

## Objektum orientált programozás

### 5. gyakorlat

#### Osztálydefiníció, referenciák, konstruktor

Az Object osztály toString metódusát minden általunk létrehozott osztály örökli. Ezt felüldefiniálva String-ként összefűzve kiírható egy tetszőleges típusú objektum minden adata (aktuális állapota). Például az Alkalmazott osztály definícióban:

```
@Override
public String toString() {
    return "Name: " + name + " Salary: " + salary;
}
Használata: System.out.println(employee_objektum);
```

1. Készítsen egy *Téglalap* osztályt két adattaggal az oldalak tárolásához (egészek).

- Írjon konstruktort, amely két paraméterként kapott értékkel inicializál.
- Írjon konstruktort, amely, amely egy paraméterként kapott értékkel inicializálja mindkét adattagot (négyzet).
- Írjon metódust, amely visszaadja a területet.
- Írjon metódust, amely egy String-be összefűzve adja vissza a téglalap adatait a következő alakban: "a\_oldal, b\_oldal: terület".
- Írjon setter metódust, amely beállítja a téglalap oldalait a paraméterként kapott két értékkel.
- Írjon setter metódust, amely beállítja a téglalap oldalait a paraméterként kapott egyetlen értéket adva mindkét oldalnak.
- Írjon metódust, amely igazat ad, ha a téglalap nagyobb területű, mint a paraméterként kapott téglalap, hamisat ad ha nem.
- Írjon metódust, amely igazat ad, ha a téglalap oldalai megegyeznek a paraméterként kapott téglalap oldalaival, hamisat ha nem.

Készítsen egy futtatható osztályt, amely a *Téglalap* osztályt használja.

- Hozzon létre két téglalap objektumot ("a" és "b"), és deklaráljon egy harmadik referencia változót ("c"), amely értékül "a"-t veszi fel.
- Írja ki mindhárom referencia változó esetén a String-be összefűzött adatokat.
- Változtassa meg "a" oldalait és írja ki újra mindhárom referencia adatait.
- Változtassa meg "a" oldalait úgy, hogy az megegyezzen "b" oldalaival. Írja ki az (a == b) és az (a == c) kifejezések értékét. Ezután "a" és "b" összehasonlítására használja az utoljára definiált metódust - amely igazat ad, ha a téglalap oldalai megegyeznek a paraméterként kapott téglalap oldalaival, hamisat ha nnowem.

2. Készítsen egy futtatható osztályt, amelyben létrehoz 10 darab *Téglalap* típusú objektumot és eltárolja a referenciájukat egy tömbben, véletlenszerűen generálva az oldalait a 2-10 tartományban.

- Írja ki az összes téglalap adatait String-be összefűzve.
- Írja ki a legkisebb területű téglalap adatait.
- Hozzon létre egy új téglalapot, amelynek az adatait beolvassa. Számolja meg az ettől nagyobb területű téglalapokat.
- Írja ki az első olyan téglalapnak az indexét, amelynek az oldalai megegyeznek a beolvasott téglalapével, vagy ha nincs ilyen, akkor azt hogy "nincs egyező".

3. Készítsen egy *Hanglemez* osztályt, melynek adattagjai az előadó, a cím és a hossz (egész).

- Írjon konstruktort, amely a paraméterként kapott értékekkel inicializálja az adattagokat.
- Írjon metódust, amely egy String-ben összefűzve adja vissza a lemez adatait a következő alakban "Elődó: Cím, hossz perc".

- Írjon metódust, amely 1-t ad vissza, ha a lemez hosszabb, mint a paraméterként kapott lemez, -1-et ad vissza, ha a paraméterként kapott a hosszabb és 0-t ad, ha egyforma hosszúak.
- Írjon metódust, amely paraméterként egy előadót kap (String) és igazat ad, ha a lemeznek ő az előadója. (A String-ek tartalmi egyezőségének vizsgálatára használja a String osztály equals() vagy equalsIgnoreCase() metódusát.)
- Az adattagok értékének lekérdezésére írjon getter metódusokat.

Készítsen egy futtatható osztályt, amelyben létrehoz és tömbben tárol beolvasott számú beolvasott adatú hanglemezt. Írja ki a leghosszabb lemez adatait. Írja ki egy beolvasott nevű előadó lemezeit.

### Házi feladat:

1. Egészítse ki az előző héten definiált Alkalmazott osztályt 2 konstruktorral. Az egyik konstruktornak az alkalmazott nevét és fizetését kell megadni, a másik konstruktornak az alkalmazott nevét, a fizetése 250000 Ft. A második konstruktor használja fel az elsőt! Definiálja felül a toString metódust. Módosítsa az Alkalmazottat használó osztályt ennek megfelelően.
2. **Github classroomban:** Egészítse ki az előző héten definiált Könyv osztályt 2 konstruktorral. Az egyik konstruktornak a könyv minden adatát meg kell adni, a másiknak csak a szerzőt és a címet, az ár 2500 Ft, a megjelenés éve pedig az aktuális év. Definiálja felül a toString metódust. Módosítsa a Könyv osztályt használó osztályt ennek megfelelően.

#### Dátumkezelés (Java 6,7):

```
java.util.Date today = new java.util.Date();//Fri Jun 17 14:54:28 PDT 2016
java.util.Calendar cal = java.util.Calendar.getInstance(); //helyi idő
cal.setTime(today);
int year = cal.get(java.util.Calendar.YEAR);
```

#### Dátumkezelés (Java 8):

```
java.time.LocalDate currentDate = java.time.LocalDate.now(); // mai dátum
int year = currentDate.getYear(); //mai dátumból az év
```