

Objektum orientált programozás

6. gyakorlat

Osztálydefiníció, statikus tagok, final tagok, függvénytúlterhelés, this, csomagok

1. a) Készítsen egy Alkalmazott osztályt saját csomagban, név, kor és fizetés adatokkal.
- Legyen egy osztályszintű adattagja a nyugdíjkorhatár tárolására, értéke kezdetben legyen 65. **Figyelem!** A static adattag kezelése csak static metódusban történhet.
 - Legyen 2 konstruktora. Az egyik a paraméterként kapott név, kor és fizetés értékekkel inicializálja az adattagokat. A másiknak csak a nevet és a kort kell megadni, a fizetés $10000 \cdot \text{kor}$. A második konstruktor használja fel az elsőt.
 - Legyen metódusa, amely visszatéríti hány éve van még nyugdíjig.
 - Legyen metódusa, amely String-be összefűzve adja vissza az Alkalmazott adatait, hozzáfűzve a nyugdíjig hátralevő éveinek számát is (*toString()* felüldefiniálása).
 - Legyen metódusa, amely a paraméterként kapott két Alkalmazott közül azt adja vissza, amelyiknek több éve van még hátra a nyugdíjig.
 - Legyen metódusa, amely a paraméterként kapott értékre állítja be a nyugdíjkorhatárt.

Készítsen egy futtatható osztályt ugyanebben a csomagban, amely beolvasson n számú alkalmazottat egy tömbbe.

- Írja ki az alkalmazottak adatait, majd módosítsa a nyugdíjkorhatárt és újra írja ki az alkalmazottak adatait.
- Írja ki azon alkalmazottak adatait, akiknek 5 évnél kevesebb van még hátra nyugdíjig.
- Írja ki azoknak az alkalmazottaknak az adatait, akiknek az átlagnál több éve van még hátra nyugdíjig.
- Rendezze az alkalmazottak tömbjét a nyugdíjig hátralevő évek alapján növekvő, majd pedig csökkenő sorrendbe.

Minimum kiválasztásos rendezés → növekvően rendezett sorozat

```
private static void sortEmployeeArray(Employee_a[] employeeArray) {
    for(int i=0; i<employeeArray.length; i++) {
        int minIndex = i;
        for(int j=i+1; j<employeeArray.length; j++) {
            if (employeeArray[j].yearsUntilRetirement() <
                employeeArray[minIndex].yearsUntilRetirement()) {
                minIndex = j;
            }
        }
        //Helycsere, ha szükséges
        if (minIndex != i) {
            Employee_a tmp = employeeArray[i];
            employeeArray[i] = employeeArray[minIndex];
            employeeArray[minIndex] = tmp;
        }
    }
}
```

b) Módosítsa az Alkalmazott osztály definícióját úgy, hogy ne az alkalmazott korát, hanem a születési dátumát tároljuk. A korát az aktuális év és a születési év különbségeként számítjuk. Java 8-ban:

```
import java.time.LocalDate;
LocalDate birthday = LocalDate.of(1998,5,10); //year, month, day
LocalDate currentDate = java.time.LocalDate.now();
int currentYear = currentDate.getYear();
```

A nyugdíjkorhatár most legyen konstans (final). Így setter metódusa nem lehet.

Függvényterhelés: a konstruktorból két változatot készítsen. Az egyik változat az alkalmazott nevét és 3 int paramétert kapjon (év, hónap, nap); míg a másik az alkalmazott nevét, egy int, egy sztring és még egy int paraméter felhasználásával adjon értéket a név és a születésnap adattagnak. A fizetést a $10000 \cdot \text{kor}$ képlettel számítsa mindkét konstruktorban.

```
public Employee_b(String name, int year, int month, int day) {
    this.name = name;
    this.birthday = LocalDate.of(year, month, day);
    this.salary = 10000*(LocalDate.now().getYear()-birthday.getYear());
}

public Employee_b(String name, int year, String month, int day) {
    this.name = name;
    int monthvalue;
    if(convertMonthName(month)==0)
        monthvalue = 1;           //default: január
    else
        monthvalue = convertMonthName(month);
    this.birthday = LocalDate.of(year, monthvalue, day);
    this.salary = 10000*(LocalDate.now().getYear()-birthday.getYear());
}
```

A *LocalDate.of()* dátum típust előállító metódus 3 int értéket vár. Ezért a hónap nevét számmá kell konvertálni. A születési dátum kiírásához a másik irányú konverzióra lesz szükség. Írjon olyan, az Alkalmazott objektum adatait kiíró metódust, amely az input paraméter függvényében (String: "age" vagy "birthday") String-be összefűzve az Alkalmazott nevét és korát; ill. az Alkalmazott nevét és születésnapját írja ki. A hónap neve magyarul jelenjen meg (Kiss Tamás, 2008. március 12.). Egy *LocalDate* típusú adattag év, hónap és nap komponenseinek lekérdezése:

```
int year = birthday.getYear();
String month = birthday.getMonth().name();
int monthvalue = birthday.getMonthValue();
int day = birthday.getDayOfMonth();
```

A *getMonth().name()* metódus a hónap angol nevét adja vissza. Írjon konverziós metódust, amely a hónap sorszámának ismeretében visszaadja a magyar hónapnevet. A magyar hónapneveket osztályszintű adattagként, 13 elemű String tömbben tároljuk, ahol a 0. indexű elemet nem használjuk. A tömb elemeit osztályszintű inicializáló blokkban adja meg.

```
private static String[] monthNames;
static {
    monthNames = new String[13];
    monthNames[0] = " ";
    monthNames[1] = "január";
    monthNames[2] = "február";
    monthNames[3] = "március";
    monthNames[4] = "április";
    monthNames[5] = "május";
    monthNames[6] = "június";
    monthNames[7] = "július";
    monthNames[8] = "augusztus";
    monthNames[9] = "szeptember";
    monthNames[10] = "október";
    monthNames[11] = "november";
    monthNames[12] = "december";
}
```

```

public String convertMonthValue(int monthvalue) {
    return monthNames[monthvalue];
}

public int convertMonthName(String monthname) {
    for(int i=1; i<monthNames.length; i++) {
        if (monthNames[i].equalsIgnoreCase(monthname))
            return i;
    }
    return 0;
}

```

Módosítsa a futtatható osztályt ennek megfelelően és próbálja ki az új metódusokat!

2. Készítsen egy Szám osztályt a *myPackage.first* nevű csomagban, amelynek van egy osztályszintű konstans adattagja 0.001 értékkel egy tûrés tárolására. Legyen egy másik `private` adattagja egy valós szám tárolására.

- Legyen konstruktora, amely a paraméterként kapott valós számmal inicializálja az adattagot.
- Legyen egy metódusa, amely igazat ad, ha a valós szám adattagnak a paraméterként kapott számtól való eltérése a tûrésen belül van.
- Legyen getter, setter metódusa a valós szám adattagra.

Készítsen egy futtatható osztályt a *myPackage.second* nevű csomagban, amelyben létrehoz egy Számot, melynek értéke 0-1 közötti intervallumba esik, és eltárolja egy `final` lokális változóban. Generáljon ciklusban 0-1 közötti valós számokat és írja ki azokat egész addig, amíg egyezőt nem talál a létrehozott Számmal (különbségük a tûrésen belül van). Írja ki hányadiknak állította elő ezt a számot.

- Próbálja ki, hogy meg tudja-e változtatni a Szám adattagjának értékét.
- Próbálja ki, hogy meg tudja-e változtatni a referenciájának az értékét.

Házi feladat: Github classroomban

A korábbi gyakorlatokon elkészített Könyv osztályt helyezze át saját csomagba.

- A szerző, cím és ár adattagok változatlanok. Egészítse ki egy oldalszám adattaggal és egy kiadó osztályszintű adattaggal, melynek kezdőértéke "Móra".
- A megjelenés éve adattag legyen konstans, az objektum létrehozás éve.
- Módosítsa a 2 konstruktort ennek megfelelően.
- Módosítsa azt a metódust, amely String-ben összefűzve adja vissza a könyv adatait.
- Írjon osztályszintű metódust, amely két paraméterként kapott könyv közül a hosszabbat adja vissza.
- Írjon metódust, amely igazat ad vissza, ha a könyv oldalszáma páros.

Ebben a csomagban készítsen futtatható osztályt, amelyben beolvas n darab könyvet egy tömbbe.

- Írja ki a leghosszabb könyv adatait.
- Írja ki a leghosszabb páros oldalszámú könyv adatait.

Algoritmus: Létrehozunk egy `maxIndex` változót, melynek értéke -1 . Megkeressük az első páros oldalszámú könyvet. Ennek az `index`-ét eltároljuk a `maxIndex` változóban. Ha a `maxIndex` ≥ 0 (azaz találtunk páros oldalszámú könyvet), akkor a `maxIndex` sorszámtól kezdve maximum kereső eljárás.

- Írja ki, hogy melyik szerzőnek hány darab könyve jelent meg.

Algoritmus: Veszem a következő könyvet. Ha a szerzője megegyezik egy korábban már vizsgált könyv szerzőjével, akkor továbblépek a következő könyvre. Ha ez az adott szerző első előfordulása, akkor létrehozok egy számláló változót 1 értékkel és kiírom a szerző nevét. Az aktuális `index`-től kezdve nézem a könyvek szerzőit, ha azonos szerzőt találok, növelem eggyel a számlálót. A végén kiírom a számláló értékét.

Ugyanezeket a feladatokat oldja meg úgy is, hogy a futtatható osztályt másik csomagban definiálja.