-label="Text"> <p>Hálózat</p> </div> <p data-bbox="1688 424 2027 456">1b_Struktogram_06_ora.png</p> <div data-bbox="1711 533 2089 577" data-label="Text"> <p>Megbeszélés</p> </div> <div data-bbox="1711 588 2089 633" data-label="Text"> <p>Frontális munka</p> </div>
31 – 32'	<p data-bbox="275 855 907 887">Ha a kódolás során elfeledkezik az importálásról.</p> <div data-bbox="512 922 1010 1362" data-label="Image"> </div>	<div data-bbox="1335 1011 1637 1056" data-label="Text"> <p>Figyelemfelhívás</p> </div> <div data-bbox="1335 1131 1637 1206" data-label="Text"> <p>Olvasott angol nyelvű szöveg megértése</p> </div>	<div data-bbox="1711 1011 2089 1056" data-label="Text"> <p>NetBeans IDE 7.3</p> </div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
32 – 34'	<p>A b) feladat kódja:</p> <pre>Random generator = new Random();  for (int i = 0; i &lt; n; ++i) {     tomb[i] = generator.nextInt(23) - 6; }</pre>	<p>Programkód átalakítása Szintaktika követése Precizitás, pontosság</p>	<p>Egyéni munka</p>
34 – 35'	<p>A c) feladat struktogramja: (feltételvizsgálatok száma!)</p> 	<p>Visszacsatolás</p>	<p>Hálózat 1c_Struktogram_06_ora.png Megbeszélés Frontális munka</p>
35 – 38'	<p>A c) feladat kódja:</p> <pre>System.out.print("A tömb elemei: "); for (int i = 0; i &lt; n - 1; ++i) {     System.out.print(tomb[i] + "   "); } System.out.println(tomb[n - 1]); }</pre>	<p>Programkód módosítása</p> <p>Szintaktika követése Precizitás, pontosság</p>	<p>Egyéni munka</p> <p>NetBeans IDE 7.3</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>Futási eredmény:</p> 	<p>Tesztelés</p>	<p>NetBeans IDE 7.3</p> <p>Egyéni munka</p>
38 – 40'	Gyűjtsük össze címszavakat felsorolva, hogy miket tanultunk ma, ebben a hat órában	<p>Összefoglalás</p>	<p>Frontális munka</p>
40 – 42'	<p>1.) Írjon programot, melyben</p> <ol style="list-style-type: none"> <li>létrehoz egy valós számokból álló, a felhasználó által megadott elemszámú tömböt,</li> <li>feltölti a tömb elemeit olyan két tizedesjegyre kerekített véletlenszámokkal, melyek a [-50, 50] zárt intervallumba esnek,</li> <li>kiírja a tömb elemeit úgy, hogy elemei egymástól a ' * ' karakterrel legyenek elválasztva, de az utolsó elem után ne legyen ' * ' karakter,</li> <li>kommentezi a forráskódot úgy, hogy az tükrözze, melyik kódrészletben mi történik!</li> </ol>	<p>A házi feladat kijelölése</p>	<p>Hálózat</p> <p>Gyakorlat13_6_HF.pdf</p> <p>Veletlenszam_tabla.png</p> <p>Az elkészített kód lesz az alapja a következő órának, az itt elkészített kódot fogják a diákok felhasználni.</p> <p>Ezt közlöm is velük.</p> <p>Tanári közlés</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>A feladathoz készítsen <b>struktogramot</b>! Mentse <i>Gyakorlat13_6_HF_Vezeteknev_Keresztnev</i> néven!</p> <p>A forráskódot mentse <i>Gyakorlat13_6_HF_Vezeteknev_Keresztnev</i> néven!</p> <p>A program futási eredménye a következő formában jelenjen meg:</p> <p>A tömb elemei:    6.39 * 3.02 * 19.95 * -5.93 * 34.58 * 15.37</p>		
42 – 45'	Tanulók munkájának értékelése	Értékelés	<div>Megbeszélés</div> <p>Egész órát végig kíséri az értékelés.</p>

# TÁBLAKÉP

`[-6;16]`     $16 - (-6) = 22$

---

`nextInt(23)`                      `[0;23[`

`[0;22]`             $/-6$

`nextInt(23) - 6`                      `[-6;16]`

---

`nextDouble()`                      `[0;1]`             $/*22$

`nextDouble() * 22`                      `[0;22]`             $/-6$

`nextDouble() * 22 - 6`                      `[-6;16]`

## Feladatok

- 1.) Írjon programot,
  - a) melyben létrehoz egy egész számokból álló, a felhasználó által megadott elemszámú tömböt,
  - b) feltölti a tömb elemeit olyan véletlenszámokkal, melyek a  $[-6, 16]$  zárt intervallumba esnek,
  - c) kiíratja a tömböt úgy, hogy elemei egymástól a `'|'` karakterrel legyenek elválasztva, de az utolsó elem után ne legyen `'|'` karakter!

## Házi feladat

1.) Írjon programot, melyben

- létrehoz egy valós számokból álló, a felhasználó által megadott elemszámú tömböt,
- feltölti a tömb elemeit olyan két tizedesjegyre kerekített véletlenszámokkal, melyek a  $[-50, 50]$  zárt intervallumba esnek,
- kiírja a tömb elemeit úgy, hogy elemei egymástól a '❄' karakterrel legyenek elválasztva, de az utolsó elem után ne legyen '❄' karakter,
- kommentezi a forráskódot úgy, hogy az tükrözze, melyik kódrészletben mi történik!

A feladathoz készítsen **struktogramot**! Mentse *Gyakorlat13\_6\_HF\_Vezeteknev\_Keresztnev* néven!

A forráskódot mentse *Gyakorlat13\_6\_HF\_Vezeteknev\_Keresztnev* néven!

A program futási eredménye a következő formában jelenjen meg:

```
A tömb elemei: 6.39 ❄ 3.02 ❄ 19.95 ❄ -  
5.93 ❄ 34.58 ❄ 15.37
```

Segítség a feladathoz:

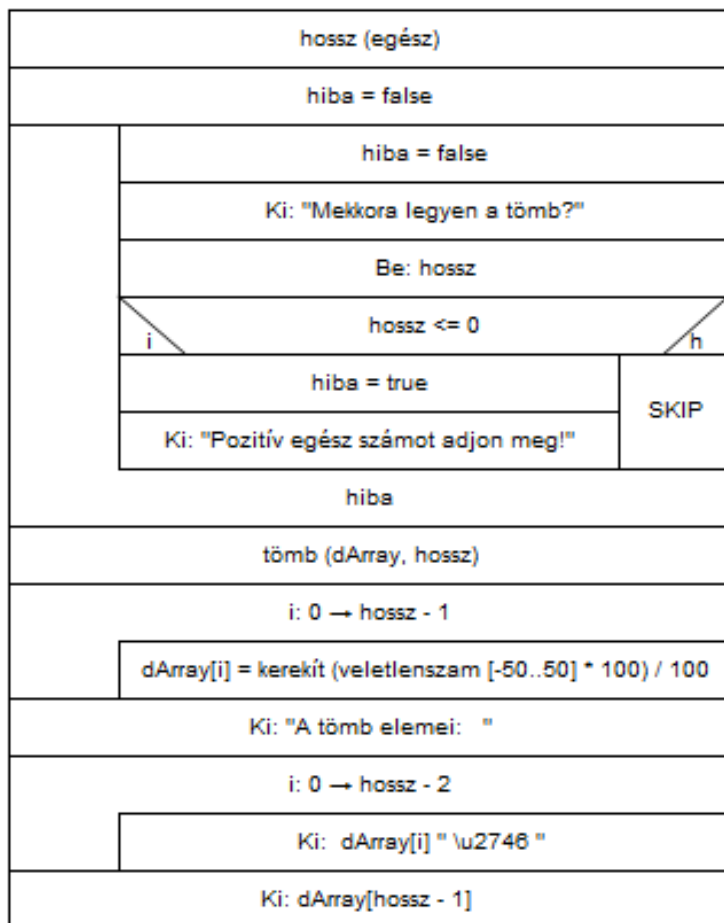
- Veletlenszam\_tabla.png-ben is megtalálható az alábbi szemléltető kép a hálózaton, töltsék le, lehet, hogy a későbbiekben még szükség lesz rá!

[-6;16]    16 - (-6) = 22		
nextInt(23)	[0;23[	
	[0;22]	/-6
nextInt(23) - 6	[-6;16]	
nextDouble()	[0;1]	/*22
nextDouble() * 22	[0;22]	/-6
nextDouble() * 22 - 6	[-6;16]	

- A Heavy chevron snowflake, ❄ karakter megtalálható a 3. órán használt unicode táblában.
  - <http://unicode-table.com/en/>
- Két tizedesjegyre kerekítés algoritmus (ha már elfelejtették volna): nézzenek utána! ☺

## Házi feladat megoldása

Struktogram:



Kód:

```
import java.util.Random;
import java.util.Scanner;

/**
 * @author Kovács Henrietta
 */
public class Gyakorlat13_6_KoVacsHenrietta {

    public static void main(String[] args) {

        Scanner billentyusok = new Scanner(System.in);
        Random generator = new Random();

        // 1.a) feladat
        // tömb elemszámának bekérése

        int hossz;
        boolean hiba = false;

        do {
            hiba = false;
            System.out.println("Mekkora legyen a tömb?");
            hossz = billentyusok.nextInt();
            billentyusok.nextLine();
            if (hossz <= 0) {
                hiba = true;
                System.out.println("Pozitív egész számot adjon meg!");
            }
        } while (hiba);

        // tömb létrehozása
        double[] darray = new double[hossz];

        // 1.b) feladat
        // tömb elemének felcserélése véletlenszámmal
        for (int i = 0; i < hossz; i++) {
            darray[i] = generator.nextDouble() * 100 - 50; // véletlenszám
            darray[i] = Math.round(darray[i] * 100) / 100.0; // kerekítés

            darray[i] = Math.abs(generator.nextDouble() * 100 - 50) * 100 / 100;
        }

        // 1.c) feladat
        // tömb elemének kiírása
        System.out.print("A tömb elemei:\n");
        for (int i = 0; i < hossz - 1; i++) {
            System.out.print(darray[i] + " \u2746 ");
        }
        System.out.println(darray[hossz - 1]);
    }
}
```

## Reflexió

Az üres String és üres tömb párhuzamba állítása az óra kezdetén nagyon hasznosnak bizonyult. Üres Stringgel már az előző órán is foglalkoztunk, sőt már többször előkerült, de az üres tömb még nem. Ennek megtanítása most azért volt aktuális, mert a felhasználó által megadott elemszámnál, - hiába létezik az üres tömb – nincs értelme nullát megadni. Ez úgy éreztem, kellőképpen tudatosult bennük. Az összehasonlítás pedig abban segített, hogy így már egy ismert fogalomhoz tudták kötni az új ismereteket és azzal, hogy kódírás által meg is tapasztalhatták ennek lényegét és nemcsak elméletben beszéltünk róla, sokkal jobban elraktározódott bennük.

A programozást, az algoritmusok tanítását nagymértékben motiválja a véletlenszám generálása. Ez most is így volt és már az előző órákon is szóba került, úgyhogy kellőképpen „fel voltak csigázva” a diákok. Már foglalkoztunk ezzel a témával, de csak a Math osztályon keresztül, amiről most megtudták a tanulók – nagy meglepetésükre - hogy csak álvéletlen számot ad. Ezek után még jobban érdekelte őket, hogy hogyan is lehet akkor véletlenszámokat előállítani. A táblaképet nagyon tudatosan megterveztem, mert az eddigi tapasztalataim alapján tudtam,

hogy ez problémát okozhat és a forráskód írása során nem szerettem volna, hogy gondot jelentsen. A tábláról szinte le tudták másolni az erre vonatkozó kódrészletet és mivel még külön file-ban is lekódoltuk, a rögzítés, megerősítés, elmélyítés szempontjából ez nagyon hasznos volt. A külön file-ban történő forráskód írásának emellett részemről az volt még a célja, ha bármikor előkerül véletlenszám generálás, akkor szükség esetén könnyen megtalálják.

Nagyon örültem, hogy végre adhattam házi feladatot is, amit a diákok mindig vasárnap éjfélig (keddenként van hat óránk) adnak be nekem. Természetesen, ha valami kérdésük van, vagy valami miatt nem fut a kódjuk, akkor is leadják, én reagálok rá, segítek nekik és ez alapján javítják. Ha nem adják le határidőre a feladatot, ha egy kicsit csúsznak, ennek még soha nem volt semmilyen következménye, mert nagyon kedvesek, ilyenkor elnézést kérnek és általában meg is indokolják a késés okát, de nem fordult elő sűrűn. Egyszer volt, hogy reggel kaptam házi feladatot, az órák előtt másfél órával, de megnéztem és az illető meglepődésére még vissza is jeleztem ☺. A házi feladat olyan szempontból is nagyon fontos, hogy a következő óra ennek a feladatnak a

folytatása lesz, az itt létrehozott tömbbel fogunk tovább dolgozni. Ha valaki valamilyen oknál fogva nem készít házi feladatot, annak a teljes megoldás elérhető lesz a hálózaton.

A feladat alapjaiban az előző öt órában megtanult ismeretek begyakorlását, megerősítését, rögzítését, tudatosítását, megszilárdítását, beidegződését, rutinná válását, valamint a diákok kommunikációs, rendszerező, tesztelő képességének, logikus és algoritmikus gondolkodásának fejlesztését szolgálta.

## ÓRATERVEZET - ALAPVETŐ ADATGYŰJTÉSI ÉS FELDOLGOZÁSI ALGORITMUSOK

**A pedagógus neve:** Kovács Henriette

**A szakmacsoport száma és megnevezése:** 7. Informatika

**A szakképesítés azonosító száma:** 54 213 05

**A szakképesítés megnevezése:** Szoftverfejlesztő

**Iskolai rendszerű szakképzés** (a szakképzési évfolyamok száma: 2)

**Szakmai követelménymodul:** 10817-12 Hálózatok, programozás és adatbázis-kezelés

**Tantárgy:** Adatbázis- és szoftverfejlesztés gyakorlat

**Programozási nyelv:** Java

**Fejlesztői környezet:** NetBeans IDE 7.3

**Osztály - csoport:** I/13. S - SZ2

**Az óra tananyaga:** Alapvető adatgyűjtési és feldolgozási algoritmusok

## **Az óra cél- és feladatrendszere**

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt. Kialakítani a tanulóknál az informatikai ismereteik folyamatos bővítésének és megújításának igényét.
- Korszerű alkalmazói készség kialakítása.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- A feladatmegoldáshoz megfelelő adattípus kiválasztásának fejlesztése.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés.

- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon a programban karakterek unicode-jával dolgozni, érdekes karaktereket unicode-juk felhasználásával megjeleníteni*
- *Tudjon a NetBeans IDE 7.3 –ban állományokat másik projektbe, másik csomagba, más néven átmásolni.*
- *Tudjon tömb típusú változót deklarálni, tömböt létrehozni, feltölteni.*
- *Tudjon tömböket megjeleníteni.*
- *Tudjon explicit konverziót alkalmazni.*
- *Tudjon algoritmust, programkódot megvalósítani, felhasználni, átalakítani, módosítani.*
- *Legyen képes a tömbök elemeit elérni, feldolgozni és velük műveleteket végezni.*
- *Legyen képes tesztelni.*

## **Fejlesztendő készségek, képességek**

- Visszaemlékezés, esztétikai érzék, állománykezelés, alkalmazás, megfigyelés, programkód átalakítása, módosítása, diagnosztizálás, hibakeresés, precizitás, pontosság, algoritmikus gondolkodás, algoritmus felismerése, megértése, algoritmus tervezése, algoritmus alkotása, struktogram készítése, programkód írása, szintaktika követése, kreativitás, tesztelés, programkód javítása, képletalkotás,

típusválasztás, futási eredmény analízisa, problémamegoldás sokszínűsége, következtetés, futási eredmény átláthatósága, követhetősége, struktogram átalakítása, rögzítés, ellenőrzés, megerősítés, összehasonlítás, módosítás, strukturáltság.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.
- Legyen képes alapvető adatgyűjtési és feldolgozási algoritmusokat (összegzés, számlálás) feladatmegoldásban alkalmazni.

### **Az óra didaktikai feladatai**

- Ellenőrzés
- Motiválás
- Ráhangolódás
- Aktivizálás
- Összehasonlítás
- Összefüggések keresése
- Feladat kitűzése
- Rögzítés
- Figyelemfelhívás
- Figyelem felkeltése
- Aktivizálás
- Ismétlés
- Ellenőrzés
- Előzetes ismeretek felidézése
- Megerősítés
- Rendszerezés
- Értékelés

**Tantárgyi kapcsolatok:** Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek

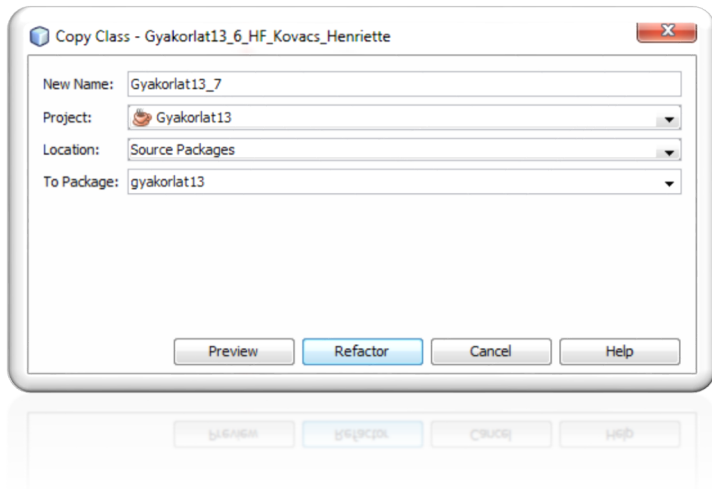
**Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):**

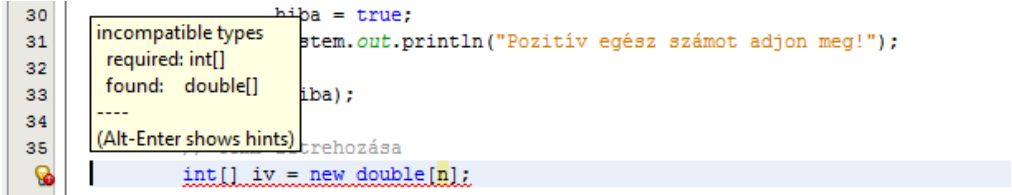
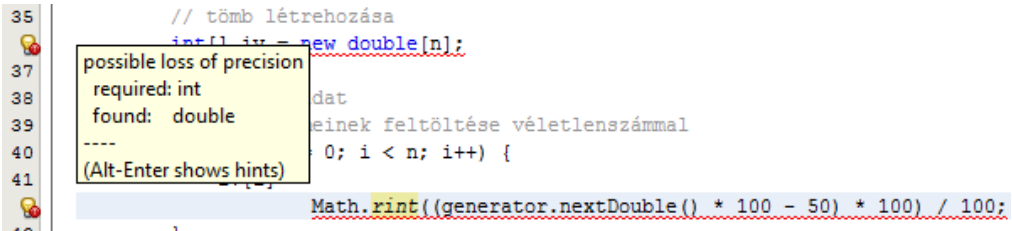
Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyne Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

<http://docs.oracle.com/javase/7/docs/api/>

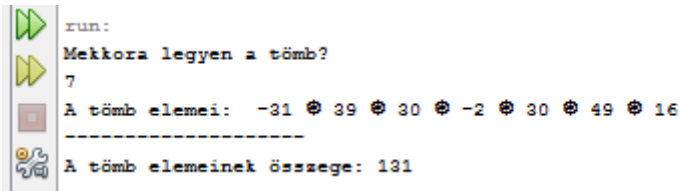
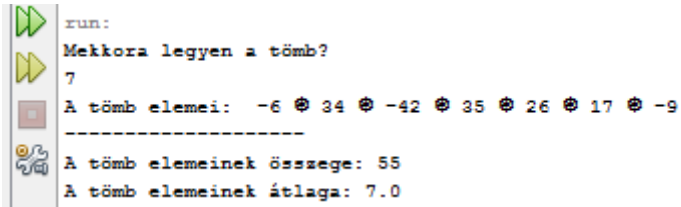
Dátum: éééé. hh. nn.

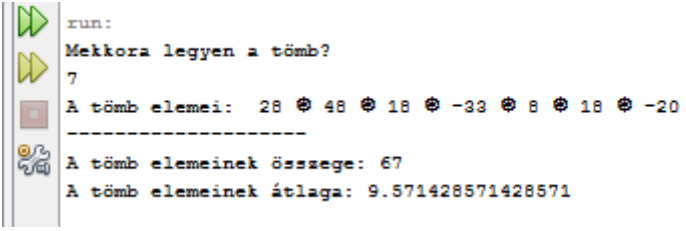
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 5'	<p>Tanulók üdvözlése ☺</p> <p>A házi feladat kódját szókétyákra vágtam, összekevertem. A szókétyákat a megfelelő sorrendbe kell rakniuk, vagy a számozás alapján a helyes sorrendet fel kell állítaniuk. (organikus okok)</p> <p>Gépek bekapcsolása.</p> <p>Házi feladat megnyitása, a benne előforduló karakterről, (Heavy chevron snowflake - Vastag ék alakú rangjelzés hópihe) néhány érdekes momentum:</p> <p>A karakter és a rendfokozat (Főtörzszászlós) jelzése közötti azonosság.</p> <div data-bbox="645 954 913 1129">  </div> <p><a href="http://hu.wikipedia.org/wiki/Rendfokozat#mediaviewer/F%C3%A1jl:Rank%20Army%20Hungary%20OR-09.svg">http://hu.wikipedia.org/wiki/Rendfokozat#mediaviewer/F%C3%A1jl:Rank%20Army%20Hungary%20OR-09.svg</a></p> <p><a href="http://hu.wikipedia.org/wiki/Rendfokozat#">http://hu.wikipedia.org/wiki/Rendfokozat#</a></p> <p><a href="http://unicode-table.com/en/">http://unicode-table.com/en/</a></p>	<p>Ellenőrzés Motiválás Visszaemlékezés</p> <p>Ráhangolódás Aktivizálás</p> <p>Esztétikai érzék</p> <p>Összehasonlítás Összefüggések keresése</p>	<p>Házi feladat ellenőrzése már elektronikusan megtörtént, a hibák ki vannak javítva, de hogy lássam, valóban tudják-e az eddigieket egy gyors ellenőrzési módszert választottam.</p> <p>Szókétyák</p> <p>Egyéni munka</p> <p>Számítógép</p> <p>Tanári elbeszélés, bemutatás</p> <p>„A részletekben rejlik a lényeg.”</p> <p>Frontális munka</p> <p>Hálózat Internet</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
5 – 7'	<p>A Gyakorlat13_6_HF_Vezeteknev_Keresztnev file másolása ugyanezen projekt gyakorlat13 csomagjába Gyakorlat13_7 néven!</p> <p>Jobb klikk a file néven/Refactor/Copy... és a megjelenő párbeszédablak kitöltése:</p> 	<div>Állománykezelés Alkalmazás Megfigyelés</div>	<div>Magyarázat Szemléltetés</div> <div>Irányított, önálló munka</div> <div>Tanári bemutatás</div> <div>Projektor Számítógép NetBeans IDE 7.3</div>
7 – 8'	<p>Feladat: 1. Változtassa meg a kódot! A tömb legyen int elemtípusú, a neve legyen iv, elemszáma legyen n.</p> <p>További szükséges változtatások:</p>	<div>Programkód átalakítása, módosítása</div>	<div>Egyéni munka</div>
8 – 10'	Visszajelzések a fejlesztői környezettől:		

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>Egyrészt:</p>  <p>new double[n] →</p> <p>new int[n]</p> <p>Másrészt: nem kell kerekíteni</p>  <p>Math.rint((generator.nextDouble() * 100 - 50) * 100) / 100 →</p> <p>generator.nextInt(101) - 50;</p>	<p>Diagnosztizálás, hiba-keresés</p> <p>Programkód átalakítása, módosítása</p>	<p>Egyéni munka</p> <p>ha nem megy →</p> <p>Megbeszélés</p> <p>Irányított, önálló munka</p> <p>Projektor Számítógép NetBeans IDE 7.3</p>
	Feladat sorszámát jelző kommentek javítása.	Precizitás, pontosság	Egyéni munka


IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
10 – 11'	Feladat: 2. Határozza meg a tömb elemeinek a) összegét, b) átlagát!	Feladat kitűzése Algoritmus felismerése, megértése	Tanári közlés Projektor Hálózat
11 – 15'	Összeg: Struktogram tervezése, létrehozása: <div data-bbox="465 611 1052 863" data-label="Diagram"> <pre> graph TD     A["sum = 0"]     B["i: 0 → n - 1"]     C["sum = sum + iv[i]"]     D["Ki: "A tömb elemeinek összege: " sum"]     A --- B     B --- C     C --- D   </pre> </div>	Algoritmikus gondolkodás Kreativitás  Algoritmus tervezése Algoritmus alkotása  Struktogram készítése	Beszélgetés Magyarázat  Irányított önálló munka Számítógép Structorizer Microsoft Office Excel 2007, 2010
15 – 17'	Kód elkészítése: <pre> // 2. feladat // tömb elemeinek összege int sum = 0; for (int i = 0; i &lt; n; ++i)     sum += iv[i]; System.out.println("A tömb elemeinek összege: " + sum); </pre>	Programkód írása Precizitás, pontosság Szintaktika követése Diagnosztizálás	Számítógép NetBeans IDE 7.3 Java 1.5 Egyéni munka

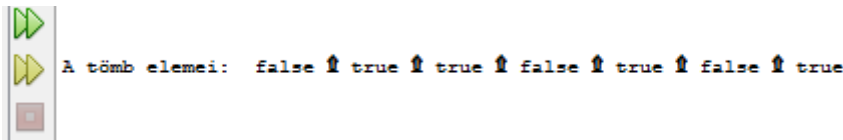
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>Futási eredmény:</p> 	<p>Tesztelés Hibakeresés Programkód javítása, módosítása Esztétikai érzék fej- lesztése</p>	<p>Egyéni munka</p> <p>NetBeans IDE 7.3 Számítógép Projektor</p>
17 – 19'	<p>Átlag:</p> <p><math>\text{átlag} = \text{elemek összege} / \text{elemek száma}</math></p> <p>Az átlag valós szám!</p> <p>Kód írása</p> <pre>// tömb elemeinek átlaga double atlag = sum / n; System.out.println("A tömb elemeinek átlaga: " + atlag);</pre> <p>Futási eredmény:</p> 	<p>Képletalkotás</p> <p>Rögzítés Figyelemfelhívás</p> <p>Típusválasztás Programkód írása</p> <p>Futási eredmény anali- zálása Diagnosztizálás Önellenőrzés</p>	<p>Megbeszélés</p> <p>Frontális munka</p> <p>Java 1.5 NetBeans IDE 7.3 Számítógép Projektor</p> <p>Egyéni munka</p>

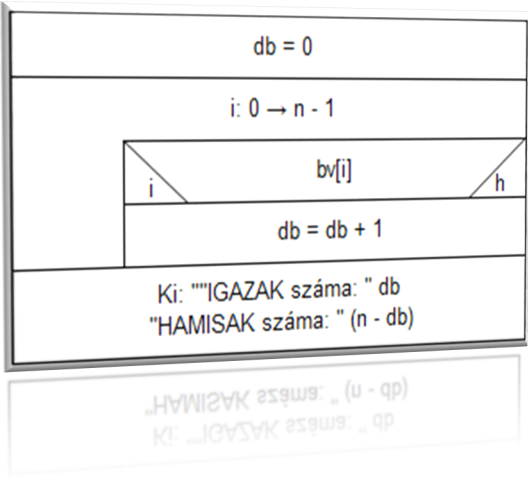
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
19 – 23'	<p>Nem jó! Egész osztás eredménye kerül a double típusú változóba! Valós osztást kell alkalmazni! → Típuskényszerítés: <b>Explicit, bővítő konverzió</b></p> <p>Kód javítása:</p> <pre>// tömb elemeinek átlaga double atlag = (double)sum / n; System.out.println("A tömb elemeinek átlaga: " + atlag);</pre> <p>Futási eredmény: Mivel érdemes tesztelni?</p>  <p>Kód átalakítása, hogy az átlag két tizedesjegy pontossággal jelenjen meg!</p> <p>Kódírási lehetőségek:</p> <pre>// tömb elemeinek átlaga double atlag = (double)sum / n; System.out.println("A tömb elemeinek átlaga: "     + Math.round(atlag * 100) / 100);</pre> <p>vagy:</p>	<div>Figyelem felkeltése Aktivizálás Ismétlés Következtetés</div> <div>Alkalmazás Programkód javítása</div> <div>Tesztelés</div> <div>Feladatkitűzés Esztétikai érzék fejlesztése</div> <div>Programkód átalakítása</div>	<div>Irányított, önálló munka</div> <div>Megbeszélés</div> <div>Java 1.5 NetBeans IDE 7.3 Számítógép Projektor</div> <div>Beszélgetés</div> <div>Egyéni munka</div>


IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<pre>// tömb elemeinek átlaga double atlag = (double)sum / n; System.out.println("A tömb elemeinek átlaga: "     + Math.round(atlag * 100) / 100.0);</pre> <p>Futási eredmény:</p>	<p>Problémamegoldás sokszínűsége</p> <p>Futási eredmény átlát- hatósága, követhető- sége Ellenőrzés</p>	<p>Megbeszélés</p> <p>Frontális munka</p> <p>NetBeans IDE 7.3</p> <p>Egyéni munka</p>
23 – 24'	<p>Feladat: 2. Határozza meg a tömb elemeinek</p> <p>c) négyzetösszegét, módosítsa az összegnél készített struktogramot, mentse Gyakorlat13_7_2c néven!</p> <p>Struktogram:</p>	<p>Struktogram átalakí- tása</p> <p>Állománykezelés</p> <p>Ellenőrzés</p>	<p>Egyéni munka</p> <p>Java 1.5 NetBeans IDE 7.3 Számítógép</p> <p>Projektor</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
24 – 25'	<p>Kód másolása, átalakítása:</p> <pre>// tömb elemeinek négyzetösszege sum = 0; for (int i = 0; i &lt; n; ++i) {     sum += Math.pow(iv[i], 2); } System.out.println("A tömb elemeinek négyzetösszege: " + sum);</pre> <p>Futási eredmény.</p>	<p>Programkód átalakítása Alkalmazás</p> <p>Rögzítés Megerősítés</p> <p>Önellenőrzés</p> <p>Ellenőrzés</p>	<p>Egyéni munka</p> <p>Java 1.5 NetBeans IDE 7.3 Számítógép</p> <p>Projektor</p>
25 – 26'	<p>Feladat: 3. Határozza meg az iv tömb elemeinek szorzatát, készítsen hozzá struktogramot, melyet mentsen Gyakorlat13_7_3 néven!</p> <p>Miben különbözik az összeg és a szorzat meghatározása?</p> <p>Az előző struktogram és kód felhasználása</p> <p>Kezdeti érték beállítása!</p> <p>Szükséges változtatások</p>	<p>Összehasonlítás</p> <p>Módosítás</p>	<p>Megbeszélés Magyarázat</p> <p>Frontális munka</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
26 – 28'	Struktogram másolása, módosítása: <div> <pre>       prod = 1       i: 0 → n - 1       prod = prod * iv[i]       Ki: "A tömb elemeinek szorzata: " prod           </pre> </div>	Állománykezelés  Struktogram átalakítása	Egyéni munka Java 1.5 NetBeans IDE 7.3 Számítógép
28 – 30'	Kód másolása, átalakítása: <pre> // 3. feladat // tömb elemeinek szorzata long prod = 1; for (int i = 0; i &lt; n; ++i) {     prod *= iv[i]; } System.out.println("A tömb elemeinek szorzata: " + prod);           </pre> Futási eredmény:  A tömb elemeinek szorzata: -3836250	Programkód átalakítása     Önellenzés → Hibajavítás	Folyamatosan követem munkájukat, igény szerint segítek.
30 – 31'	Feladat: 4. Írjon programot, <ol style="list-style-type: none"> <li>melyben létrehoz egy logikai értékekből álló, szintén n elemszámú bv nevű tömböt,</li> <li>feltölti a tömb elemeit véletlen logikai értékekkel,</li> <li>kiíratja a tömb elemeit tetszőleges elválasztó karakterrel,</li> </ol>	Feladatkitűzés Előzetes ismeretek felidézése Alkalmazás	Megbeszélés Frontális munka Projektor

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
27 – 31'	<p>Kód írása:</p> <pre>// 4. feladat // tömb létrehozása boolean[] bv = new boolean[n];  // tömb elemeinek feltöltése véletlen logikai értékkel for (int i = 0; i &lt; n; i++) {     bv[i] = generator.nextBoolean(); }  // tömb elemeinek kiírása System.out.print("A tömb elemei:\t"); for (int i = 0; i &lt; n - 1; ++i) {     System.out.print(bv[i] + " \u00E9 "); } System.out.println(bv[n - 1]);</pre> <p>Futási eredmény:</p> 	<p>Programkód írása Szintaktika követése Precizitás, pontosság Strukturáltság</p> <p>Alkotó képesség Önellenőrzés Esztétikai érzék</p>	<p>Egyéni munka</p> <p>Számítógép NetBeans IDE 7.3 Java 1.5</p>
31 – 32'	<p>Feladat:</p> <p>d) meghatározza a true és false értékek darabszámát!</p> <p>Ehhez a feladatrészhez készítsen algoritmust!</p>	<p>Algoritmikus gondolkodás Algoritmus felismerése, megértése Algoritmus alkotása</p>	<p>Tanulói elbeszélés Beszélgetés</p> <p>Frontális munka</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
32 – 37'	Struktogram elkészítése: 	Algoritmus megvalósítása Struktogram készítése	Frontális munka  Structorizer Microsoft Office Excel 2007, 2010
37 – 41'	Kód írása: <pre data-bbox="392 941 1025 1252"> // true és false elemek darabszámának megszámlálása int db = 0; for (int i = 0; i &lt; n; i++) {     if (bv[i]) {         db++;     } } System.out.println("IGAZAK száma: " + db     + "\nHAMISAK száma: " + (n - db));           </pre>	Programkód készítése Szintaktika követése  Strukturáltság Esztétikai érzék	Egyéni munka  NetBeans IDE 7.3 Java 1.5

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
41 – 42'	<p>Futási eredmény</p>  <pre>A tömb elemei: false &amp; false &amp; false &amp; false &amp; true &amp; false &amp; true IGAZAK száma: 2 HAMISAK száma: 5 BUILD SUCCESSFUL (total time: 18 seconds)</pre>	<p>Önellenőrzés → Hibajavítás</p>	<p>NetBeans IDE 7.3</p> <p>Egyéni munka</p>
42 – 44'	<p>Új algoritmusok:</p> <p>Sorozatszámítás</p> <p>Megszámolás</p>	<p>Ismétlés Rögzítés Megerősítés Rendszerezés</p>	<p>Megbeszélés</p> <p>Frontális munka</p>
44 – 45'	Értékelés	Értékelés	<p>Beszélgetés, elbeszélés</p> <p>Frontális munka</p>

## Feladatok

1. A Gyakorlat13\_6\_HF\_Vezeteknev\_Keresztnev file-t másolja ugyanezen projekt gyakorlat13 csomagjába Gyakorlat13\_7 néven!  
Változtassa meg a kódot! A tömb legyen int elemtípusú, a neve legyen iv, elemszáma legyen n.
2. Határozza meg az iv tömb elemeinek
  - a) összegét,  
készítsen hozzá struktogramot, melyet mentsen Gyakorlat13\_7\_2a néven,
  - b) átlagát,
  - c) négyzetösszegét,  
módosítsa az összegnél készített struktogramot, mentse Gyakorlat13\_7\_2c néven!
3. Határozza meg az iv tömb elemeinek szorzatát,  
készítsen hozzá struktogramot, melyet mentsen Gyakorlat13\_7\_3 néven!
4. Írjon programot,
  - a) melyben létrehoz egy logikai értékekből álló, szintén n elemszámú bv nevű tömböt,
  - b) feltölti a tömb elemeit véletlen logikai értékekkel,
  - c) kiírja a tömb elemeit tetszőleges elválasztó karakterrel,
  - d) meghatározza a true és false értékek darabszámát,  
ehhez a feladatrészhez készítsen algoritmus, melyet mentsen Gyakorlat13\_7\_4 néven!

## Reflexió

A házi feladat ellenőrzése részemről mindig elektronikus úton történik, meghatározott időre adják be a feladatokat a diákok, amiket ha kell többszörös levélváltás után javítanak. A feladat kapcsán ismét előkerült a unicode-tábla, mely alkalmas volt annak megválasztására, hogy az apró részletekben is milyen mély dolgok rejlenek, hogy milyen összefüggések fedezhetők fel még látszólag teljesen különböző objektumok között is. Ki gondolná, hogy egy karakter és egy rendfokozat jelzése között is találunk kapcsolatot. Ezek után a diákok már kellőképpen motiváltak voltak és kezdhették a feladatokat.

A programozás tanítása során nagyon fontosnak tartom már megírt kódrészletek ismételt felhasználását, alkalmazását, annak szükség esetén megfelelő átalakítását, módosítását, javítását. Tulajdonképpen, amikor beépített metódusokat használunk, felülírunk, munkánk alapja egy már megírt kód, tehát használjuk, amit már megírtak. Mindig arra ösztönzöm diákjaimat, hogy ami már elkészült azt ne akarjuk még egyszer megvalósítani (ha tüzet akarunk gyújtani, gyufát keressünk és ne követ ☺). Ez a szemlélet az egész órát végig kísérte, tulajdonképpen a kezdetektől fogva, amikor még a házi feladatot is felhasználtuk további

munkánkhoz, és az első feladatban leírt átalakításokat végrehajtva dolgoztunk tovább. Ezáltal fejlesztettem többek között a diákok logikus és algoritmikus gondolkodását, a precizitást, a pontosságot, a szintaktika követését, a diagnosztizálást, a hibakeresést, az önellenőrzést.

Az új ismereteket a gyakorlásba úgy építettem be, hogy tulajdonképpen szinte észre sem vették a diákok, hogy most már a gyakorlatban is alkalmaztak két programozási tételt, a sorozatszámítást (már Stringekkel kapcsolatban is előkerült) és a megszámlálást, ezáltal így tömb elemeivel különböző műveleteket tudtak végezni, mint például összeadást, szorzást és hatványozást. Az óra végi összefoglalásban mindezt tudatosítottam, azáltal, hogy nevén neveztem a dolgokat, így látták a párhuzamot az elmélet és a gyakorlat között. Meg is jegyezték, hogy elméletben mindez sokkal bonyolultabbnak tűnt.

Az utolsó feladat nagyon jó választásnak bizonyult, a logikai értékekből álló tömb egy kicsit feldobta azt a tényt, hogy nagyrészt csak szám típusú tömbökkel dolgoztunk és mellesleg tökéletesen alkalmas volt a megszámlálás algoritmusának elsajátítására.

Nagyon sok feladat megoldására sor került, nem voltam benne biztos, hogy lesz idő a negyedik feladat megvalósítására. Ebben az esetben az a), b), c) feladatok forráskódját a hálózatról töltötték volna le, a kódot elemeztük volna és a d) feladat megoldására került volna „csak” sor.

Az órán megvalósítottam azokat a dolgokat, amiket célul tűztem ki magam elé.

## **ÓRATERVEZET - MAXIMUMKIVÁLASZTÁS. TÖMBÖK ELEMEIN VÉGZETT MŰVELETEK.**

**A pedagógus neve:** Kovács Henriette

**A szakmacsoport száma és megnevezése:** 7. Informatika

**A szakképesítés azonosító száma:** 54 213 05

**A szakképesítés megnevezése:** Szoftverfejlesztő

**Iskolai rendszerű szakképzés** (a szakképzési évfolyamok száma: 2)

**Szakmai követelménymodul:** 10817-12 Hálózatok, programozás és adatbázis-kezelés

**Tantárgy:** Adatbázis- és szoftverfejlesztés gyakorlat

**Programozási nyelv:** Java

**Fejlesztői környezet:** NetBeans IDE 7.3

**Osztály - csoport:** I/13. S - SZ2

**Az óra tananyaga:** Maximumkiválasztás. Tömbök elemein végzett műveletek.

### **Az óra cél- és feladatrendszere**

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt.
- Korszerű alkalmazói készség kialakítása.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.

- *Tudjon tömb típusú változót deklarálni, létrehozni, feltölteni.*
- *Tudjon tömböt megjeleníteni.*
- *Legyen képes algoritmust átalakítani, módosítani, javítani.*
- *Legyen képes forráskódot átalakítani, módosítani, javítani.*
- *Legyen képes algoritmizálni.*
- *Legyen képes a tömbök elemeit elérni, feldolgozni és velük műveleteket végezni.*
- *Tudjon tömb elemei közül maximumot kiválasztani.*
- *Tudjon algoritmust, programkódot megvalósítani, felhasználni, átalakítani, módosítani.*
- *Legyen képes tesztelni.*

### **Fejlesztendő készségek, képességek**

- Algoritmikus gondolkodás, algoritmus felismerése, algoritmus megértése, algoritmizálás, algoritmus elemzése, algoritmus alkotása, visszaemlékezés, struktogram készítése, önellenőrzés, algoritmus megvalósítása, programkód átalakítása, módosítása, javítása, tesztelés, programkód írása, logikus gondolkodás, problémamegoldás, algoritmus módosítása, átalakítása, elemzés, fokozatoság, összehasonlítás, precizitás, pontosság, összefüggések

felismerése, algoritmus végrehajtása, alkotó képesség, szintaktika követése, önállóság.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanuló társa is fel tudja használni.
- Legyen képes alapvető adatgyűjtési és feldolgozási algoritmusokat (összegzés, számlálás) feladatmegoldásban alkalmazni.

### **Az óra didaktikai feladatai**

- Motiválás
- A figyelem felkeltése
- Ismétlés
- Feladat kitűzése
- Aktivizálás
- A tanulók informálása a tanulási célról
- Szemléletesség
- Az új ismeretet megalapozó tények biztosítása
- A folyamat elemzése
- Következtetés
- Absztrakció
- Általánosítás
- Rögzítés
- Ellenőrzés
- Értékelés
- Visszacsatolás

**Tantárgyi kapcsolatok:** Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek, Biológia


**Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):**

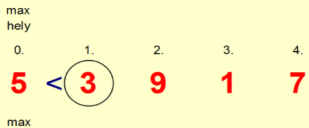
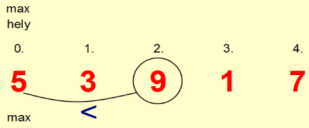

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyne Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

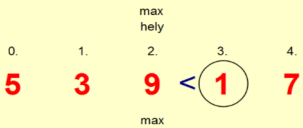
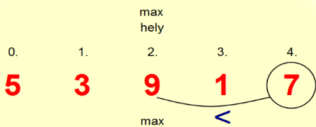

<http://docs.oracle.com/javase/7/docs/api/>

Dátum: éééé. hh. nn.

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 4'	<p>Egészséges táplálkozás – legjobb, legegészségesebb áru megvételére törekvés</p> <p>A mai órán mi is a maximális megtalálására törekszünk, megkeressük, hol van egy tömb legnagyobb eleme és melyik az!</p> <p>Az előző órán algoritmusokat alkottunk: kiszámoltuk a tömbelemek összegét, négyzetösszegét, átlagát, szorzatát, illetve megszámoltunk adott tulajdonságú elemeket.</p> <p>Vegyünk egy 5 elemű tömböt a következő elemekkel: 5, 3, 9, 1, 7</p> <p>Milyen algoritmust tudnánk kitalálni, hogy meg tudjuk valósítani célunkat?</p> <p>Kövesse figyelemmel a következő képek sorozatát, melyek szemléltetik az algoritmus lényegét!</p> <p>Próbálja megfogalmazni a képek alapján az algoritmust!</p>	<p>Motiválás, a figyelem felkeltése,.</p> <p>Ismétlés</p> <p>Feladat kitűzése</p> <p>Aktivizálás</p>	<p>Elbeszélés Megbeszélés</p> <p>Frontális munka</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
4 – 10'	 <p><b>5 3 9 1 7</b></p> <p><i>Van egy ötelemű tömbünk.</i></p> <p> <small>max hely</small>  0.    1.    2.    3.    4.  <b>5    3    9    1    7</b>  <small>max</small> </p> <p><i>Tegyük fel, hogy a 0. elem a legnagyobb, azaz az 5, így a 0 lesz a maximumhely, az 5 pedig a maximumérték.</i></p>	<p>A tanulók informálása a tanulási célról</p> <p>Szemléletesség</p> <p>Az új ismeretet megalapozó tények biztosítása</p> <p>A folyamat elemzése</p>	<p>Frontális munka</p> <p>Projektor</p> <p>Szemléltetés</p> <p>Tanári bemutatás</p> <p>Tanári, tanulói elbeszélés</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<div>  <p>Az 1. elemet összehasonlítjuk az aktuális maximummal, azaz a 0. elemmel. Ha nagyobb, akkor ő lesz az aktuális maximum, ha nem, megyünk tovább.</p> </div> <div>  <p>A következő, azaz a 2. elemet összehasonlítjuk az aktuális maximummal, azaz a 0. elemmel. Mivel nagyobb nála,</p> </div> <div>  <p>ezért a 2. elem lesz az aktuális maximum. Tehát most a maximumhely a 2, a maximumérték a 9.</p> </div>	<div>Algoritmikus gondolkodás</div> <div>Algoritmus felismerése</div> <div>Algoritmus megértése</div>	<div>Frontális munka</div> <div>Szemléltetés</div> <div>Projektor</div> <div>Magyarázat</div> <div>Megbeszélés</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<div>  <p><i>A következő, azaz a 3. elemet összehasonlítjuk az aktuális maximummal, azaz a 2. elemmel. Nem nagyobb nála.</i></p> </div> <div>  <p><i>A következő, azaz a 4. indexű elemet összehasonlítjuk az aktuális maximummal, azaz a 2. elemmel. Nem nagyobb nála.</i></p> </div> <div>  <p><i>Végig értünk a tömbön, nincs több eleme, tehát a 2. indexű eleme a legnagyobb, azaz a 9.</i></p> </div>	<div>Algoritmizálás</div> <div>Algoritmus elemzése</div> <div>Algoritmus alkotása</div> <div>Következtetés</div>	<div>Frontális munka</div> <div>Szemléltetés</div> <div>Projektor</div> <div>Tanulói elbeszélés</div> <p>Ha úgy ítélem meg, hogy van rá idő: mi történne, ha nem <math>&lt;</math>, hanem <math>\leq</math> relációjel szerepelne.</p> <p>szükség esetén</p> <div>Tanári elbeszélés Magyarázat</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
10 – 12'	Fogalmazza meg az algoritmust konkrét adatok nélkül!	Absztrakció  Általánosítás	Tanulói elbeszélés szükség esetén  Tanári elbeszélés
12 – 17'	Struktogram : <div data-bbox="497 596 1059 959" data-label="Diagram"> <pre> graph TD     Start([tömb (iv, n)]) --&gt; Init[maxhely = 0]     Init --&gt; LoopStart[i: 1 → n - 1]     LoopStart --&gt; Decision{iv[maxhely] &lt; iv[i]}     Decision --&gt; Assign[maxhely = i]     Assign --&gt; LoopEnd[SKIP]     LoopEnd --&gt; Exit[Ki: maxhely + 1, iv[maxhely]]           </pre> </div>	Visszaemlékezés Struktogram készítése  Rögzítés  Önellenőrzés Ellenőrzés	Structorizer Microsoft Office Excel 2007, 2010  Irányított, önálló munka  Projektor
17 – 18'	Feladat: 1. a) Készítsen alkalmazást, amely [1; 20] zárt intervallumban lévő véletlen-számmal feltölt egy 10 elemű, egész elemtípusú, iv nevű tömböt! A kódolás során a length mezőt használja!  Mely kódrészletünket tudnánk felhasználni? Gyakorlat13_7-ből a 2. feladatig a kódrészlet átmásolása	Feladat kitűzése      Visszaemlékezés	Hálózat Projektor   Megbeszélés

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
18 – 22'	<p>Javítások:</p> <p>Scanner objektum törlése</p> <p>hátultesztelő ciklus törlése</p> <p>tömb létrehozásánál <math>n \rightarrow 10</math></p> <p><math>n</math> átnevezése (CTRL+R) <math>\rightarrow</math> iv. length <math>\rightarrow</math></p> <p>feleslegessé vált deklaráló sor törlése</p> <p>véletlenszám javítása <math>\rightarrow</math> nextInt(20) + 1</p> <p>A kódrészlet:</p> <pre> Scanner billentyusát = new Scanner(System.in); Random generátor = new Random();  // 1. feladat // tömb elemzésének bekérése int n; boolean hiba = false;  do {     hiba = false;     System.out.println("Mekkora legyen a tömb?");     n = billentyusát.nextInt();     billentyusát.nextLine();     if (n &lt;= 0) {         hiba = true;         System.out.println("Positív egész számot adjon meg!");     } } while (hiba);  // tömb létrehozása int[] iv = new int[n];  // tömb elemeinek feltöltése véletlenszámmal for (int i = 0; i &lt; n; i++) {     iv[i] = generátor.nextInt(101) - 50; }  // tömb elemeinek kiírása System.out.print("A tömb elemei:\n"); for (int i = 0; i &lt; n - 1; i++) {     System.out.print(iv[i] + " \u2746 "); } System.out.println(iv[n - 1]); </pre>	<p>Algoritmus felismerése</p> <p>Algoritmus alkotása</p> <p>Algoritmizálás</p> <p>Algoritmus megvalósítása</p> <p>Programkód átalakítása, módosítása, javítása</p>	<p>Megbeszélés</p> <p>Fontos a sorrend!</p> <p>Irányított, önálló munka</p> <p>Projektor</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>A helyes kódrészlet:</p> <pre> package gyakorlat13;  import java.util.Random; import java.util.Scanner;  /**  *  * @author Kovács Henriette  */ public class Gyakorlat13_8 {      public static void main(String[] args) {          // 1. feladat         Random generator = new Random();          // tömb létrehozása         int[] iv = new int[10];          // tömb elemeinek feltöltése véletlenszámmal         for (int i = 0; i &lt; iv.length; i++) {             iv[i] = generator.nextInt(20) + 1;         }          // tömb elemeinek kiírása         System.out.print("A tömb elemei:\t");         for (int i = 0; i &lt; iv.length - 1; ++i) {             System.out.print(iv[i] + " \u2746 ");         }         System.out.println(iv[iv.length - 1]); </pre> <p>A programkód futtatása</p>	<p>Programkód írása</p> <p>Önellenőrzés</p> <p>Ellenőrzés</p> <p>Tesztelés</p>	<p>Importálást ne töröljük, ha a további kódrészletek során nincs rá szükség, azt úgyis jelzi a fejlesztői környezet.</p> <p>(szükségünk lesz rá ☺)</p> <p>Egyéni munka</p> <p>Projektor</p> <p>Számítógép NetBeans IDE 7.3 Java 1.5</p>

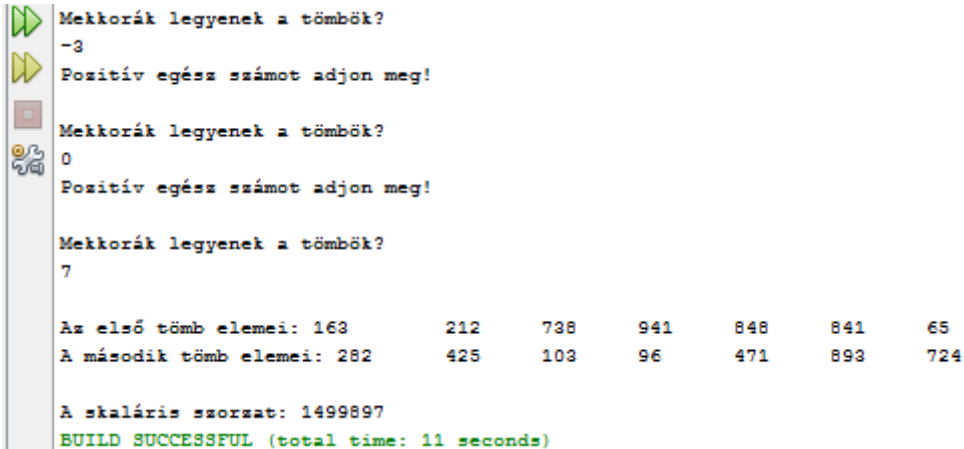
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
22 – 26'	<p>Feladat:</p> <p>b) Adja meg, hogy a tömbben hányadik helyen fordul először elő a tömb legnagyobb eleme és ez mennyi!</p> <p>A feladat tulajdonképpen az elkészített algoritmus megvalósítása, kódolása..</p> <p>Kód elkészítése:</p> <pre>// maximumkiválasztás int maxhely = 0; // az első elem lokális maximum  for (int i = 1; i &lt; iv.length; ++i) {     if (iv[i] &gt; iv[maxhely]) {         maxhely = i;     } }  System.out.println("A legnagyobb elem a/az " + (maxhely + 1) + "., "     + "aminek az értéke " + iv[maxhely] + ".");</pre> <p>Futási eredmény:</p>  <p>A tanulók munkájának elemzése</p>	<p>Feladat kitűzése</p> <p>Visszaemlékezés Visszacsatolás</p> <p>Algoritmus megvalósítása</p> <p>Programkód írása</p> <p>Önellenőrzés</p> <p>Ellenőrzés</p> <p>Értékelés</p>	<p>Frontális munka</p> <p>Projektor Hálózat</p> <p>Számítógép NetBeans IDE 7.3 Java 1.5</p> <p>Projektor</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
26 – 30'	<p>Feladat:</p> <p>2. a) Töltsön fel 2 azonos méretű tömböt egész számokkal (a tömb méretét beolvasással határozza meg; a feltöltés automatikusan, véletlenszámokkal történjen). Írassa ki az első, majd a második tömb elemeit tabulátorral elválasztva!</p> <p>Mely kódrészletünket tudnánk felhasználni?</p> <p>Gyakorlat13_6.java file-ban található kódot.</p> <p>Gyakorlat13_6-hoz tartozó struktogram (Gyakorlat13_6.xlsx) megnyitása.</p> <p>A szükséges változtatások megbeszélése:</p> <ul style="list-style-type: none"> <li>• a ciklusban a kiíratás megváltoztatása</li> <li>• tömb neve tomb → tomb1</li> <li>• tomb2 létrehozása (KIEGÉSZÍTÉS)</li> <li>• véletlenszám-generálás javítása</li> <li>• tomb2 tömb feltöltése véletlenszámmal (KIEGÉSZÍTÉS)</li> <li>• kiíratás javítása → tabulátor</li> <li>• tomb2 elemeinek kiíratása (KIEGÉSZÍTÉS)</li> </ul> <p>Gyakorlat13_8_2.xlsx tartalmazza az új struktogramot, melyet a hálózatra is felteszek, ahonnan a diákok letöltve a kódolást ez alapján végzik.</p>	<p>Feladatkitűzés</p> <p>Visszaemlékezés Visszacsatolás</p> <p>Összefüggések felismerése</p> <p>Algoritmizálás Logikus gondolkodás Problémamegoldás</p> <p>Algoritmus módosítása, átalakítása</p>	<p>Projektor</p> <p>Gyakorlat13_6.xlsx</p> <p>Megbeszélés Frontális munka</p> <p>Gyakorlat13_8_2.xlsx Hálózat</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>Gyakorlat13_6.xlsx      A szükséges változtatá-      Gyakorlat13_8_2.xlsx</p> <p>Struktogram      sok megbeszélése      Megváltoztatott strukto- gram</p>	<p>Elemzés Fokozatosság Összehasonlítás</p> <p>Szemléletesség</p>	<p>Hálózat Gyakorlat13_8_2.xlsx</p> <p>Szemléltetés</p> <p>Megbeszélés</p> <p>Projektor</p> <p>Frontális munka</p>
30 – 34	<p>Kód elkészítése:</p> <p>A beírt kommentek, megjegyzések módosítása</p>	<p>Algoritmus megvalósí- tása</p> <p>Precizitás, pontosság</p>	<p>Számítógép NetBeans IDE 7.3 Java 1.5</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<pre> // 2. feladat Scanner billentyuset = new Scanner(System.in);  // tömbök elemszámának bekérése int n; boolean hiba = false;  do {     hiba = false;     System.out.println("Mekkora legyenek a tömbök?");     n = billentyuset.nextInt();     billentyuset.nextLine();     if (n &lt;= 0) {         hiba = true;         System.out.println("Positív egész számot adjon meg!\n");     } } while (hiba);  // tömbök létrehozása int[] tomb1 = new int[n]; int[] tomb2 = new int[n];  // 2 tömb elemeinek feltöltése véletlenszámmal for (int i = 0; i &lt; n; ++i) {     tomb1[i] = generator.nextInt(1000);     tomb2[i] = generator.nextInt(1000); }  // 2 tömb elemeinek kiírása System.out.print("\nAz első tömb elemei: "); for (int i = 0; i &lt; n; ++i) {     System.out.print(tomb1[i] + "\t"); } System.out.println();  System.out.print("\nA második tömb elemei: "); for (int i = 0; i &lt; n; ++i) {     System.out.print(tomb2[i] + "\t"); } System.out.println(); </pre>	<div data-bbox="1335 316 1635 395">Algoritmus végrehajtása</div> <div data-bbox="1335 472 1635 517">Alkotó képesség</div> <div data-bbox="1335 593 1635 750">Programkód módosítása, átalakítása, javítása, írása Szintaktika követése</div> <div data-bbox="1335 826 1635 871">Önállóság</div> <div data-bbox="1335 948 1635 992">Aktivizálás</div> <div data-bbox="1335 1069 1635 1200">Gyakorlás Megerősítés Ellenőrzés</div>	<div data-bbox="1713 373 2087 421">Gyakorlat13_6.java</div> <div data-bbox="1713 561 2087 692">Számítógép NetBeans IDE 7.3 Java 1.5</div> <div data-bbox="1713 903 2087 951">Egyéni munka</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
34 – 35'	Az eddigi kódrészlet futtatása	Futási eredmény elem- zése	Egyéni munka
35 – 38'	<p>Feladat:</p> <p>b) Számítsa és írassa ki a 2 tömb skaláris szorzatát! (A skaláris szorzat a megfelelő számpárok szorzatának összege.)</p> <p>Struktogram elkészítése:</p> <p>Valóban szorzatot kell számolnunk?</p> <div data-bbox="398 874 1111 1142" data-label="Diagram"> <pre> graph TD     A[osszeg = 0] --&gt; B[i: 0 → n-1]     B --&gt; C[osszeg = osszeg + tomb1[i] * tomb2[i]]     C --&gt; D[Ki: "A skaláris szorzat: " osszeg] </pre> </div>	<p>Feladatkitűzés</p> <p>Figyelemfelhívás</p> <p>Visszaemlékezés</p> <p>Algoritmikus gondol- kodás Algoritmus alkotása</p> <p>Struktogram készítése</p> <p>Ellenőrzés</p>	<p>Projektor</p> <p>Hálózat</p> <p>Frontális munka</p> <p>Megbeszélés</p> <p>Egyéni munka</p> <p>Structorizer Microsoft Office Excel 2007, 2010</p>
38 – 42'	Kód elkészítése:		

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<pre>//Skaláris szorzat long osszeg = 0; for (int i = 0; i &lt; n; ++i) {     osszeg += tomb1[i] * tomb2[i]; } System.out.println();  System.out.println("A skaláris szorzat: " + osszeg);  } }</pre> <p>Futási eredmény:</p> 	<div>Algoritmus megvalósítása Programkód írása</div> <div>Önellenőrzés Ellenőrzés</div> <div>Futási eredmény elemzése</div>	<div>Egyéni munka</div> <div>Számítógép NetBeans IDE 7.3 Java 1.5</div>
42 – 44'	<p>Maximumkiválasztás lényege</p> <p>Ha van idő: minimumkiválasztás miben különbözik ettől?</p>	<div>Rögzítés Megerősítés</div>	<div>Beszélgetés</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
44 – 45'	Tanulók munkájának minősítése	Értékelés	Megbeszélés

## Feladatok

1. a) Készítsen alkalmazást, amely  $[1; 20]$  zárt intervallumban lévő véletlenszámmal feltölt egy 10 elemű, egész elemtípusú, `iv` nevű tömböt! A kódolás során a `length` mezőt használja!
- b) Adja meg, hogy a tömbben hányadik helyen fordul először elő a tömb legnagyobb eleme és ez mennyi!
2. a) Töltsön fel 2 azonos méretű tömböt egész számokkal (a tömb méretét beolvasással határozza meg; a feltöltés automatikusan, véletlenszámokkal történjen). Írassa ki az első, majd a második tömb elemeit tabulátorral elválasztva!
- b) Számítsa és írassa ki a 2 tömb skaláris szorzatát! (A skaláris szorzat a megfelelő számpárok szorzatának összege.)

## Reflexió

Aki kezébe veszi az óravázlatomat, meglepődhet azon, hogy hogyan kerül ide az egészséges táplálkozás, mi köze van a maximumkiválasztáshoz. Ehhez tudni kell, hogy nagyon sokat beszélgetek a diákokkal az egészséges életmódról, annak élettani hatásairól és bár annak ellenére, hogy kerekesszékhez kötöttek (1 kivételével), sportolnak, de náluk különösen fontos, hogy mit visznek be szervezetükbe, hogy az elhízást elkerüljék, mert ugye amit beviszünk, azzá válunk. A cél tehát a legegészségesebb, legjobb kiválasztása – ami manapság elég nehéz feladat a tartósítószeres térhódítása miatt – vagyis törekszünk, mint mindenben a maximalizmus elvére. Így tehát a tömbelemek közül is szeretnénk kiválasztani a maximumot. Ezzel a kis bevezetővel kellőképpen ráhangoltam a diákokat az óra anyagára, tudtam, erről mindig szívesen beszélgetnek velem.

Az óra az algoritmikus gondolkodás kompetencia szintjeit képező, valamennyi összetevőt fejleszti. Ezek: az algoritmus megértése, azaz megérteni, hogy mit kell tenni és megérteni, hogy miért azt kell tenni, az algoritmus felismerése, az algoritmus elemzése, az algoritmus

végrehajtása, az algoritmus alkotása, az algoritmus megvalósítása, az algoritmus módosítása, átalakítása, az algoritmus tervezése.

A maximumkiválasztás algoritmusára nem jöttek rá, nem tudták megközelíteni a problémát, de a szemléletesség, mint didaktikai feladat bevetése most igazán célszerű volt, mert teljesen akadálymentesen tudtam haladni a diasorokon és a végén az általánosítás, az absztrakt gondolkodás sem ütközött akadályokba. A feladat algoritmusának leírása, a struktogram készítése ennek ellenére egy kicsit nehézkesen ment, a kódolás viszont, az algoritmus megvalósítása nem okozott problémát.

A második feladatban nagy hangsúlyt fektettem elkészített struktogramok és megírt kódrészletek felhasználására. Ezáltal lehetőség nyílt újabb algoritmizálásra (hiszen a forráskódban nem mindegy, hogy milyen sorrendben hajtjuk végre a módosításokat), a logikus gondolkodás fejlesztésére, forráskód átalakítására, módosítására. Utolsó lépésként pedig azt gondoltam, hogy a sorozatszámítás gondot fog jelenteni, mivel két tömbön kellett dolgozniuk, de ez nem okozott problémát; viszont nem összeadásban, hanem szorzásban gondolkodtak volna a skáláris szorzat elnevezés miatt, ekkor felhívtam figyelmüket a zárójelbe

tett magyarázó szövegre, melyből egyből kiderült számukra, hogy mit is kell tenniük. Tudatosítottam bennük, hogy mindig legyenek precízek, pontosak és a zárójelben lévő, látszólag nem lényeges gondolatok is jelentősek lehetnek.

Az órát sikeresnek ítélem meg, mert a reflexiókban felsoroltakon kívül még számos, a terv elején felsorolt készségeket és képességeket is fejlesztettem.

## **ÓRATERVEZET - ÖSSZETETT ELEMTÍPUSÚ TÖMBÖK. TÖMBELEMENK CSERÉJE.**

**A pedagógus neve:** Kovács Henriette

**A szakmacsoport száma és megnevezése:** 7. Informatika

**A szakképesítés azonosító száma:** 54 213 05

**A szakképesítés megnevezése:** Szoftverfejlesztő

**Iskolai rendszerű szakképzés** (a szakképzési évfolyamok száma: 2)

**Szakmai követelménymodul:** 10817-12 Hálózatok, programozás és adatbázis-kezelés

**Tantárgy:** Adatbázis- és szoftverfejlesztés gyakorlat

**Programozási nyelv:** Java

**Fejlesztői környezet:** NetBeans IDE 7.3

**Osztály - csoport:** I/13. S - SZ2

**Az óra tananyaga:** Összetett elemtípusú tömbök. Tömbelemek cseréje.

## Az óra cél- és feladatrendszere

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt. Kialakítani a tanulóknál az informatikai ismereteik folyamatos bővítésének és megújításának igényét.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- A feladatmegoldáshoz megfelelő adattípus kiválasztásának fejlesztése.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon a szakirodalmak között tájékozódni.*
- *Tudjon algoritmust alkotni.*
- *Tudjon algoritmust magvalósítani.*
- *Tudjon megfelelő adatszerkezetet és adattípust választani.*
- *Tudjon a NetBeans IDE-ben osztályt létrehozni, állományt átnevezve másolni.*
- *Tudjon tömböt létrehozni, billentyűzetről feltölteni, kiírni.*
- *Tudja egyszerű és összetett elemtípusú tömb alapértelmezett értékeit meghatározni.*
- *Tudjon a Java hivatalos dokumentációjában keresni.*
- *Tudjon Internetről információt gyűjteni.*
- *Tudjon Scanner objektumot úgy létrehozni, hogy a magyar 'ő' és 'ű' karakterek is megjeleníthetők legyenek.*
- *Tudjon programkódot módosítani, futtatni, a futási eredményt elemezni.*
- *Tudjon tömbelemeket cserélni.*
- *Tudja, hogy összetett típus értékadásakor, a referencia állítódik át.*
- *Ismerje a RuntimeException futási hibát.*
- *Tudja tömb referenciájának alapértelmezett értékét meghatározni.*

### **Fejlesztendő készségek, képességek**

- Algoritmizálás, algoritmikus gondolkodás, algoritmus felismerése, algoritmus alkotása, adatszerkezet-választás, típusválasztás, állománykezelés, programkód írása, visszaemlékezés, precizitás, pontosság, futtatás, absztrakt gondolkodás, megfigyelés, kódrészlet írása, olvasott szakmai szöveg megértése, programkód átalakítása, önállóság, logikus gondolkodás, visszaemlékezés, algoritmus megvalósítása, programkód átalakítása, önellenőrzés, futási hiba felismerése, diagnosztizálás, hibakeresés, fejlesztői környezet jelzésének figyelembevétele, futási eredmény elemzése.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait. Tudjon adatokat megkeresni, elérni a hálózati szolgáltatások alkalmazásával.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.

- Legyen képes elemi és összetett adatok megkülönböztetésére, kezelésére, használatára.
- Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.
- Legyen képes eligazodni és tudjon keresni a Java hivatalos dokumentációjában.

### **Az óra didaktikai feladatai**

- Motiválás
- A figyelem felkeltése
- Szemléletesség
- Figyelemfelhívás
- Szakirodalom megismertetése
- Tudományosság
- Feladat kitűzése
- Ismétlés
- A szükséges előzetes ismeretek felidézése
- A tanulók informálása a tanulási célról
- Absztrakció, általánosítás
- Új ismeret közlése
- Fokozatosság
- Információ gyűjtése, elemzése
- Problémafelvetés
- Új ismereteket megalapozó tények biztosítása
- Futási eredmény elemzése
- Megvalósítás
- Következtetés
- Visszacsatolás
- Megerősítés
- Rögzítés
- Aktivizálás
- Ellenőrzés
- Rendszerezés
- Értékelés
- Futási eredmény átláthatósága, követhetősége

**Tantárgyi kapcsolatok:** Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek, Könyvtárismeret, Magyar

**Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):**

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyiné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

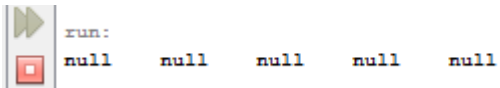
<http://docs.oracle.com/javase/7/docs/api/>

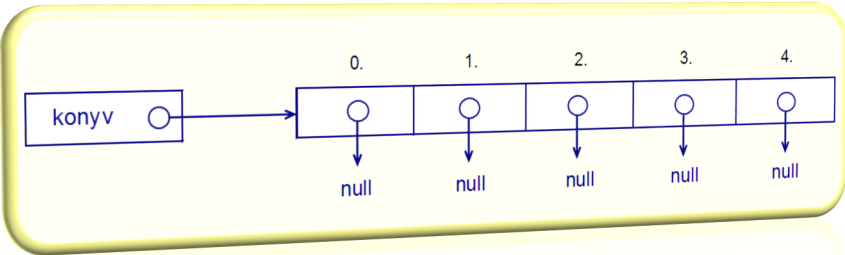
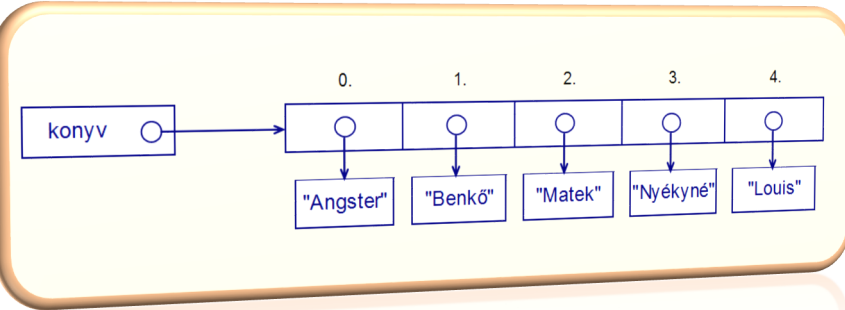
[http://hu.wikipedia.org/wiki/ISO/IEC\\_8859-2](http://hu.wikipedia.org/wiki/ISO/IEC_8859-2)

Dátum: éééé. hh. nn.

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 3'	<p>Szoktak-e olvasni?</p> <p>Milyen könyveket?</p> <p>Valamilyen szempont szerint rendezve vannak a könyvek a könyvespolcon?</p> <p>Nekem igen, külön vannak a szakkönyvek, és azon belül egymás mellett a Java, illetve a matek könyvek. A múltkor az egyik matek könyvet nem talál- tam, mert takarítás során rossz helyre, a Java könyvekhez lett betéve. Fogtam és kicseréltem őket.</p> <p>Mondjuk el azon lépéseket, hogyan cseréltem ki a két könyvet!</p> <p>Az órára a következő Java könyveket és egy matematikával kapcsolatos köny- vet viszek be, ebbe a sorrendbe elhelyezve:</p> <p>Angster Erzsébet: Objektorientált tervezés és programozás Java 1 - 2</p> <p>Benkő Tiborné, Tóth Bertalan: Együtt könnyebb a programozás Java</p> <p>Obádovics J. Gyula: Matematika</p> <p>Dirk Louis – Peter Müller: Java 5 Belépés a programozás világába</p>	<p>Motiválás A figyelem felkeltése</p> <p>Algoritmizálás</p> <p>Szemléletesség Figyelemfelhívás Szakirodalom megis- mertetése Tudományosság</p>	<p>Beszélgetés</p> <p>Frontális munka</p> <p>Elbeszélés</p> <p>Megbeszélés</p> <p>Elbeszélés</p> <p>Könyvek</p> <p>Szemléltetés</p>


IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
3 – 4'	<p>Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők): Java 2 útikalauz programozóknak</p> <p>Cserélés lépései, az algoritmus (közben végrehajtom amit mondanak):</p> <ul style="list-style-type: none"> <li>- Megfogom a matek könyvet és leteszem az asztalra.</li> <li>- Java könyvet beteszem a matek könyv helyére, (ami jelen pillanatban az asztalon van).</li> <li>- Az asztalról beteszem a matek könyvet a már levett Java könyv helyére.</li> </ul>	<p>Algoritmikus gondolkodás</p> <p>Algoritmus felismerése</p> <p>Algoritmus alkotása</p> <p>Algoritmizálás</p> <p>Szemléletesség</p>	<p>Frontális munka</p> <p>Tanulói elbeszélés</p> <p>Könyvek</p> <p>Szemléltetés</p>
4 – 5'	<p>Feladat: Készítse el a feladat programját, azaz a program cseréljen ki két könyvet!</p> <p>Mibe tároljuk a könyveket?</p> <p><i>Válasz: tömbbe tároljuk a könyveket</i></p> <p>Milyen típusú elemei lesznek a tömbnek?</p> <p><i>Válasz: String típusú elemei lesznek a tömbnek</i></p> <p>Hány elemű lesz a tömb?</p> <p><i>Válasz: 5 elemű lesz a tömb.</i></p> <p>Milyen nevet adna a file-nak?</p>	<p>Feladat kitűzése</p> <p>Adatszerkezet-választás</p> <p>Típusválasztás</p>	<p>Elbeszélés</p> <p>Frontális munka</p> <p>Megbeszélés</p>

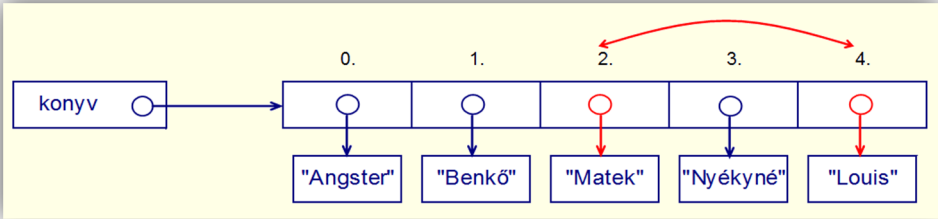
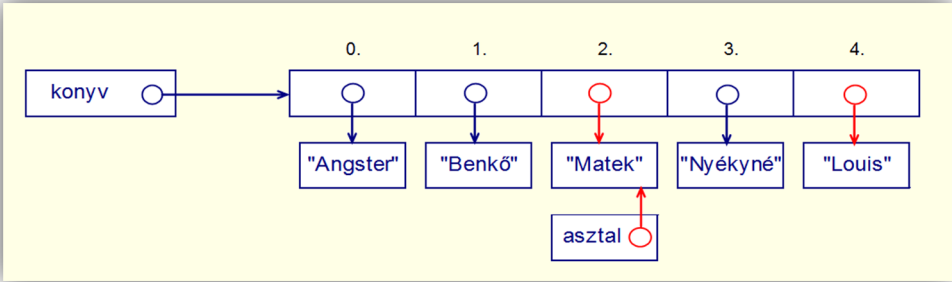
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
5 – 6'	<p>Hozzon létre a Gyakorlat13 projekt alapértelmezett csomagjába egy Elem-Csere nevű osztályt és abban egy 5 elemű String tömböt!</p> <p>Tömb létrehozása:</p> <pre>String[] konyv = new String[5];</pre>	<div>Állománykezelés</div> <div>Programkód írása</div>	<div>Egyéni munka</div> <div>Számítógép NetBeans IDE 7.3 Java 1.5</div>
6 – 8'	<p>Az újonnan létrehozott tömb elemei alapértelmezés szerinti értékeket vesznek fel. Egyszerű típusú változók esetén ezeket megtapasztaltuk.</p> <p><b>primitív</b> típusú változók esetén: <b>0, 0.0, \u0000, false</b></p> <p>Hogyan tudjuk megnézni, hogy összetett elemtípusú tömb elemeinek mik az alapértelmezett értékei?</p> <p>Ki kell íratni a tömböt.</p> <pre>// tömb kiírása for (int i = 0; i &lt; konyv.length; i++) {     System.out.print(konyv[i] + "\t"); } System.out.println();</pre> <p>Futási eredmény:</p>  <p>A memóriafoglalás:</p>	<div>Visszaemlékezés Ismétlés A szükséges előzetes ismeretek felidézése</div> <div>Motiválás A tanulók informálása a tanulási célról</div> <div>Programkód írása Precizitás, pontosság Futtatás</div> <div>Szemléletesség</div>	<div>Megbeszélés</div> <div>Frontális munka</div> <div>Számítógép NetBeans IDE 7.3 Java 1.5 Projektor</div> <div>Egyéni munka</div>

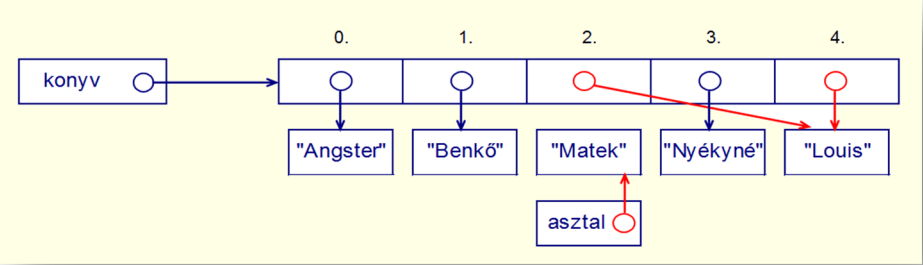
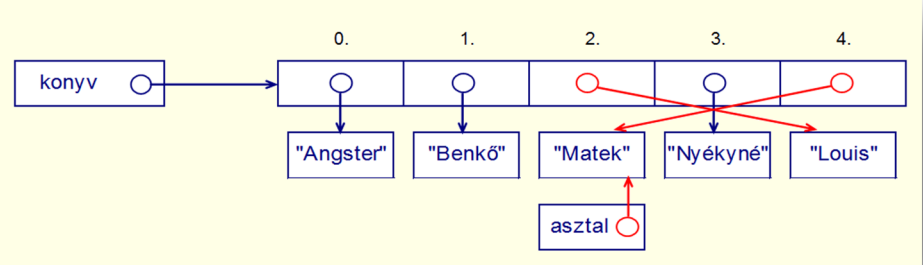
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
8 – 9'	 <p>Az alapértelmezett értékek: <b>referenciák</b> esetén: <b>null</b></p>	<p>CÉLOK ÉS FELADATOK</p> <p>Figyelem felkeltése</p> <p>Absztrakt gondolkodás</p> <p>Absztrakció, általánosítás</p> <p>Új ismeret közlése</p>	<p>Szemléltetés Tanári bemutatás Magyarázat</p> <p>Frontális munka</p> <p>Tábla</p>
9 – 10'	<p>Töltsük fel a tömböt! Bevitt könyvek alapján történik</p> 	<p>Megfigyelés</p> <p>Fokozatosság</p>	<p>Szemléltetés Tanári bemutatás</p> <p>Frontális munka</p> <p>Tábla</p>

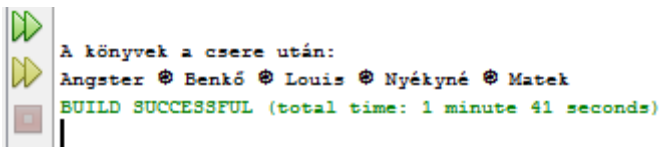
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
10 – 13'	<p>A tömb elemeit a felhasználótól kérje be!</p> <p>Kód elkészítése:</p> <pre>// tömb feltöltése beolvasással Scanner billentyuzet = new Scanner(System.in);  for (int i = 0; i &lt; konyv.length; i++) {     System.out.println("Kérem a könyvespolcon lévő "         + (i + 1) + ". könyvet! ");     konyv[i] = billentyuzet.nextLine(); }</pre>	<p>Megvalósítás</p> <p>Kódrészlet írása</p>	<p>Számítógép NetBeans IDE 7.3 Java 1.5 Projektor</p> <p>Egyéni munka</p>
13 – 15'	<p>Ékezetes karakterek beolvasásánál gond van.</p> <p>Hogyan tudnánk megoldani a problémát?</p> <p>A segítség a Java dokumentációja:</p> <p><a href="http://docs.oracle.com/javase/7/docs/api/">http://docs.oracle.com/javase/7/docs/api/</a></p> <div> <p><b>Scanner</b></p> <pre>public Scanner(InputStream source,     String charsetName)</pre> <p>Constructs a new Scanner that produces values scanned from the specified input stream. Bytes from the stream are converted into characters using the specified charset.</p> <p><b>Parameters:</b></p> <p>source - An input stream to be scanned</p> <p>charsetName - The encoding type used to convert bytes from the stream into characters to be scanned</p> <p><b>Throws:</b></p> <p>IllegalArgumentException - if the specified character set does not exist</p> </div>	<p>Problémafelvetés</p> <p>Információ gyűjtése, elemzése</p>	<p>Irányított, önálló munka</p> <p>Számítógép Internet</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
15 – 17'	<p><i>The encoding type used to convert bytes from the stream into characters to be scanned.</i></p> <p>Vagyis: A kódolás típusát használjuk a beolvasott karakterek konvertálására.</p> <p>Mi lesz vajon a kódolás típusa?</p> <p>Nézzünk utána pontosan!</p> <p><a href="http://hu.wikipedia.org/wiki/ISO/IEC_8859-2">http://hu.wikipedia.org/wiki/ISO/IEC_8859-2</a></p> <p><b>ISO-8859-2</b> az ISO/IEC 8859-es karakterkódolási szabvány második része</p> <p>Az ISO által kettes számú latin ábécének nevezett 191 karakter mindegyikének egybájtos (nyolcbites) kódját adja meg. A 191 karakter között minden magyar ékezetes betű megtalálható (a sok más készletből hiányzó <i>ő</i> és <i>ű</i> is).</p> <p>Egészítsük ki a kódot:</p>	<p>Olvasott szakmai szöveg megértése</p>	<p>Irányított, önálló munka</p>
17 – 18'	<pre>// tömb feltöltése beolvasással Scanner billentyuzet = new Scanner(System.in, "ISO-8859-2");  for (int i = 0; i &lt; konyv.length; i++) {     System.out.println("Kérem a könyvespolcon lévő "         + (i + 1) + ". könyvet! ");     konyv[i] = billentyuzet.nextLine(); }</pre> <p>Futtatás:</p> <p>most már nincs probléma a „Benkő”-vel sem.</p>	<p>Új ismereteket megalapozó tények biztosítása</p> <p>Megvalósítás Programkód átalakítása</p> <p>Futási eredmény elemzése Következtetés</p>	<p>Megbeszélés</p> <p>Számítógép Internet</p> <p>Egyéni munka</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
18 – 20'	 <p>Kérem a könyvespolcon lévő 1. könyvet! Angster Kérem a könyvespolcon lévő 2. könyvet! Benkő Kérem a könyvespolcon lévő 3. könyvet! Matek Kérem a könyvespolcon lévő 4. könyvet! Nyékyné Kérem a könyvespolcon lévő 5. könyvet! Louis</p>	Önállóság	<div>Számítógép NetBeans IDE 7.3 Java 1.5 Projektor</div>
20 – 22'	<p>A végén szeretném látni a könyvek sorrendjét a csere előtt és a csere után is, ezt hogy érem el?</p> <p><i>Válasz: Kiíratom a könyvek sorrendjét a csere előtt is és a csere után is.</i></p> <p>Kódrészlet:</p> <pre>// tömb kiírása System.out.println("\nA könyvek a csere előtt: "); for (int i = 0; i &lt; konyv.length - 1; i++) {     System.out.print(konyv[i] + " \u2746 "); } System.out.println(konyv[konyv.length - 1]);</pre>	<div>Feladatkitűzés</div> <div>Logikus gondolkodás</div> <div>Precizitás, pontosság</div>	<div>Megbeszélés</div> <div>Egyéni munka</div>
22 – 23'	<p>És most jön végre maga a csere. ☺</p> <ul style="list-style-type: none"> <li>- Megfogom a matek könyvet és leteszem az asztalra.</li> <li>- Java könyvet beteszem a matek könyv helyére, (ami jelen pillanatban az asztalon van).</li> <li>- Az asztalról beteszem a matek könyvet a már kivett Java könyv helyére.</li> </ul>	<div>Visszacsatolás Visszaemlékezés</div> <div>Megerősítés</div>	<div>Frontális munka</div> <div>Tanulói elbeszélés</div>

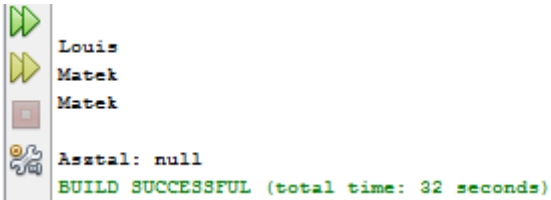
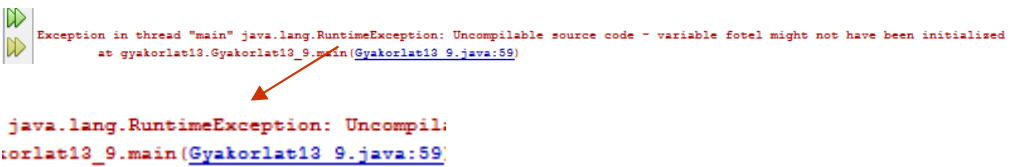
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
23 – 24'	<p>A kódolással párhuzamosan kövessük azt is, hogy „körülbelül” mi is történik a memóriában:</p>  <p>Az előző ismereteinkből tudjuk, hogy Javában: összetett típus értékadásakor a referencia állítódik át.</p>	<p>Szemléletesség</p> <p>Ismétlés Rögzítés Megerősítés</p>	<p>Frontális munka</p> <p>Tábla</p> <p>Megbeszélés</p>
24 – 25'	<p>Megfogom a matek könyvet és leteszem az asztalra.</p> <pre>String asztal = konyv[2];</pre> 	<p>Algoritmikus gondolkodás</p> <p>Absztrakt gondolkodás</p>	<p>Frontális munka</p> <p>Tábla</p> <p>Elbeszélés Magyarázat</p>

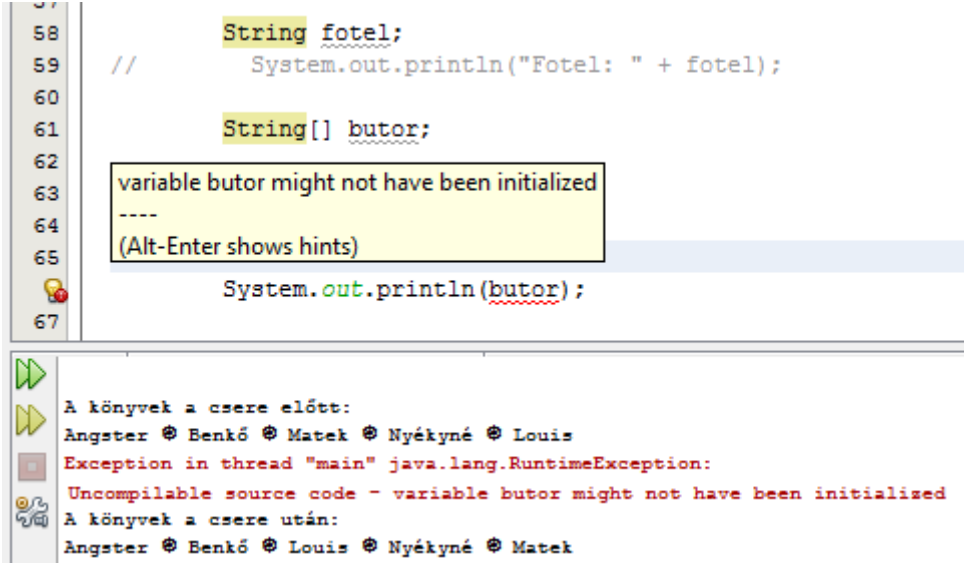
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
25 – 27'	<p>Java könyvet beteszem a matek könyv helyére.</p> <pre>konyv[2] = konyv[4];</pre> 	<p>Fokozatosság</p> <p>Logikus gondolkodás</p> <p>Megfigyelés</p>	<p>Frontális munka</p> <p>Tábla</p> <p>Megbeszélés Magyarázat</p>
27 – 28'	<p>Az asztalról beteszem a matek könyvet a már kivett Java könyv helyére</p> <pre>konyv[4] = asztal;</pre> 	<p>Tudományosság</p> <p>Aktivizálás</p>	<p>Frontális munka</p> <p>Tábla</p> <p>Tanulói elbeszélés</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
28 – 30'	<p>Kódrészlet elkészítése (a kommentekről ugye nem feledkeznek meg!):</p> <pre>// tömbelemek cseréje String asztal = konyv[2]; konyv[2] = konyv[4]; konyv[4] = asztal;</pre> <p>Kód elkészítése - csere utáni kiíratás:</p> <pre>// tömb kiíratása System.out.println("\nA könyvek a csere után: "); for (int i = 0; i &lt; konyv.length - 1; i++) {     System.out.print(konyv[i] + " \u2746 "); } System.out.println(konyv[konyv.length - 1]); }</pre>	<p>Figyelemfelhívás</p> <p>Algoritmus megvalósítása Programkód írása</p> <p>Programkód átalakítása</p>	<p>Számítógép NetBeans IDE 7.3 Java 1.5 Projektor</p> <p>Egyéni munka</p>
30 – 31'	<p>Futtatás:</p> 	<p>Futási eredmény elemzése</p>	<p>NetBeans IDE 7.3</p> <p>Egyéni munka</p>
	Az 1. feladat kódja :		

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
32 – 33'	<pre> import java.util.Scanner;  /**  *  * @author Kovács Henriette  */ public class ElemCsere {     public static void main(String[] args) {         // tömb létrehozása         String[] konyv = new String[5];          // tömb kiírása         for (int i = 0; i &lt; konyv.length; i++) {             System.out.print(konyv[i] + "\t");         }         System.out.println("\n");          // tömb feltöltése beolvasással         Scanner billentyuzet = new Scanner(System.in, "ISO-8859-2");          for (int i = 0; i &lt; konyv.length; i++) {             System.out.println("Kérem a könyvespolcom lávó "                                + (i + 1) + ". könyvét! ");             konyv[i] = billentyuzet.nextLine();         }          // tömb kiírása         System.out.println("\nA könyvek a csere előtt: ");         for (int i = 0; i &lt; konyv.length - 1; i++) {             System.out.print(konyv[i] + " \u2746 ");         }         System.out.println(konyv[konyv.length - 1]);          // tömbelemek cseréje         String aasztal = konyv[2];         konyv[2] = konyv[4];         konyv[4] = aasztal;          // tömb kiírása         System.out.println("\nA könyvek a csere után: ");         for (int i = 0; i &lt; konyv.length - 1; i++) {             System.out.print(konyv[i] + " \u2746 ");         }         System.out.println(konyv[konyv.length - 1]);     } } </pre>	<div>Önellenőrzés</div> <div>Ellenőrzés</div> <div>Rendszerezés Rögzítés Megerősítés</div> <div>Értékelés</div>	<div>Szemléltetés</div> <div>Projektor</div> <div>Elbeszélés</div> <div>Frontális munka</div>

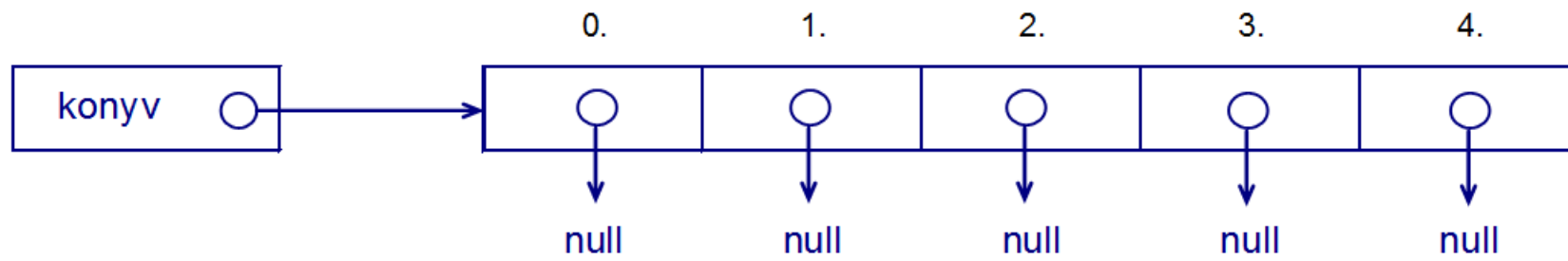
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
33 – 34'	<p>Futási eredmény:</p> <pre> run: null    null    null    null    null  Kérem a könyvespolcon lévő 1. könyvet! Angster Kérem a könyvespolcon lévő 2. könyvet! Benkő Kérem a könyvespolcon lévő 3. könyvet! Matek Kérem a könyvespolcon lévő 4. könyvet! Nyékyné Kérem a könyvespolcon lévő 5. könyvet! Louis  A könyvek a csere előtt: Angster @ Benkő @ Matek @ Nyékyné @ Louis  A könyvek a csere után: Angster @ Benkő @ Louis @ Nyékyné @ Matek BUILD SUCCESSFUL (total time: 1 minute 41 seconds) </pre>	<p>Precizitás, pontosság</p> <p>Futási eredmény átláthatósága, követhetősége</p>	<p>Megbeszélés</p> <p>Szemléltetés</p> <p>Projektor</p> <p>Frontális munka</p>
34 – 35'	<p>Feladat:</p> <p>2. a) Győződjön meg róla, hogy mely objektumra mutatnak a cserében részt vevő tömbelemek és az asztal referencia (a cserénél bevezetett segédváltozó)!</p> <p>b) Állítsa az asztal referenciát null-ra, írassa ki, nézze meg, mi történik!</p> <p>c) Deklaráljon egy fotel nevű változót, mely egy String objektumra képes mutatni, írassa ki, nézze meg, mi történik!</p>	<p>Feladat kitűzése</p> <p>Fokozatosság</p>	<p>Projektor</p> <p>Hálózat</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
35 – 38'	<p>Kód:</p> <pre> // 2. feladat System.out.println(konyv[2]); System.out.println(konyv[4]); System.out.println(asztal);  asztal = null; System.out.println("\nAsztal: " + asztal);  String fotel; // System.out.println("Fotel: " + fotel);      } } </pre>	<p>Megerősítés</p> <p>Ismétlés Új ismeretet megalapozó tények biztosítása</p>	<p>Irányított, önálló munka</p> <p>Projektor Számítógép NetBeans IDE 7.3 Java 1.5</p> <p>Megbeszélés</p>
38 – 39'	<p>Futási eredmény:</p>  <p>De, ha az utolsó utasítás nincs kikommentezve:</p>  <p><b>RuntimeException:</b> futási hiba keletkezik, a fotel változó nem lett inicializálva</p>	<p>Futási eredmény elemzése</p> <p>Olvasott szakmai szöveg megértése</p> <p>Futási hiba felismerése</p>	<p>Frontális munka</p> <p>Megbeszélés Magyarázat</p> <p>Projektor Számítógép NetBeans IDE 7.3 Java 1.5</p>

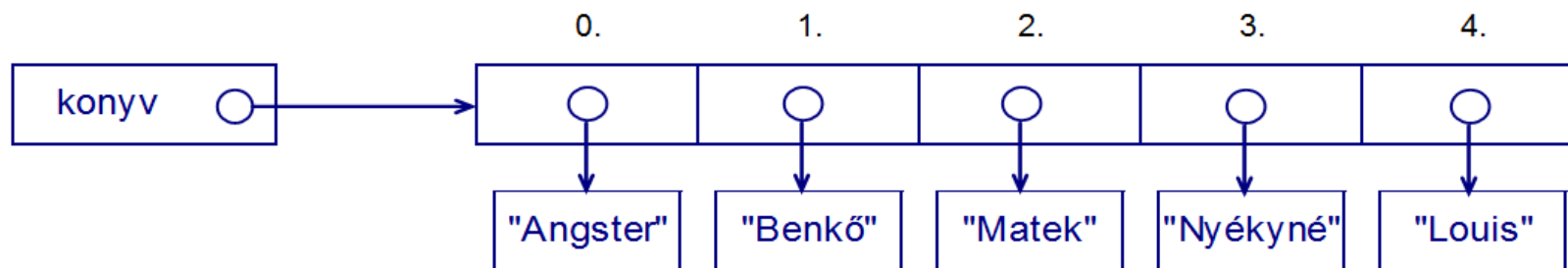
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
39 – 40'	<p>Feladat:</p> <p>d) Deklaráljon egy butor nevű változót, mely egy olyan tömb objektumra képes mutatni, melynek elemei String-ek, írassa ki, nézze meg, mi történik!</p> <pre>String[] butor;</pre> <pre>System.out.println(butor);</pre> <p>Futás:</p> 	<p>Feladat kitűzése</p> <p>Programkód írása</p> <p>Diagnosztizálás, hibakeresés</p> <p>Fejlesztői környezet jelzésének figyelembe vétele</p> <p>Futási eredmény elemzése</p>	<p>Projektor Hálózat</p> <p>Számítógép NetBeans IDE 7.3 Java 1.5</p> <p>Frontális munka</p> <p>Megbeszélés</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
40 – 41'	<p>Tehát tömböknél is ugyanaz történik, mint az előbb a fotek változónál:</p> <p><b>RuntimeException</b>: futási hiba keletkezik, a butor nevű változó nem lett inicializálva – erre a fejlesztői környezet is felhívja a figyelmünket, ezért mindig figyeljük a NetBeans IDE jelzéseit.</p>	Általánosítás	<div>Megbeszélés</div> <div>Projektor NetBeans IDE</div> <div>Frontális munka</div>
41 – 42'	<p>A tömb referenciájának alapértelmezés szerinti értéke</p> <p><b>lokális deklaráció</b> esetén: <b>definiálatlan</b></p>	<p>Következtetés, absztrakció</p> <p>Új ismeret közlése</p>	Elbeszélés
42 – 44'	<p>Ismétlés:</p> <p>Tömbelemek cseréje</p>	<p>Rögzítés</p> <p>Megerősítés</p>	Beszélgetés
44 – 45'	Tanulók munkájának minősítése	Értékelés	Megbeszélés

# TÁBLAKÉP I.

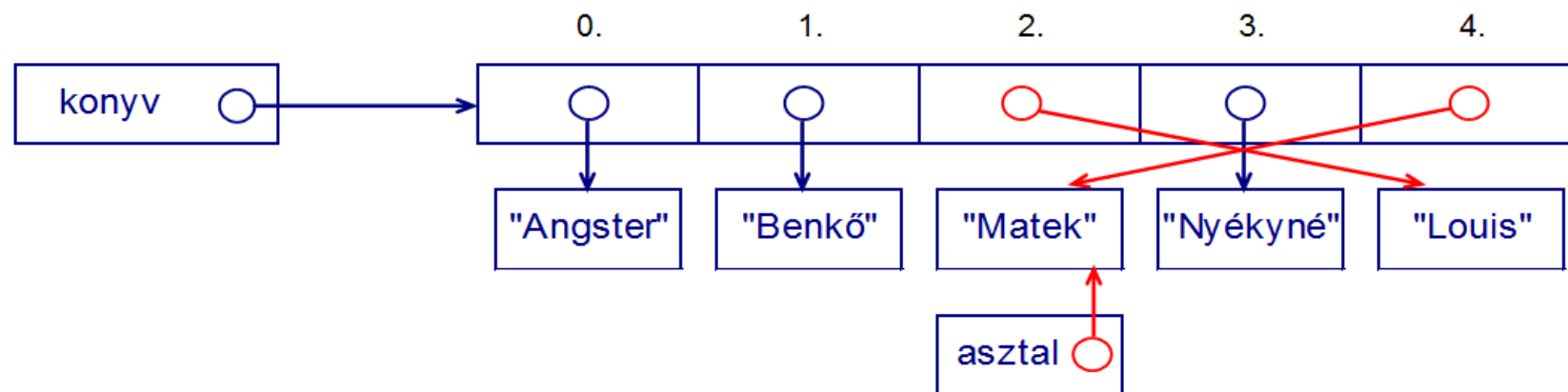


## TÁBLAKÉP II.



## TÁBLAKÉP III.

### Tömbelemek cseréje



```
String asztal = konyv[2];  
konyv[2] = konyv[4];  
konyv[4] = asztal;
```

## Feladatok

1. a) Írjon programot, mely egy könyvespolcon a harmadik és ötödik helyen lévő két könyvet kicseréli! A könyvespolcon 5 db könyv legyen, az egyes könyveket kérje be a felhasználótól (a könyveket ne egyszerű típusú változókkal azonosítsa)!
  - b) Írassa ki a könyvespolcon lévő könyveket úgy, hogy látható legyen a csere!
2. a) Győződjön meg róla, hogy mely objektumra mutatnak a cserében részt vevő tömbelemek és az asztal referencia (a cserénél bevezetett segédváltozó)!
  - b) Állítsa az asztal referenciát null-ra, írassa ki, nézze meg, mi történik!
  - c) Deklaráljon egy fotel nevű változót, mely egy String objektumra képes mutatni, írassa ki, nézze meg, mi történik!
  - d) Deklaráljon egy butor nevű változót, mely egy olyan tömb objektumra képes mutatni, melynek elemei String-ek, írassa ki, nézze meg, mi történik!

## Reflexió

Az óra egyik fő célja a tömbelemek cseréjének megvalósítása volt. Ezen ismeretanyag motiválására sikerült egy olyan hétköznapi, valóságos példát kitalálnom, mely egyben realizálta másik célomat is, mégpedig azt, hogy a diákok összetett elemtípusú tömböket használjanak. Sőt, tovább megyek, nem én közöltem velük, hogy tömböt kell létrehozniuk, és azt sem, hogy milyen típusú elemekkel töltsék fel, így hozzájárultam ahhoz, hogy fejlesszem azon képességeiket, miszerint tudjanak megfelelő adatszerkezetet és adattípust választani. A példa ezen túl kiváló alkalmat teremtett a tárgyban megjelent szakirodalom ismertetésére, bemutatására, és a könyvek bevitelével valóság közelivé tettem a problémát és szó szerint el tudtuk „játszani” az algoritmust, konkrétan megvalósítottuk a cserét.

Az algoritmikus gondolkodás fejlesztésének egyik legnehezebb komponense az algoritmus felismerése és az algoritmus alkotása. Ez az esetek többségében eddig akadályokba ütközött, de ez a hétköznapi példa megtörte a jeget és képesek voltak az algoritmus megalkotására. ☺

Az eddigiek során egy létrehozott tömb elemeinek értékadás előtti kiíratásával tapasztalták meg, hogy mik az egyszerű elemtípusú tömb elemeinek alapértelmezett értékei. Most ugyanezt tettük az összetett elemtípusú tömbbel és megtudták, hogy ebben az esetben az alapértelmezett érték a null referencia.

Ezt, és a továbbiakban a tömb elemeinek feltöltését, a tömbelemek cseréjének minden egyes lépését a szemléltetés módszerével a táblán tettem érthetővé, mely tapasztalataim alapján határozottan alkalmas arra, hogy megtudják és megértsék: az objektumok „maradnak”, csak a referencia állítódik át. A táblára általam felrajzolt ábrák mindegyike és természetesen a kódok is a hálózaton számukra elérhetők. Ezt minden egyes csoportomnál alkalmazom, de itt különösen fontos, tekintettel a csoport összetételére.

A kellő előkészítésnek köszönhetően két tanuló önállóan, egy tanuló egy kis segítséggel lekódolta az algoritmust.

Valószínű, hogy emiatt volt idő a második feladatra (nem voltam biztos, hogy ez belefér az órába, de azt az elvet vallom, hogy a van-tól nem kell félni, tehát inkább több feladat legyen a tarsolyomban, mint-hogy ott álljak üres zsebbel ☺), melyből kiderült számukra, hogy tömb referenciájának alapértelmezés szerinti értéke lokális deklaráció esetén definiálatlan.

Az óra a legjobb példája annak, hogy a megfelelően motivált diák sok mindenre képes.

## **ÓRATERVEZET - TÖMB ELEMEINEK MEGFORDÍTÁSA. GYAKORLÁS.**

**A pedagógus neve:** Kovács Henriette

**A szakmacsoport száma és megnevezése:** 7. Informatika

**A szakképesítés azonosító száma:** 54 213 05

**A szakképesítés megnevezése:** Szoftverfejlesztő

**Iskolai rendszerű szakképzés** (a szakképzési évfolyamok száma: 2)

**Szakmai követelménymodul:** 10817-12 Hálózatok, programozás és adatbázis-kezelés

**Tantárgy:** Adatbázis- és szoftverfejlesztés gyakorlat

**Programozási nyelv:** Java

**Fejlesztői környezet:** NetBeans IDE 7.3

**Osztály - csoport:** I/13. S - SZ2

**Az óra tananyaga:** Tömb elemeinek megfordítása. Gyakorlás.

### **Az óra cél- és feladatrendszere**

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt.
- Korszerű alkalmazói készség kialakítása.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- A feladatmegoldáshoz megfelelő adattípus kiválasztásának fejlesztése.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.

- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, létrehozni, feltölteni.*
- *Tudjon tömböt megjeleníteni.*
- *Legyen képes a tömbök elemeit elérni, feldolgozni és velük műveleteket végezni.*
- *Tudjon tömböt létrehozni inicializáló blokk segítségével.*
- *Legyen képes a tömbök elemeit absztrakt módon elérni.*
- *Tudjon tömbelemeket cserélni.*

### **Fejlesztendő készségek, képességek**

- Visszaemlékezés, algoritmikus gondolkodás, logikus gondolkodás, absztrakt gondolkodás, problémamegoldás, szabályalkotás, becslés, képletalkotás, algoritmus megvalósítása, struktogram elkészítése, precizitás, pontosság, az absztrakciók, általánosítások felhasználása, önellenőrzés, kód elkészítése, kommentek használata, szintaktika követése, futási eredmény elemzése, követhetősége, átláthatósága.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.

- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.

### **Az óra didaktikai feladatai**

- Ismétlés
- Megerősítés
- Motiváció
- A figyelem felkeltése
- Feladat kitűzése
- Aktivizálás
- Fokozatosság
- Szemléletesség
- Következtetés
- Összegzés
- Általánosítás
- A tények elemzése, összefüggések feltárása
- Visszacsatolás
- Összehasonlítás
- Ellenőrzés
- Értékelés
- Figyelemfelhívás

**Tantárgyi kapcsolatok:** Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek, Testnevelés

**Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):**

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyiné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

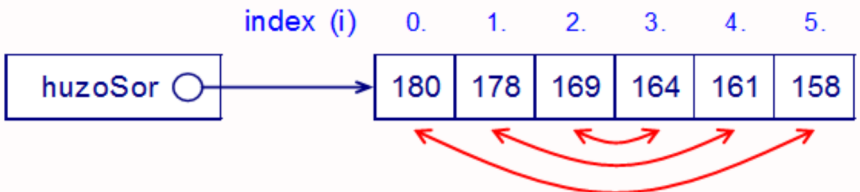
<http://docs.oracle.com/javase/7/docs/api/>

Dátum: éééé. hh. nn.

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 4'	<p>Ismétlés:</p> <p>Tömb deklarálása, létrehozása, elemeinek alapértelmezett értékei.</p> <p>Tömb elemeinek megjelenítése.</p> <p>Tömbelemek cseréje.</p> <p>A mai órán ezzel kapcsolatosan egy nagyon érdekes feladatot valósítunk meg.</p> <p>Tegnap, ahogy a tornaterem előtt elmentem, nyitva volt az ajtó és láttam, hogy pont akkor tornasorban sorakoztak a gyerekek. Eszembe jutott, hogy amikor én jártam iskolába az egyik barátnóm volt a legkisebb és mindig a tornasor végén állt. Soha nem értette, miért nem lehetne legalább egyszer fordítva állni. Vált-suk valóra álmát!</p> <p>Feladat:</p> <p>1. Egy csoport 7 diákját állítsa fordított tornasorba! A diákokat milyen jellemzőjükkal azonosítaná? Jelenítse meg a változást!</p> <p>Gondoljuk át, hogyan tudnánk megvalósítani a feladatot!</p>	<p>Ismétlés Megerősítés Visszaemlékezés</p> <p>Motiváció A figyelem felkeltése</p> <p>Feladat kitűzése</p> <p>Logikus gondolkodás</p>	<p>Megbeszélés</p> <p>Frontális munka</p> <p>Hálózat Projektor</p>

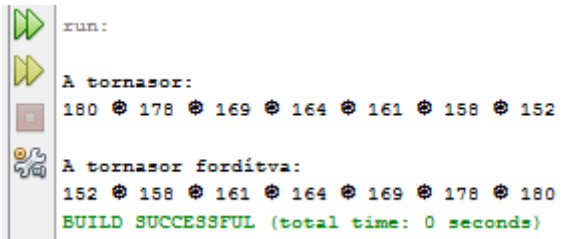
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK																		
4 – 6'	Milyen adatszerkezettel valósítjuk meg a feladatot? Válasz: tömb adatszerkezettel valósítjuk meg. Milyen tulajdonságukkal azonosítjuk a diákokat? magasság – cm Melyik módszerrel hozzuk létre a tömböt? inicializáló blokkal. Milyen típusú elemekből áll? egész	Aktivizálás Absztrakt gondolkodás Fokozatosság	Vita Megbeszélés Frontális munka																		
6 – 11'	Hogyan tudom megfordítani a tornasort? <div><p>tömb hossza (h): 7</p><p>index (i) 0. 1. 2. 3. 4. 5. 6.</p><p>tornaSor [180, 178, 169, 164, 161, 158, 152]</p><p>Indexek összege</p><table><tr><td>i</td><td>0</td><td>6</td><td>h-i-1</td><td>6</td><td>6-0=7-0-1</td></tr><tr><td>i</td><td>1</td><td>6</td><td>h-i-1</td><td>5</td><td>6-1=7-1-1</td></tr><tr><td>i</td><td>2</td><td>6</td><td>h-i-1</td><td>4</td><td>6-2=7-2-1</td></tr></table></div> Mit gondolnak, melyik az én adataim a tömbben? Egy cserét két tanuló eljátszik. Tehát a tömb i. és (h – i - 1). indexű elemeit kell kicserélni. Mely elemek ezek? tornaSor[i] és tornaSor[h – i - 1]	i	0	6	h-i-1	6	6-0=7-0-1	i	1	6	h-i-1	5	6-1=7-1-1	i	2	6	h-i-1	4	6-2=7-2-1	Problémamegoldás Szemléletesség Logikus gondolkodás Algoritmikus gondolkodás Absztrakt gondolkodás Szabályalkotás  Becslés Következtetés Összegzés Általánosítás	Szemléltetés Vita  Magyarázat  Frontális munka Tábla Hálózat  Szerepjáték
i	0	6	h-i-1	6	6-0=7-0-1																
i	1	6	h-i-1	5	6-1=7-1-1																
i	2	6	h-i-1	4	6-2=7-2-1																

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK																
11 – 13'	<p>Hány cserét kell végezni? Mindegy, hogy páratlan vagy páros elemszámú a tömb?</p> <div><p>Páratlan elemszámú tömb esetén</p><p>tömb hossza (h): 7</p><table><tr><td>index (i)</td><td>0.</td><td>1.</td><td>2.</td><td>3.</td><td>4.</td><td>5.</td><td>6.</td></tr><tr><td>tomaSor</td><td>180</td><td>178</td><td>169</td><td>164</td><td>161</td><td>158</td><td>152</td></tr></table><p>3 csere 7 / 2 = 3 (egész osztás) h / 2</p></div> <p>Páratlan elemszámú tömb esetén (tömb hossza / 2) cserét kell végezni.</p>	index (i)	0.	1.	2.	3.	4.	5.	6.	tomaSor	180	178	169	164	161	158	152	<p>CÉLOK ÉS FELADATOK</p> <p>Fokozatosság</p> <p>A tények elemzése, összefüggések feltárása</p> <p>Szemléletesség</p> <p>Visszacsatolás</p> <p>Következtetés</p> <p>Képletalkotás</p>	<p>Szemléltetés</p> <p>Vita</p> <p>Frontális munka</p> <p>Magyarázat</p> <p>Tábla Hálózat</p>
index (i)	0.	1.	2.	3.	4.	5.	6.												
tomaSor	180	178	169	164	161	158	152												

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
13 – 15'	<p>Páros elemszámú tömb esetén</p> <p>tömb hossza (h): 6</p> <p>index (i) 0. 1. 2. 3. 4. 5.</p>  <p>3 csere  <math>6 / 2 = 3</math> (egész osztás)  <math>h / 2</math></p> <p>Páros elemszámú tömb esetén (tömb hossza / 2) cserét kell végezni.  Páratlan és páros elemszámú tömb esetén is (tömb hossza / 2) cserét kell végezni.</p>	<p>CÉLOK ÉS FELADATOK</p> <p>Aktivizálás</p> <p>Összehasonlítás</p> <p>Fokozatosság</p> <p>Következtetés Képletalkotás</p> <p>Összegzés</p>	<p>Frontális munka</p> <p>Előző ábra projektorral kivetítve, ez az előzőt módosítva, a</p> <p>Tábla</p> <p>Frontális munka</p> <p>Megbeszélés Magyarázat</p>

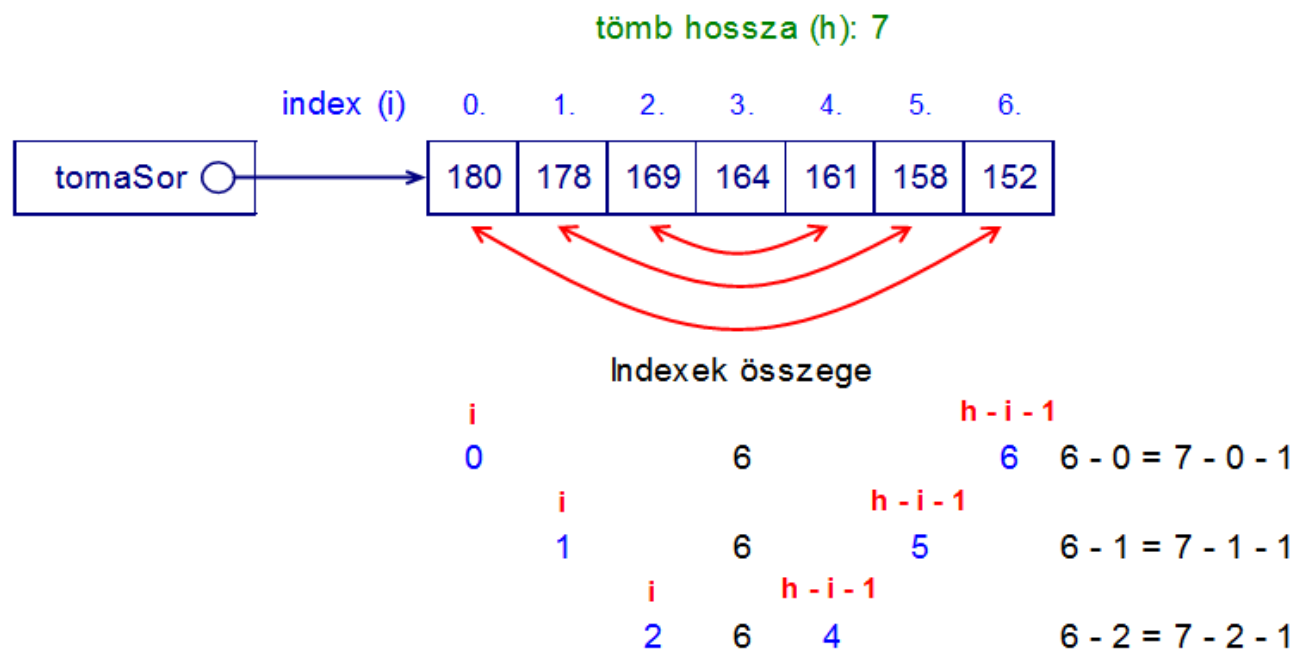
IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
15 – 28'	<p>Struktogram elkészítése:</p> <p>A hasonló vagy azonos részeket másolja és javítsa!</p>	<p>Algoritmikus gondolkodás</p> <p>Algoritmus megvalósítása Struktogram elkészítése</p> <p>Precizitás, pontosság</p> <p>Az absztrakciók, általánosítások felhasználása</p> <p>Önellenőrzés</p> <p>Ellenőrzés</p>	<p>Önálló munka</p> <p>nehézségek esetén</p> <p>Irányított, önálló munka</p> <p>(A struktogramban a tömb létrehozását, és az 1. kiíratást, ha úgy látom jónak készen megkapják.)</p> <p>Számítógép Structorizer Microsoft Office Excel 2007, 2010</p> <p>Frontális munka</p>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
28 – 40'	<p>Programkód:</p> <pre> package gyakorlat13;  /**  *  * @author Kovács Henriette  */ public class Gyakorlat13_10 {     public static void main(String[] args) {          // tömb létrehozása, feltöltése         int[] tornasor = {180, 178, 169, 164, 161, 158, 152};          //tömb kiírása         System.out.println("\nA tornasor: ");         for (int i = 0; i &lt; tornasor.length - 1; i++) {             System.out.print(tornasor[i] + " \u2746 ");         }         System.out.println(tornasor[tornasor.length - 1]);          //tömbelemek sorrendjének megfordítása         for (int i = 0; i &lt; tornasor.length / 2; i++) {             int ujHely = tornasor[i];             tornasor[i] = tornasor[tornasor.length - i - 1];             tornasor[tornasor.length - i - 1] = ujHely;         }          //tömb kiírása fordított sorrendben         System.out.println("\nA tornasor fordítva: ");         for (int i = 0; i &lt; tornasor.length - 1; i++) {             System.out.print(tornasor[i] + " \u2746 ");         }         System.out.println(tornasor[tornasor.length - 1]);     } } </pre>	<div>Kód elkészítése Kommentek használata Szintaktika követése</div> <div>Precizitás, pontosság</div> <div>Az absztrakciók, általánosítások felhasználása</div> <div>Önellenőrzés</div> <div>Ellenőrzés</div>	<div>Önálló munka</div> <div>Számítógép NetBeans IDE 7.3 Java 1.5</div> <p>(Ha nem tudnák befejezni a kódo- lást, akkor vasárnapi határidővel házi feladatnak feladom.)</p> <div>Megbeszélés</div> <div>Projektor</div> <div>Frontális munka</div>

IDŐKERET	A Z Ó R A M E N E T E	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
40 – 41'	<p>Futási eredmény:</p>  <pre> run: A tornasor: 180 178 169 164 161 158 152 A tornasor fordítva: 152 158 161 164 169 178 180 BUILD SUCCESSFUL (total time: 0 seconds) </pre> <p>Ellenőrzés – közben is folyamatosan követtem munkájukat, ha szükséges segíték.</p>	<p>Futási eredmény elemzése, követhetősége, átláthatósága</p> <p>Ellenőrzés</p>	<p>Projektor</p> <p>Frontális munka</p>
41 – 43'	Értékelés	Értékelés	Frontális munka
43 – 45'	Tájékoztatom őket, hogy jövő héten az eddigiek számonkérése lesz és az eddigi elméleti anyagról egy Tomb.pdf file-t raktam fel a hálózatra.	Figyelemfelhívás	<p>Tanári közlés</p> <p>Tomb.pdf</p>

# TÁBLAKÉP I.

## Tömbelemek cseréje

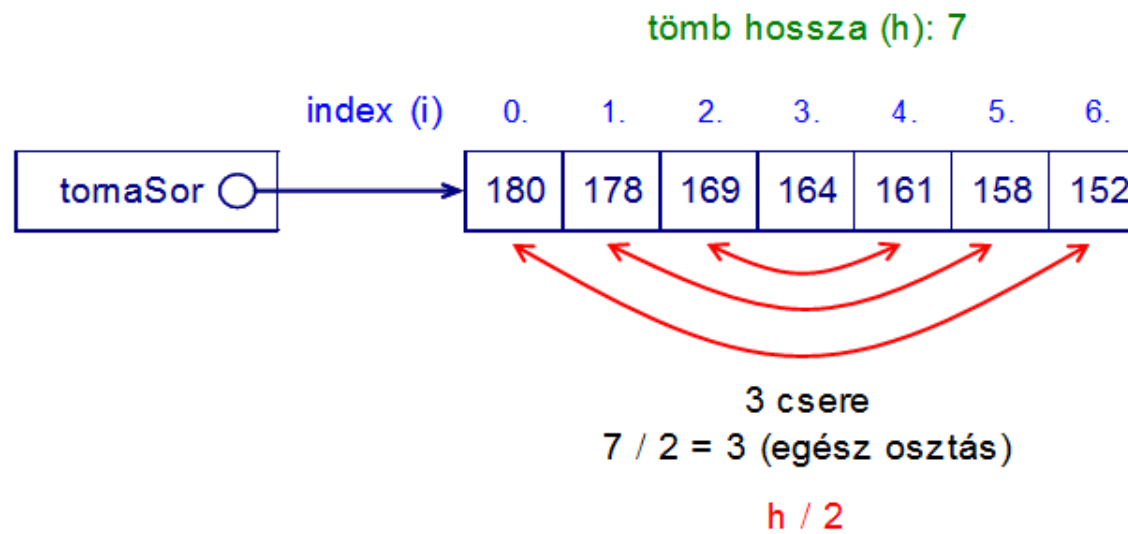


tomaSor[i] és tomaSor[h - i - 1] elemek cseréjét kell elvégezni

## TÁBLAKÉP II.

Cserék száma

Páratlan elemszámú tömb esetén

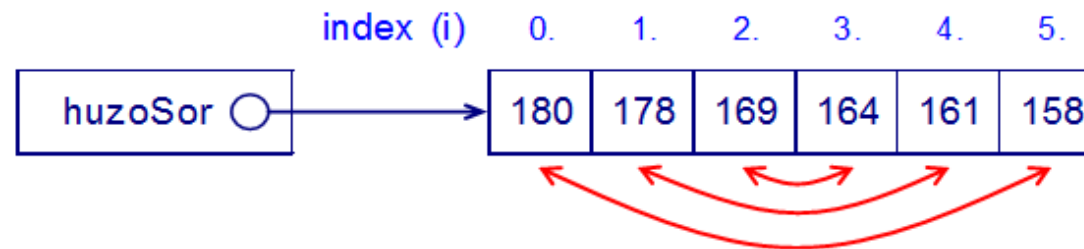


## TÁBLAKÉP III.

Cserék száma

Páros elemszámú tömb esetén

tömb hossza (h): 6



3 csere

$6 / 2 = 3$  (egész osztás)

$h / 2$

## Feladat

Egy csoport 7 diákját állítsa fordított tornasorba! A diákokat milyen jellemzőjükkel azonosítaná? Jelenítse meg a változást! (Ha szükséges, a lehető legáltalánosabban alkosson szabályokat!)

A feladathoz készítsen algoritmust!

Fix elemszámú, azonos típusú elemekből álló összetett adatszerkezet.

Az elemek

- indexelhetők, az index minden esetben 0-ról induló egész szám.
- lehetnek primitív vagy referencia típusúak.
- az elemek deklarált típusa azonos, de a referencia típusú változók utód típusú objektumokat is azonosíthatnak.

(Alacsony szintű konténer, melynek csak adatai vannak, funkciói nincsenek.)

Használatában az a nagyszerű, hogy nem kell az elemeknek egyedi neveket kitalálni, hanem egy gyűjtőnevet adunk az egész csoportnak, és az egyes elemekre indexekkel hivatkozunk.

A tömb referencia típusú változó, melyet **deklarálnunk** kell.

1. elemek típusa
2. tömbképző operátor: szögletes zárójelpár
3. tömb azonosítója

ide is tehetjük!

```
<elemtípus>[] <tömbAzonosító>;
<elemtípus> <tömbAzonosító>[];
```

```
pl.:    int[] iArray1;
        int iArray2[];
```

Feladat: Melyik tömb a következő közül?

```
int[] iArray1, iArray2;
int iArray3[], iArray4;
```

Az így deklarált változó képes egy megadott elem típusú tömbre mutatni.

A változóban az objektum memóriabeli címét tároljuk.  
Deklaráció hatására nem jön létre objektum.

A deklarálással csak a referenciának foglalunk helyet, ezen a ponton még nincs tömb.

A tömböt külön **létre** kell **hoznunk!**

new operátorral hozhatjuk létre a program futása közben

1. new operátor
2. elemek típusa
3. tömb mérete (elemek száma): nem negatív, int típusú kifejezés.  
maximum `Integer.MAX_VALUE` eleme lehet e tömbnek.

```
new <elemtípus>[<méret>];
```

Feladat: Mennyi a mérete, mi az elemtípusa?

```
new int[5];           // 5, int
int k = 20;
new double [k/3+1];   // 7, double
new String[3];        // 3, String
```

new operátor: létrehozza a tömböt a memóriában

visszaadja a tömb referenciáját (memóriabeli mutatóját)

Ezt a referenciát aztán értékül adhatjuk egy olyan referencia típusú változónak, amely egy ilyen tömbre képes rámutatni.

<pre>int[] iArray;</pre>	}	deklaráció	}	1 lépésben is elvégezhetők
<pre>double[] dArray;</pre>				
<pre>String[] sArray;</pre>				
<pre>iArray = new int[5];</pre>	}	létrehozás		
<pre>dArray = new double[k/3+1];</pre>				
<pre>sArray = new String[3];</pre>				
<pre>int[] iArray = new int[5];</pre>				
<pre>double[] dArray = new double[k/3+1];</pre>				
<pre>String[] sArray = new String[3];</pre>				

Minden tömb objektumnak van egy length konstansa, amely megadja a tömb hosszát.

`<tömbAzonosító>.length`

A length értéke a tömb létrehozásától állandó, nem lehet megváltoztatni.

A tömb indexei 0 és length-1 közöttiek.

A tömb egyes elemeire így hivatkozhatunk:

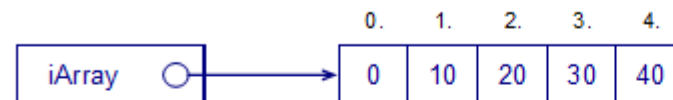
`<tömbAzonosító>[index]`

Az érvényes indextartományon kívül eső indexelés

`ArrayIndexOutOfBoundsException` kivételt vált ki.

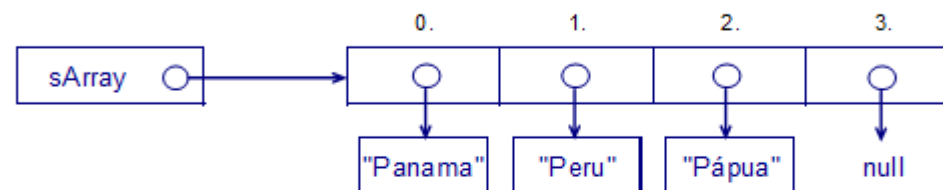
Tömb elemeinek feltöltése:

```
int[] iArray = new int[5];           //0, 10, 20, 30, 40-nel
for (int i = 0; i < iArray.length; i++)
    iArray[i] = i * 10;
```



```
-----
double[] dArray = new double[20]; // beolvasás konzolról
for (int i = 0; i < dArray.length; i++){
    System.out.println("Írd be a tömb" + (i + 1) + ". elemét!");
    dArray[i] = billentyuzet.nextDouble();
    billentyuzet.nextLine();
}
```

```
String[] sArray = new String[4];    //Az elemek referenciák
sArray[0] = "Panama";
sArray[1] = "Peru";
sArray[2] = "Pápua";
```



#### Inicializáló blokk:

A tömb elemeinek kezdeti értékek adhatók

```
<elemtípus>[] <tömbAzonosító> = {<érték0>, <érték1>, ...};
```

A tömb mérete akkora lesz, amennyi a felsorolt értékek száma.

```
int[] iArray = {0, 10, 20, 30, 40};           //5 elemű
char[] mgh = {'a', 'e', 'i', 'o', 'u'};       //5 elemű
boolean[] szep = {true, false};               //2 elemű
```

#### Kezdeti értékek:

A tömb referenciájának alapértelmezés szerinti értéke

**osztályszintű deklaráció** esetén: **null**

**lokális deklaráció** esetén: **definiálatlan**

Az újonnan létrehozott tömb elemei alapértelmezés szerinti értékeket vesznek fel.

**referenciák** esetén: **null**

**primitív** típusú változók esetén: **0, \u0000, false**

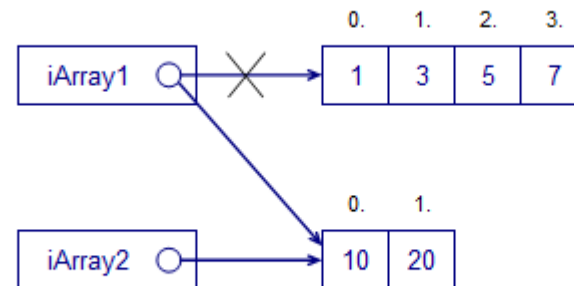
#### Értékadás:

A **t2** tömb értékadás szerint kompatibilis a **t1** tömbbel (**t1 = t2** megengedett), ha

- primitív elemtípus esetén **t1** és **t2** elemtípusa azonos
- referencia elemtípus esetén  
**t2** elemtípusa **t1** elemtípusával azonos, vagy annak leszármazottja.

Tömb értékadásakor a tömb referenciája kap értéket, nem a tömb elemei. Így különböző hosszúságú tömbök is értékül adhatók egymásnak.

```
pl.:    int[] iArray1 = {1, 3, 5, 7};
        int iArray2[] = {10, 20};
        iArray1 = iArray2;           //{10, 20}
```



Mindkét mutató  
ugyanazt a tömböt  
azonosítja.

```
double[] dArray1 = {1, 2, 3, 4, 5, 6, 7.0}, dArray2 = {1.5, 6.0};
Object[] oArray = null;
String[] sArray = {"hétfő", "kedd"};
dArray1 = dArray2;           //{1.5, 6.0}
oArray = sArray;             //{ "hétfő", "kedd" }
```

Helytelen értékadások:

```
iArray1 = dArray1;           //nem egyforma primitív típusú elemek
sArray = dArray1;            //egyik elemei objektumok, másiké primitívek
sArray = oArray              //Object-nek nem őse a String!
```

Értékdáskor a tömb nem másolódik, hanem átállítódik a mutatója.

## Reflexió

Az első óra volt a téma elkezdése óta, mely tisztán gyakorló óra és új ismeretek elsajátítására nem került sor. Ezek az órák - így ez is - igazán alkalmasak az addig megtanított ismeretek rögzítése, elmélyítése, megerősítése mellett a logikus gondolkodás, az algoritmikus gondolkodás és az absztrakt gondolkodás fejlesztésére. Központi szerepet töltött be az órán a következtetés, az absztrakció, az általánosítás, mint didaktikai feladat megvalósulása, mely maga után vont az összefüggések felismerésének és a szabályalkotás képességének fejlesztését.

A bevezető elbeszélésem megfelelően motiválta a diákokat, mert egy konkrét tevékenység megvalósítása, egy tényleges cél elérése mindig sokkal jobban felkelti érdeklődésüket és úgymond értelmét látják, mintha pusztán csak annyit mondtam volna, hogy „fordítsuk meg egy tömb elemeit”. A figyelem felkeltése - és ezen túl - a figyelem ébren tartása, az egyik legfontosabb feladatok közé tartozik, melyek az órán megvalósultak. A csoport kis létszáma és a csoportprofilban leírt okok miatt változatos munkaformák alkalmazhatósága korlátozott, de érdekes oktatási módszereket, mint pl. vita, játék ebbe az órába be tudtam vonni. Emellett nagy erőfeszítéseket tettem a szemléletesség, mint didaktikai feladat megvalósítására, mely nemcsak az adott probléma

megértését, hanem a figyelem ébren tartását és ezen felül a következtetések, általánosítások, absztrakciók megkönnyítését is szolgálták.

A tábla, a számítógép és a projektor elengedhetetlenek voltak és azt szolgálták, hogy a diákok számára élvezetessé tegyem az órát. A lényeges, magyarázó, a megértést szolgáló táblai képeket elektronikusan is elkészítettem, így a tömbelemek cseréje és a cserék számával kapcsolatos képek is nagymértékben hozzájárultak a szabályok, képletek, összefüggések feltárásához, megértéséhez, segítették a logikus, absztrakt, valamint algoritmikus gondolkodás fejlesztését és mindemellett lehetővé teszik - szükség esetén - az összefüggések újbóli bemutatását, elmélyítését.

A megbeszéltek alapján a struktogram elkészítését önálló feladatnak szántam, de éreztem, hogy valószínű, hogy ez kisebb akadályokba ütközik, ezért is tüntettem fel az óratervben és sejtésem beigazolódott. A programkód írását viszont szépen teljesítették, használva a már megírt kódrészleteket, és a futási eredmény átláthatóan jelent meg a képernyőn.

Az órát eredményesnek ítélem meg, a diákok képességei fejlődtek, az ismeretek elmélyültek.

## **FOGLALKOZÁS TERVEZETE – AZ ISMÉTLÉS - ITERÁCIÓ. CIKLUSOK FAJTÁI, ÁTALAKÍTHATÓSÁGUK, ALKALMAZÁSUK.**

**A pedagógus neve:** Kovács Henriette

**Az ágazat megnevezése - besorolása:** Informatika és távközlés - 12

**A szakma azonosító száma:** 5 0613 12 03

**A szakma megnevezése:** Szoftverfejlesztő és -tesztelő

**Szakmai oktatás időtartama:** 2 év

**A szakma Európai Képesítési Keretrendszer szerinti szintje:** 5

**A szakma Magyar Képesítési Keretrendszer szerinti szintje:** 5

**Digitális Kompetencia Keretrendszer szintje:** 8

**Tantárgy:** Programozási alapok

**Programozási nyelv:** Python

**Fejlesztői környezet:** Visual Studio Code

**Osztály:** 1/13. RL

**A foglalkozás tananyaga:** Az ismétlés - iteráció. Ciklusok fajtái, átalakíthatóságuk, alkalmazásuk.

## A foglalkozás cél- és feladatrendszere

**A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:**

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a programozás iránt.
- Korszerű alkalmazói készség kialakítása.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- A feladatmegoldáshoz megfelelő adattípus kiválasztásának fejlesztése.
- Legyen képes a Visual Studio Code fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.

- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon változókat azonosítani, értékadást végezni.*
- *Tudjon kifejezéseket az aritmetikai, logikai, relációs, értékadó operátorok felhasználásával létrehozni.*
- *Tudjon a print() függvénnyel, annak paraméterezésével adatokat megjeleníteni.*
- *Tudja használni az input() függvényt és a típuskonverziót.*
- *Tudja alkalmazni az escape karaktereket, megjegyzéseket.*
- *Tudjon egy- és többágú elágazásokat létrehozni és használni.*
- *Tudjon ciklusokat létrehozni, adott feladatnál célszerűen közülük választani.*
- *Legyen képes a végtelen ciklus felismerésére, elkerülésére.*
- *Tudjon a range() függvény segítségével sorozatot létrehozni.*
- *Legyen képes egy sorozat elemeit bejárni és velük műveleteket végezni.*

## Fejlesztendő készségek, képességek

- Visszaemlékezés, precizitás, megfigyelés, feladatértelmezés, absztrakt gondolkodás, logikus gondolkodás, algoritmikus gondolkodás, problémamegoldás, programkód analízisa, írása, szintaktika követése, hibakeresés, tesztelés, önellenőrzés, önállóság, együttműködő képesség, meggyőzőképesség, szakmai nyelvű

beszédkészség, íráskészség, mondatszerű leírás elkészítése, rendszerező képesség.

### **Fejlesztendő attitűd**

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a tevékenységek algoritmizálható részleteit.
- Legyen képes adatok megkülönböztetésére, kezelésére, használatára.
- Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Python programnyelven implementálni.
- Legyen képes értelmezni és javítani a Visual Studio Code által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.

### **Az óra didaktikai feladatai**

- Motiválás
- Aktivizálás
- Szemléletesség
- Ismétlés
- Visszacsatolás
- Új ismeret közlése
- A tények, jelenségek, folyamatok sokoldalú elemzése
- Rögzítés
- Gyakorlás
- Fokozatosság
- Megerősítés
- Ellenőrzés
- Értékelés
- Összefoglalás
- Rendszerezés
- A házi feladat kijelölése

**Tantárgyi kapcsolatok:** Matematika, Angol, Informatikai alapismeretek, Sport

**Felhasznált források:**

Szerző(k)	Cím
Gérard Swinnen	Tanuljunk meg programozni Python nyelven
Raphael Marvie	Bevezetés a Pythonba példákkal
Full Circle magazin	Programozzunk Pythonban
Takács Gábor	A Python programozási nyelv - jegyzet

[https://github.com/nlharri/Pythonarium/blob/master/Documentation/hu/01\\_introduction.md](https://github.com/nlharri/Pythonarium/blob/master/Documentation/hu/01_introduction.md)

<https://infopy.eet.bme.hu/>

<https://www.w3schools.com/python/>

<https://pythonidomar.wordpress.com/python-parancsok-magyar-jelentes/>

<https://www.tutorialspoint.com/python/>

[http://alea.uw.hu/c/Progr\\_tetelekRendezesek.htm](http://alea.uw.hu/c/Progr_tetelekRendezesek.htm)

<https://medium.com/pythonarium/halad%C3%B3-python-els%C5%91-r%C3%A9sz-cb0a898faa2e>

<https://gyires.inf.unideb.hu/EFOP344/PythonHTML/conditionals.html>

<https://sulipython.fandom.com/hu/wiki/Ciklus?action=edit&section=1>

<http://slpython.blogspot.com/2016/03/05elagazasokciklusokmd.html>

<http://people.ubuntu.com/~kelemeng/ufp3/native-datatypes.html>

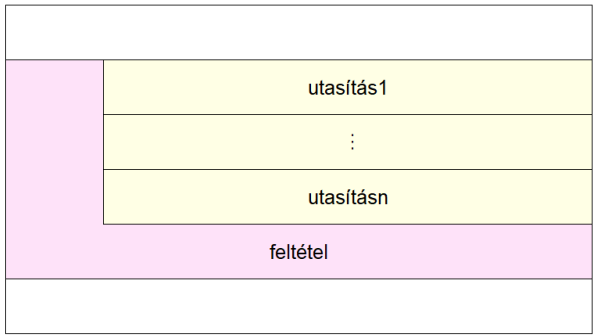
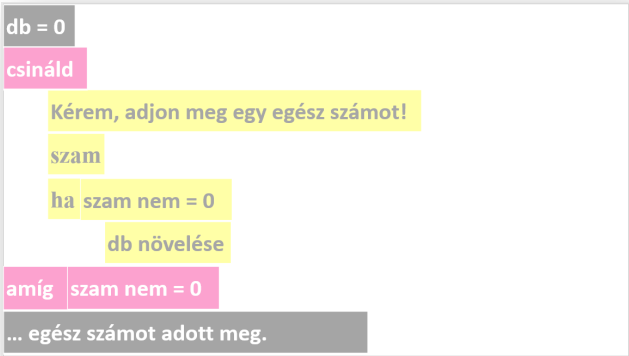
Dátum: éééé. hh. nn.

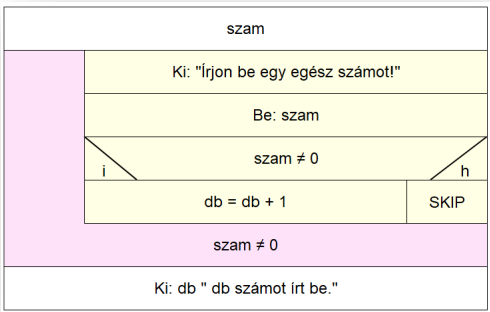
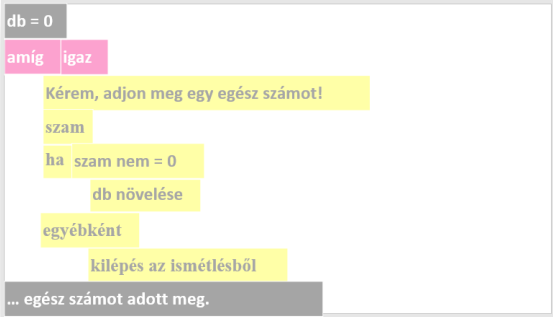
IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 5'	<p>Tanulók üdvözlése ☺, névsor olvasása – kártyák kiosztásával</p> <p>Házi feladatok, szorgalmi feladatok elkészítésével kapcsolatos észrevételek</p> <p>Ismétlés:</p> <p>Ciklus szükségessége, létrehozása.</p> <p>Ciklus működésének elve.</p> <p>Ciklus kulcsszava, a blokk lefutásának feltétele.</p> <div data-bbox="313 810 884 1198"> </div> <pre data-bbox="974 1061 1220 1268"> while feltétel:     utasítás1     utasítás2     ...     utasításn </pre>	<p>Ismétlés Megerősítés Visszaemlékezés</p> <p>Szemléletesség</p> <p>Esztétikai érzék</p>	<p>Megbeszélés Frontális munka</p> <p>Struktogramok_alapok.xlsx</p> <p>Projektor</p> <p>Python_07_het_foglalkozas.pdf</p>

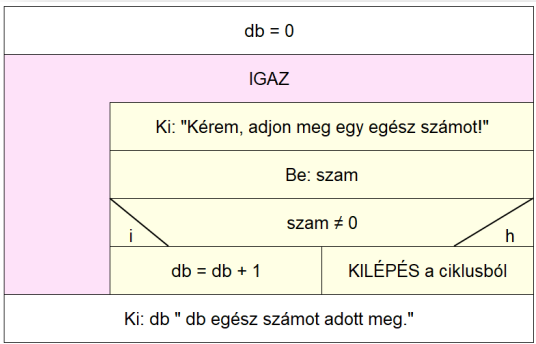


IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
10 – 13'	<p>A db változó (üres a kártyán kezdetben) hova kell, hogy kerüljön?</p> <p>Melyik kódrészlet ismétlődik?</p> <p>Miért van szükség arra, hogy a ciklus előtt is bekérjük a változót?</p> <p><i>Válasz: A feltétel a felhasználó által megadott számot vizsgálja.</i></p> <p>Mikor fut le a ciklus blokkja?</p> <p><i>Válasz: Ha a ciklus feltétele igaz</i></p> <p>Hányszor ismétlődik a ciklusmag (sárga szín)?</p> <p><i>Válasz: Nem tudjuk megmondani.</i></p> <p>Struktogrammal:</p> <div data-bbox="421 823 1086 1311" data-label="Diagram"> <pre> graph TD     Start([db = 0]) --&gt; LoopEntry(( ))     LoopEntry --&gt; LoopBody     subgraph LoopBody [szam ≠ 0]         direction TB         L1[db = db + 1]         L2[Ki: "Kérem, adjon meg egy egész számot!"]         L3[Be: szam]     end     LoopBody --&gt; LoopEntry     LoopEntry --&gt; End([Ki: db " db egész számot adott meg."]) </pre> </div>	<p>Aktivizálás Fokozatosság</p> <p>Visszacsatolás</p> <p>Problémamegoldás</p> <p>Szemléletesség Logikus gondolkodás Algoritmikus gondolkodás Absztrakt gondolkodás</p>	<p>Vita Megbeszélés</p> <p>Frontális munka</p> <p>Szemléltetés</p> <p>Struktogramok_alapok.xlsx</p> <p>Önálló munka</p>

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
13 – 18	<p>Forráskód elkészítése:</p> <p>Kiíratás és bekérés egymásba ágyazása – paraméter</p> <pre>db = 0 szam = int(input("Kérem, adjon meg egy egész számot! ")) while szam != 0:     db += 1     szam = int(input("Kérem, adjon meg egy egész számot! ")) print(db, "db egész számot adott meg.")</pre>	<p>Kód elkészítése Ismétlés</p> <p>Kommentek használata Szintaktika követése Precizitás, pontosság Önellenőrzés Futási eredmény elem- zése, követhetősége, átláthatósága</p>	<p>Önálló munka</p> <p>Visual Studio Code</p>
18 – 20'	<p>Keressünk egy egyszerűbb módszert adatok bekérésére, ne kelljen kétszer ugyanazt leírunk.</p> <p>Ha a feltétel a ciklus blokkja után lenne, nem lenne szükség a ciklus előtti be- kérésre.</p> <p>Ez az ún. hátultesztelő ciklus, melynek blokkja egyszer mindenképpen lefut.</p>	<p>Problémamegoldás</p> <p>Szemléletesség</p> <p>Új ismeret közlése</p>	<p>Számítógép Visual Studio Code Python 3.</p>

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
		<div>Szemléletesség</div> <div>Esztétikai érzék</div>	<div>Struktogramok_alapok.xlsx</div>
	<p>Algoritmizálás mondatszerű leírással</p> 	<div>Szemléletesség</div> <div>Megfigyelés</div> <div>Fokozatosság</div>	<div>Szemléltetés Tanári bemutatás Magyarázat</div> <div>Frontális munka</div>

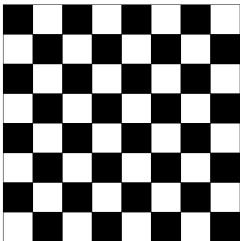
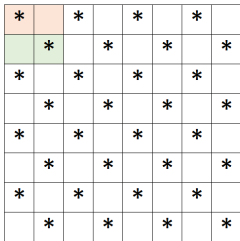
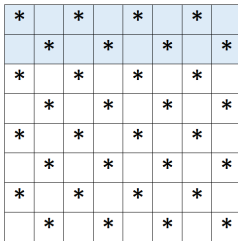
IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>Struktogrammal:</p> 	<p>Szemléletesség</p>	<p>Szemléltetés Tanári bemutatás Magyarázat</p> <p>Struktogramok_alapok.xlsx</p>
20 – 23'	<p>Pythonban nincs.</p> <p>Oldjuk meg, hogy ne használjuk a feltételben a bekért számot! – Végtelen ciklus</p> 	<p>Új ismeret közlése</p> <p>Visszacsatolás</p> <p>Ismétlés A szükséges előzetes ismeretek felidézése az új ismeretek elsajátításához</p>	<p>Magyarázat Megbeszélés</p> <p>Szemléltetés</p>

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>Struktogrammal:</p>  <p>Alakja:</p> <pre> while True:     utasítás     if feltétel:         utasítás  while True:     utasítás1     utasítás2      if feltétel:         utasítás      utasításn </pre>	<p>Szemléletesség</p> <p>Új ismeret közlése Absztrakció</p>	<p>Magyarázat</p> <p>Struktogramok_alapok.xlsx</p> <p>Python_07_het_foglalkozas.pdf</p>
23 – 28'	<p>Forráskód elkészítése:</p> <pre> db = 0 while True:     szam = int(input("Kérem, adjon meg egy egész számot! "))     if szam != 0:         db += 1     else:         break print(db, "db egész számot adott meg.") </pre> <p>Hogyan tudjuk kikerülni a break-et? <i>Logikai változó bevezetésével.</i></p>	<p>Megvalósítás</p> <p>Kódrészlet írása</p> <p>Fokozatosság</p>	<p>Számítógép</p> <p>Önálló munka</p> <p>Visual Studio Code</p>

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<pre> db = 0 szeretnenkUjBekerest = True while szeretnenkUjBekerest:     szam = int(input("Kérem, adjon meg egy egész számot! "))     if szam != 0:         db += 1     else:         szeretnenkUjBekerest = False print(db, "db egész számot adott meg.") </pre>	<div>Megvalósítás</div> <div>Programkód átalakítása</div>	<div>Frontális munka</div> <div>Visual Studio Code</div>
28 – 29'	<p>Mikor használjuk a True logikai értéket tartalmazó előtesztelő ciklust?</p> <p><i>Adott tulajdonságú adatok bekérésére nagyon hasznos.</i></p>	<div>Rendszerezés</div> <div>Aktivizálás</div> <div>Megerősítés</div>	<div>Beszélgetés</div>
29 – 30'	<p>1. feladat - Python_07_het_foglalkozas.pdf 21. feladata</p> <p>Írjon programot, ami bekér a felhasználótól egy pozitív számot! Ha a felhasználó nem a feltételnek megfelelő számot ad meg, ismétlje meg a bekérést! Ha a szám megfelelő, írja ki a képernyőre a szám 5-ös maradékát!</p>	<div>Gyakorlás</div> <div>Fokozatosság</div>	<div>Python_07_het_foglalkozas.pdf</div>
30 – 35'	<p>Megfelelő típuskonverzió alkalmazása</p> <p>Kód írása:</p> <pre> while True:     szam = int(input("Kérem, adjon meg egy pozitív számot! "))     if szam &gt; 0:         print("A/az {} ötös maradéka: {}".format(szam, szam % 5))         break </pre> <p>Futási eredmény, hibajavítás</p>	<div>Megfelelő adattípus kiválasztása</div> <div>Visszacsatolás</div> <div>Önellenőrzés</div>	<div>Megbeszélés</div> <div>Önálló munka</div> <div>Visual Studio Code</div> <div>Frontális munka</div>

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
35 - 38	<p>Alakítsuk át úgy a kódot, hogy a felhasználónak jelezzük, ha nem megfelelő számot adott meg!</p> <p>Kód másolása, átalakítása, futási eredmény</p> <pre>while True:     szam = int(input("Kérem, adjon meg egy pozitív számot! "))     if szam &lt;= 0:         print("Nem pozitív számot adott meg!")     else:         print("A/az {} ötös maradéka: {}".format(szam, szam % 5))         break</pre>	<p>Kód átalakítása</p> <p>Ellenőrzés</p>	
38 – 39'	<p>Visual Studio Code kényelmi funkciója:</p> <p>Több azonos sor egyszerre történő beírása</p> <p><span>Ctrl</span> + <span>Alt</span> + <span>Le kurzormozgató</span></p> <pre>szam1 = int(input("Kérem, ....")) szam2 = int(input("Kérem, ...."))</pre>	<p>Fejlesztői környezet előnye</p>	<p>Bemutató</p>
39 – 40	<p>2. feladat - Python_07_het_foglalkozas.pdf 27. feladata</p> <p>Generáljon le egy véletlen egész számot az [1; 12[ intervallumban, és írja ki a képernyőre a számot, valamint a szám 3-as maradékát! Ezek után kérdezze meg a felhasználót, hogy szeretne-e újabb számot! Ha igen, ismétlje meg az előzőt, ha nem, lépjen ki a programból! A felhasználó választását (i/n) egy változóban tárolja el!</p>	<p>Feladat kitűzése</p>	<p>Python_07_het_foglalkozas.pdf</p>

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
40 - 43	<p>Véletlenszámok generálása</p> <p><code>import random</code> – a random modul importálása</p> <p><code>random.randrange(alsó határ, felső határ)</code> visszaad egy véletlen egész számot az [alsó határ; felső határ [ balról zárt intervallumban pl.: <code>random.randrange(3, 10)</code> → a lehetséges értékek: 3, 4, 5, 6, 7, 8, 9,</p> <p><code>random.randrange(alsó határ, felső határ, növekmény)</code> a 3. paraméter egy növekményt megadó egész szám, alpertelmezetten 1 pl.: <code>random.randrange(3, 10, 2)</code> → a lehetséges értékek: 3, 5, 7, 9</p> <p><code>random.randint(alsó határ, felső határ)</code> visszaad egy véletlen egész számot az [alsó határ; felső határ] zárt intervallumban pl.: <code>random.randint(0,3)</code> → a lehetséges értékek: 0, 1, 2, 3</p> <p><code>random.random()</code> véletlenszerű valós számot (float) ad vissza a [0; 1] zárt intervallumban</p> <p><code>random.uniform(alsó határ, felső határ)</code> véletlenszerű valós számot ad vissza az [alsó határ; felső határ] zárt intervallumban pl.: <code>random.uniform(20, 60)</code></p> <p>A feladat megoldása során melyik függvényre lesz szükségünk? Válasz: <code>randrange()</code> Hány paraméterrel? Válasz: 2</p>	<p>Visszacsatolás</p> <p>Megerősítés</p> <p>Rögzítés</p> <p>Alkalmazás</p>	<p>Projektor</p> <p>Python_07_het_foglalkozas.pdf</p> <p>Frontális munka</p> <p>Megbeszélés</p>

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
43 – 50'	<p><b>Kód:</b></p> <pre>print("\n\n27. feladat\n") print("Véletlenszámot generálunk az [1; 12[ intervallumban.") while True:     veletlenEgesz = random.randrange(1, 12)     print("A véletlen egész szám", veletlenEgesz, ", a hármas maradéka:", veletlenEgesz % 3)     ch = input("Szeretne egy újabb véletlenszámot? ")     while not(ch == 'i' or ch == 'n'):         ch = input("Nem megfelelő karaktert ütött le! Adjon meg 'i' vagy 'n' karaktert!")     if ch == "n":         break print("VÉGE!")</pre> <p>ch != 'i' and ch != 'n' De Morgan azonosság  <a href="https://hu.wikipedia.org/wiki/De_Morgan-azonoss%C3%A1gok">https://hu.wikipedia.org/wiki/De_Morgan-azonoss%C3%A1gok</a></p> <p>Futási eredmény</p>	<p>Programkód írás Szintaktika követése Precizitás, pontosság Strukturáltság</p> <p>Alkotó képesség</p> <p>De-Morgen azonosságok alkalmazása</p> <p>Önellenőrzés</p>	<p>Egyéni munka</p> <p>Ha szükséges:</p> <p>Frontális munka</p> <p>Internet</p>
50 – 54'	<p>3. feladat - Python_07_het_foglalkozas.pdf 2/2. feladata</p> <p>Rajzoljon le egy 8 * 8-as sakktáblát csillagokból és szóközőkből!</p> <div>    </div>	<p>Algoritmikus, absztrakt. logikus gondolkodás</p> <p>Esztétikai érzék</p>	<p>Python_07_het_foglalkozas.pdf SEGEDanyagok_Gyakorlat_07.xlsx Sakk munkalap</p> <p>Projektor</p>

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK																																																		
54 – 60'	<p>Kód:</p> <pre>for i in range(4):     for j in range(4):         print("* ", end="")     print()     for j in range(4):         print(" ", end="")     print() print()</pre> <p>Futási eredmény</p>	<div>Programkód készítése Szintaktika követése Strukturáltság</div> <div>Önellenőrzés → Hibajavítás</div>	<div>Egyéni munka vagy</div> <div>Frontális munka</div>																																																		
60 – 62'	<p>4. feladat - Python_07_het_foglalkozas.pdf 2/12. feladata</p> <p>Program, amely kiírja az első N Fibonacci számot. Fibonacci-számok: Az első két elem 0 és 1, a további elemeket az előző kettő összegeként kapjuk.</p> <p>Képletben: Ha <math>F_0 = 0</math> és <math>F_1 = 1</math> akkor a következő elemeket úgy kapjuk, hogy <math>F_n = F_{n-1} + F_{n-2}</math></p> <p>Az első néhány Fibonacci-szám: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34</p>	<div>Feladat kijelölése</div>	<div>Python_07_het_foglalkozas.pdf</div>																																																		
62 – 65'	<table><tr><td>0.</td><td>1.</td><td>2.</td><td>3.</td><td>4.</td><td>5.</td><td>6.</td><td>7.</td><td>8.</td><td>9.</td></tr><tr><td>0</td><td>1</td><td>1</td><td>2</td><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>a</td><td>b</td><td>a + b</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>a</td><td>b</td><td>a + b</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>a</td><td>b</td><td>a + b</td><td></td><td></td><td></td><td></td><td></td></tr></table>	0.	1.	2.	3.	4.	5.	6.	7.	8.	9.	0	1	1	2	3						a	b	a + b									a	b	a + b									a	b	a + b						<div>Algoritmikus gondolkodás</div>	<div>SEGEDanyagok_Gyakorlat_07.xlsx Fibonacci munkalap</div> <div>Szemléltetés</div>
0.	1.	2.	3.	4.	5.	6.	7.	8.	9.																																												
0	1	1	2	3																																																	
a	b	a + b																																																			
	a	b	a + b																																																		
		a	b	a + b																																																	

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK																																												
65 – 67'	<table border="1"> <tr><td>Változók</td><td>a</td><td>b</td><td>i</td></tr> <tr><td>Kezdő értékek</td><td>0</td><td>1</td><td>1</td></tr> <tr><td></td><td>1</td><td>1</td><td>2</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td>2</td><td>3</td><td>4</td></tr> <tr><td></td><td>3</td><td>5</td><td>5</td></tr> <tr><td></td><td>5</td><td>8</td><td>6</td></tr> <tr><td></td><td>8</td><td>13</td><td>7</td></tr> <tr><td></td><td>13</td><td>21</td><td>8</td></tr> <tr><td></td><td>21</td><td>34</td><td>9</td></tr> <tr><td>Értékkadás</td><td>b</td><td>a + b</td><td>i + 1</td></tr> </table>	Változók	a	b	i	Kezdő értékek	0	1	1		1	1	2		1	2	3		2	3	4		3	5	5		5	8	6		8	13	7		13	21	8		21	34	9	Értékkadás	b	a + b	i + 1	<p>Algoritmus felismerése, megértése</p> <p>Algoritmus alkotása</p>	<p>Projektor</p> <p>Megbeszélés</p> <p>SEGEDanyagok_Gyakorlat_07.xlsx Fibonacci munkalap</p>
Változók	a	b	i																																												
Kezdő értékek	0	1	1																																												
	1	1	2																																												
	1	2	3																																												
	2	3	4																																												
	3	5	5																																												
	5	8	6																																												
	8	13	7																																												
	13	21	8																																												
	21	34	9																																												
Értékkadás	b	a + b	i + 1																																												
67 – 69'	<p>Számok cseréje:</p> <pre>a, b = 0, 1 print(a, b) a, b = b, a print(a, b)</pre>	<p>Visszacsatolás</p> <p>Rögzítés</p>	<p>Frontális munka</p>																																												
69 – 72'	<p>Kód írása:</p> <pre>n = int(input("Hány Fibonacci számot szeretne kiíratni? ")) a, b, i = 0, 1, 0 while i &lt; n :     print(a, end=" ")     a, b, i = b, a + b, i + 1</pre>	<p>Alkalmazás</p>	<p>Egyéni munka</p>																																												
72 – 74'	<p>Néhány érdekesség:</p> <p>math modul factorial függvénye</p>	<p>Új ismeret közlése</p>																																													

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<pre>n = int(input("Kérem, adjon meg egy egész számot a faktoriálishoz! ")) print(math.factorial(n))</pre>		<p>Ha marad idő, akkor bemutatom, ha nem, akkor önálló feloldozásra adom ki az alábbiakat.</p> <p>Bemutatás</p>
74 – 76'	<p>Törtek kezelése</p> <p>fractions modul Fraction osztálya</p> <pre>print("Kérem, adjon meg két egész számot, a tört számlálóját és nevezőjét!") a = int(input()) b = int(input()) tort = fractions.Fraction(a, b) print(tort)</pre>	Új ismeret közlése	Bemutatás
76 – 79'	Tanultak összefoglalása	Összefoglalás Rendszerezés	Frontális munka
79 – 80'	Értékelés, házi feladat kiadása	<p>Értékelés</p> <p>A házi feladat kijelölése</p>	<p>Beszélgetés, elbeszélés</p> <p>Frontális munka</p>

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	<p>Tatalék feladat - Python_07_het_foglalkozas.pdf 28. feladata</p> <p>Készítsen programot, amely bekér egész számokat mindaddig, amíg nem adja meg a 0-t! A program határozza meg és írja ki a beadott egész számok közül a legkisebbet és a legnagyobbat! (A 0-t ne számítsa bele a beadott számokba, ez csak a bevitel végét jelzi, ez az ún. végjel!) A számok beolvasását a 0 végjelig ciklus (while True) segítségével valósítsa meg!</p>	<p>Feladat kitűzése</p>	<p>Python_07_het_foglalkozas.pdf</p>
	<p>Feladat megoldása:</p> <pre> min = sys.maxsize # A sys.maxint konstans eltávolításra került, mivel az egész számok értékére # már nincs korlátozás. A sys.maxsize azonban használható egész számként. max = -sys.maxsize while True:     egeszSzam = int(input("Kérem, adjon meg egy egész számot! "))     if egeszSzam &lt; min:         min = egeszSzam     elif egeszSzam &gt; max:         max = egeszSzam     elif egeszSzam == 0:         break </pre>	<p>Új ismeret közlése</p> <p>Algoritmikus gondolkodás</p> <p>Programkód készítése Szintaktika követése</p>	<p>Frontális munka</p> <p>Egyéni munka</p>

## Feladatok

### 0. Alapfeladat:

Olvasson be egész számokat mindaddig, amíg 0-t nem adnak meg!  
Ha 0-t adnak, írja ki, hogy hány darab szám került beolvasásra a 0 kivételével, és vége a feldolgozásnak!

*Más megfogalmazásban:* Olvasson be egész számokat 0 végjelig, majd írja ki a beolvasott értékek számát! A végjel nem eleme a sorozatnak!

1. Írjon programot, ami bekér a felhasználótól egy pozitív számot!  
Ha a felhasználó nem a feltételnek megfelelő számot ad meg, ismétlje meg a bekérést! Ha a szám megfelelő, írja ki a képernyőre a szám 5-ös maradékát!
2. Generáljon le egy véletlen egész számot az  $[1; 12[$  intervallumban, és írja ki a képernyőre a számot, valamint a szám 3-as maradékát!  
Ezek után kérdezze meg a felhasználót, hogy szeretne-e újabb számot! Ha igen, ismétlje meg az előzőt, ha nem, lépjen ki a programból! A felhasználó választását (i/n) egy változóban tárolja el!

3. Rajzoljon le egy  $8 * 8$ -as sakktáblát csillagokból és szóközökből!
4. Program, amely kiírja az első  $N$  Fibonacci számot. Fibonacci-számok: Az első két elem 0 és 1, a további elemeket az előző kettő összegeként kapjuk. Képletben: Ha  $F_0 = 0$  és  $F_1 = 1$  akkor a következő elemeket úgy kapjuk, hogy  $F_n = F_{n-1} + F_{n-2}$

Az első néhány Fibonacci-szám: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

### 5. Tartalék feladat:

Készítsen programot, amely bekér egész számokat mindaddig, amíg nem adja meg a 0-t! A program határozza meg és írja ki a beadott egész számok közül a legkisebbet és a legnagyobbat! (A 0-t ne számítsa bele a beadott számokba, ez csak a bevitel végét jelzi, ez az ún. végjel!) A számok beolvasását a 0 végjelig ciklus (while True) segítségével valósítsa meg.

## Megjegyzés

Eddig digitális munkarendben folyt az oktatás, egyetlen kontakt órámm volt az osztállyal, ha a haladás nem a terv szerint megy, akkor a kimaradt feladatot - a megoldás csatolásával - önálló feldolgozásra adom ki az osztálynak.

Amennyiben látom, hogy a 0. alapfeladat nem okoz nehézséget, akkor az algoritmizálás folyamatát lerövidítem. Pl.: elhagyható a PowerPoint megoldása, elegendő csak a kész algoritmus kivetítése.