

Java programozás

4. óra

Stringműveletek

Matematika

OSZTÁLYOK EGYMÁSRA HATÁSA

- Az osztályok egymásnak üzenetet küldenek, és ezzel valamilyen művelet elvégzésére, vagy tulajdonsága megváltoztatására kényszeríti a küldő a fogadót, azaz:
- Az adott osztály private tulajdonságát egy másik osztályból annak public metódusán keresztül állíthatjuk.

Tankcsata: a harctéren egy Tigris tank rálő egy katonára.

- Tank adatai: típus: Tigris, sebzés: 100 (páncél: 1000).
- Katona adatai: név: Béla, sebzés: 10 (életerő: 100).

Tank metódusa: rálő (tigris.ralo(bela)).

Bela meghívott metódusa: sebzodik(sebzes).

OSZTÁLYOK EGYMÁSRA HATÁSA – MAIN CLASS

- Figyeljük meg, hogy az egyik osztály végrehajt egy műveletet, ami a másik osztályon fejt ki a hatását
→ `ralo(objektum)` metódus

```
public class Tankcsata {  
  
    public static void main(String[] args) {  
        katona bela = new katona("Béla", 100, 200);  
        Tank tigris = new Tank("Tigris", 100, 1000);  
        tigris.ralo(bela);  
    }  
}
```

- Egy szuicid hajlamú tank rá tud lőni saját magára?

OSZTÁLYOK EGYMÁSRA HATÁSA – TANK ÉS KATONA

- Tank „sebzes” értékét vonjuk le Katona „eletero” értékéből

```
public void ralo(katona aldozat) {  
    System.out.println(this.getClass().getSimpleName() +  
        " rálő " + aldozat.getNev() + " katonara");  
    aldozat.sebzodik(this.sebzes);  
}
```

```
new Tank("Tigris", 100, 1000);
```

```
public void sebzodik(int sebzes) {  
    if (this.eletero > sebzes) {  
        this.eletero = this.eletero - sebzes;  
    } else {  
        this.eletero = 0;  
    }  
}
```

16. FELADAT – OSZTÁLYOK EGYMÁSRA HATÁSA

Készítsünk Gyumolcsfa osztályt (név, gyümölcsök száma)

Készítsünk Gyermekek osztályt (becenév, éhes-e)

Mivel a Gyermekek éhes, ezért gyümölcsöt eszik a fáról.

Megehet bármennyi gyümölcsöt a megadott fáról (de nyilván ne többet, mint ahány gyümölcs van rajta).

- Gyermekek metódusa: megeszik(Gyumolcsfa fa, int mennyit);
- Evéskor a fa setGyumolcsokSzama(int gyumolcsokSzama) metódusa kerül meghívásra
- Ha nincs elég gyümölcs, írjuk ki.
- Az elfogyasztott gyümölcsök számot vonjuk le a fától (akár többet is).
- Ha sikerül enni, akkor a Gyermekek már nem éhes.
- Vegyünk fel 1 fát, 2 gyermeket, és felváltva egyenek a fáról.
- Minden evés után írjuk ki az aktuális állapotokat (Fa, Gyerek1, Gyerek2)

A 16. FELADAT EGY LEHETSÉGES KIMENETE:

```
run:
Almafa      : 12 db gyümölcs van
Évike       : éhes
Gerzsonka   : éhes
Évike megevett 5 db gyümölcsöt.
Almafa      : 7 db gyümölcs van
Évike       : jóllakott
BUILD SUCCESSFUL (total time: 1 second)
```

SZÖVEGKEZELÉS ALAPJAI: A STRING

- Karakterek (char) tömbjeként kezelendő
- Objektum → **példányosítható** a new utasítással.
Pl. `String s=new String(„abc”);`
- Szerializálható (állapota elmenthető, byte stream konv.)
- Sorbarendezhető (`Collections.sort(s); Comparable` (később))
- **Hasonlítása mindig equals metódussal** (nem `==` infix)
- Mivel objektum, **vannak metódusai**

STRING: METÓDUSOK

`String s = „alma”; // nem kötelező a „new” hívása!`

`String s2 = new String(„Körte”); // de lehet úgy is`

`System.out.println("Szöveg: " + s);`

`System.out.println("Hossza: " + s.length());`

`System.out.println("2. karaktere: " + s.charAt(2));`

`System.out.println(„Az 'm' betű első előfordulása: " + s.indexOf('m'));`

`System.out.println(„A 'h' betű első előfordulása: " + s.indexOf('h')); // -1`

`System.out.println("egyezik-e 'alma' szóval: " + s.equals("alma"));`

`System.out.println("egyezik-e „Alma” szóval: " + s.equals(„Alma"));`

STRING: METÓDUSOK (FOLYTATÁS)

`System.out.println("üres-e: " + s.isEmpty());` //üres: "" , de **nem a null**

`System.out.println("'ma'-ra végződik-e: " + s.endsWith("ma"));` //startsWith

`System.out.println("Rész kivágása, pl.: " + s.substring(1, 4));`

`System.out.println("Az 'a' cseréje '?' jelre: " + s.replace('a', '?'));`

`System.out.println("Nagybetűs alak lekérése: " + s.toUpperCase());`
// maga a String változik ettől? **Hogyan változik?** – `s = s.toUpperCase`

`System.out.println("Kisbetűs alak lekérése: " + s.toLowerCase());`

`System.out.println("szerepelt-e az \"a\" szövegrész: " + s.contains("a"));`

KITÉRŐ: SHORTIF

1. `System.out.println("szerepelt-e \"a\" szövegrész: " + s.contains("a"));`
// kimenet: true/false
2. `String valasz = "";`
 `if (s.contains("a")) {`
 `valasz = "Igen";`
 `} else {`
 `valasz = „Nem”;`
 `System.out.println(valasz);` // kimenet: Igen/Nem
3. `String valasz = (s.contains("a") ? "Igen" : "Nem");`
 `System.out.println("Szerepelt-e a? " + valasz);` //csak eldöntendőnél!
4. `System.out.println("Szerepelt-e a? " + (s.contains("a") ? "Igen" : "Nem"));`

STRING: METÓDUSOK (FOLYTATÁS)

```
//Elé szóköz, Egyesítve, közé szóköz, utána szóköz, utána egyben");  
String egyben = " " + s + " " + s2 + " egyben";  
System.out.println("Egyben:" + egyben);
```

```
System.out.println("Távolítsuk el a fehér karaktereket!");  
egyben = egyben.trim();  
System.out.println("Egyben: " + egyben);           // Honnét törli csak?
```

```
String[] darabok = egyben.split(" ");              //String-ek tömbje  
System.out.println(Arrays.toString(darabok));
```

```
System.out.println(Arrays.toString("192.168.1.102".split("\\.")));  
Vannak kivételek: . + ? ...
```

STRING: METÓDUSOK - KARAKTER-MŰVELETEK

```
if (Character.isLetter('k')) {  
    System.out.println("Betű!");  
}  
if (Character.isLowerCase('k')) {  
    System.out.println("Kisbetű!");  
}                                     // isUpperCase('k');  
if (Character.isDigit('1')) {  
    System.out.println("Szám!");  
}  
if (Character.isLetterOrDigit('k')) {  
    System.out.println("Betű vagy szám!");  
}  
if (Character.isWhitespace(' ')) {  
    System.out.println("Fehér karakter");  
}
```

GYAKORLÁS: STRING MŰVELETEK

11. Feladat:

Vegyünk fel egy szöveget: „Holnap kedd lesz!”;

a) hány 'a' betű volt?

b) utolsó karakter '?' volt-e

c) cseréljük le az 'e' betűket '+' jelre!

d) írjuk ki egy-egy karakter (pl.: 'p' és 'q') helyét, de ha nincs benne, akkor konzolra -1 helyett: "nincs találat"

11. FELADAT MEGOLDÁSA

```
String s = "Holnap kedd lesz!";
System.out.println(s);

int aDb = 0;
for (int i = 0; i < s.length(); i++) {
    if(s.charAt(i) == 'a'){
        aDb++;
    }
}
System.out.println("a) hány 'a' betű volt? --> " + aDb);

System.out.print("b) utolsó karakter '?' --> ");
if(s.endsWith("?")) {
    System.out.println("igen");
} else {
    System.out.println("nem");
}

s = s.replace('e', '+');
System.out.println("c) cseréljük le az 'e' betűket '+' jelre! --> " + s);

System.out.println("d) írjuk ki egy karakter helyét");
char keresettKarakter = 'd';
System.out.print("\tKeresett karakter: " + keresettKarakter + ", első előfordulása:");
int index = s.indexOf(keresettKarakter);
if(index == -1) {
    System.out.print("\tnincs találat");
} else {
    System.out.print("\t" + (index + 1));
}
System.out.println("");
keresettKarakter = 'q';
System.out.print("\tKeresett karakter: " + keresettKarakter + ", első előfordulása:");
index = s.indexOf(keresettKarakter);
if(index == -1) {
    System.out.print("\tnincs találat");
} else {
    System.out.print("\t" + (index + 1));
}
```

STRING.FORMAT

- Konzolra kiírandó szöveg formázására szolgál

- Általános alakja:

`String.format("-##Δ...##Δ", változók felsorolva)`

ahol: `##`: hány karakterrel töltse fel (jobbra igazítva, ha balra: -)

`Δ`: karakter adattípusa, ami lehet:

szöveg (`%s`), szám (`%d`), helyiértékes szám (`%f`)

(a boolean is szöveg)

- Kiegészíthető egyéb szöveggel a paraméterlistában (lent pl. : °C)

- Példa:

```
System.out.println(String.format("%-24s%2d:%02d  %10.2f°C  
\tÉpp esik:%s", "2020.01.01", 16, 42, 9.2, "Nem"));
```

17. FELADAT

Írassa ki konzolra a saját adatait, előre definiált változókból:

- Név
- Neptun-kód
- Életkor
- Jegyeinek átlaga
- Van-e jogosítványa

- Formázás: 20 karakter balra, 8 karakter balra, 4 karakter jobbra, 10+2 karakter balra, 8 karakter jobbra
- Tegyen bele egy tabulátort is!

17. FELADAT MEGOLDÁS, EGY LEHETSÉGES KIMENET

```
String nev = "Kiss Bence";
String neptun = "AAA12B";
int kor = 22;
double jegyatlag = 4.32;
boolean van = true;
System.out.println(String.format(
    "%-20s%-8s%4d év\tJegyatlag:%-10.2fJogosítvány:%8s",
    nev, neptun, kor, jegyatlag, (van?"van":"nincs")));
```

```
Kiss Bence          AAA12B      22 év      Jegyatlag:4,32      Jogosítvány:      van
BUILD SUCCESSFUL (total time: 0 seconds)
```

ALAPSZINTŰ LISTASZERKEZET: ARRAYLIST

- Azonos típusú elemek listába foglalása
- Testvérei: Vector, LinkedList
- Létrehozás: `ArrayList<Típus> neve = new ArrayList<>();`

ALAPVETŐ MŰVELETEK

- Hozzáadás → `lista_neve.add(elem);`
- Bejárás/ForEach: → `for (Típus egyed: lista_neve) {... }`
- Beszúrás adott helyre → `lista_neve.add(index, elem);`
- Eltávolítás → `lista_neve.remove(index);`
- Lekérdezés → `lista_neve.get(index);`
- Méret lekérdezése: `lista_neve.size();`
- Lista rendezése ABC szerint: `Collections.sort(lista_neve);`

ALAPSZINTŰ LISTASZERKEZET: ARRAYLIST

05. Feladat:

Hozzunk létre egy legalább 3, legfeljebb 10 elemű, pozitív kétjegyű számokból ArrayListet számok néven!

- Töröljük a 3. elemet!
- A 2. helyre szúrjuk be a 88-at!
- Írassuk ki a lista elemeit egymás mellé, TAB-bal elválasztva
- Rendezzük növekvő sorrendbe!
- Írassuk ki az első és az utolsó elemet!
- Ismét írassuk ki a teljes listát TAB-bal elválasztva!

MŰVELETEK ARRAYLIST-TEL

- Véletlen szám létrehozása: `Math.random()`; függvény
- Szintaxisa: `x = (int)(Math.random()*100)); // *100?`
- Lottószámok generálása listába:

```
ArrayList<Integer> lottoSzamok = new ArrayList<>();  
for (int i=0;i<5;i++) {  
    lottoSzamok.add((int) (Math.random() * 100));  
}
```

hol a hiba?

- Átlagszámítás

```
int osszeg = 0;  
for (int i=0;i<lottoSzamok.size();i++) {  
    osszeg+= lottoSzamok.get(i);  
}
```

```
System.out.println("A lottószámok átlaga: " + osszeg/lottoSzamok.size());
```

KITÉRŐ – NETBEANS AUTO INFÓK KIKAPCSOLÁSA

- Ki tudjuk kapcsolni a sok „plusz infót”, amit új osztály létrehozásakor a NetBeans automatikusan generál.
- Tools → Templates → Java → Java class → Open in Editor
(Main-ben is és Class-ban is meg kell csinálni)
- Ami maradjon:

```
<#if package?? && package != "">  
package ${package};  
  
</#if>  
public class ${name} {  
  
}
```

MATEMATIKA A JAVA-BAN – MATH OSZTÁLY

```
System.out.println("Euler-szám: " + Math.E); //term. logaritmus alapja
System.out.println("PI értéke: " + Math.PI);
System.out.println("Gyök 30: " + Math.sqrt(30.0));
System.out.println("Absz.érték -10 számnak: " + Math.abs(-10));
System.out.println("Sin(40): " + Math.sin(40));
System.out.println("Kettő az ötödiken: " + Math.pow(2, 5));
System.out.println("2.6666 kerekítve (round): " + Math.round(2.6666));
System.out.println("2.3333 kerekítve (round): " + Math.round(2.3333));
System.out.println("2.6666 kerekítve (fel): " + Math.ceil(2.6666));
System.out.println("2.3333 kerekítve (fel): " + Math.ceil(2.3333));
System.out.println("-10 és 6 minumuma: " + Math.min(-10, 6));
```

MATEMATIKA A JAVA-BAN - VÉLETLENSZÁM

A metódus egy kvázi-véletlen 0.0 és 1.0 közé eső számmal tér vissza
($0.0 \leq \text{Math.random()} < 1.0$).

Ahhoz, hogy más intervallumban kapjunk meg számokat, műveleteket hajthatunk végre a függvény által visszaadott értéken.

Például véletlenszám 1 és 10 között:

```
int number = (int)(Math.random() * 10) + 1);
```

Egy lottószám kisorsolása (0 nem lehet, ezért kell a +1!):

```
int randSzam= (int)(Math.random()*90)+1;
```

Köszönöm a figyelmet!