

Objektum orientált programozás

3. gyakorlat (Java)

Véletlenszám generálás, a Math osztály és az Arrays osztály metódusai

1. Írja meg a 4 alapműveletet megvalósító Kalkulátor programot Java nyelven. Az elvégzendő műveletet és az operandusokat a felhasználó adja meg. A program a megadott művelettől függően végzi el a számítási feladatot. A művelet vizsgálatához használja a *switch* utasítást. Az osztás művelet végrehajtásakor ügyeljen az operandusok típusára, és ne engedje a 0-val való osztást. A felhasználói input megadását próbálja meg kétféleképpen:

- Külön olvassa be a két int operandust ellenőrzött módon, és külön az operátort (char). Karakter beolvasása: `char op = (char)input.next().charAt(0);`
- Egyetlen sztringként olvassa be az inputot space-el elválasztva a tagokat (pl.: 3 + 5). A beolvasott sztringet a space-ek mentén szét kell bontani tagokra a *split* metódussal, melynek paramétere a tagoló karaktersorozat. Az eredmény egy sztring tömb, melynek egyes elemeit a megfelelő típusra konvertálva kapjuk meg az operandusokat (a csomagoló osztály *parseXXX* metódusát használva) és az operátort.

Input sztring felbontása:

```
String inString = input.nextLine();
String[] strArray = inString.split(" ");
int a = Integer.parseInt(strArray[0]);
char op = strArray[1].charAt(0);
int b = Integer.parseInt(strArray[2]);
```

Tegye a Kalkulátor programot folyamatos működésűvé. A számítás elvégzése után kérdezze meg a felhasználót, hogy akar-e új műveletet megadni. Olvassa be a választ. Amíg igennel válaszol, addig fusson a program. **Figyelem!** Igen válasz esetén a válasz beolvasása után az input bufferben maradt Enter karaktert ki kell olvasni a *nextLine()* hívással.

2. Deklaráljon, majd foglaljon helyet egy 10 elemű *int* tömb számára. Töltse fel 1 és 50 közötti véletlenszámokkal a tömböt. Használja a *java.lang.Math* osztály *random()* metódusát! Ez a metódus a [0,1) felülről nyitott intervallumból egy lebegőpontos véletlenszámot ad vissza. A számtartományt a C-ből ismert $(felsőhatár-alsóhatár+1)+alsóhatár$ képlettel adhatjuk meg.

```
double random = (Math.random() * 50) + 1;
```

Ha int típusú véletlenszámot szeretnénk: `int random = (int)(Math.random() * 50) + 1;`

A következő műveleteket külön függvényekben valósítsa meg:

- Tömbelemek kiírása *foreach* ciklussal.
- Keresse meg a tömbben a legnagyobb páros számot. Figyelje meg mi a különbség, ha a *for* ciklust, illetve ha a *foreach* ciklust használja!
- Számolja meg, hány négyzetszám van a tömbelemek között.
A négyzetszám vizsgálatához használja a *Math.sqrt()* metódust. Ez a metódus egy *double* típusú argumentumot vár és *double* értéket ad vissza.
- Számítsa ki a tömbelemek mértani átlagát: a tömbelemek szorzata annyiadik gyök alatt, ahány tagú a szorzat. Gyökvonáshoz használja a *Math* osztály *pow()* hatványozó metódusát, amelynek első paramétere a hatványalap (*double*), második paramétere a hatványkitevő (*double*) és *double* értéket ad vissza.

A *java.util.Arrays* osztály metódusaival kényelmesen kezelhetők a tömbök. Például:

- Tömb rendezése (nem ad vissza értéket): `Arrays.sort(myArray);`
- Bináris keresés tömbben (a megtalált elem indexét adja vissza):
`Arrays.binarySearch(myArray, searchKey);`
- Két tömb azonosságának vizsgálata (*boolean*-el tér vissza): `Arrays.equals(a1, a2);`

Próbálja ki a metódusokat *int*, *double* és *char* tömbökön is.

Figyelem! A *double* típusú szám 15 tizedesjegy pontosságú. A 2 tizedesjegy pontosságú tárolásához kerekíteni kell:

```
double value = value*100;  
value = Math.round(value);  
value = value/100;
```

vagy levágni a végét:

```
double value = value*100;  
value = (double)((int) value);  
value = value/100;
```

3. Írja meg a másodfokú egyenlet valós gyökeit meghatározó Java programot. A másodfokú egyenlet konstansait a felhasználó adja meg. Beolvasáshoz használja a *Scanner* osztályt, a számításához pedig a *Math* osztály *sqrt()* metódusát, amely egy *double* típusú argumentumot vár és *double* értéket ad vissza. Az első feladathoz hasonlóan 1) kérje be egyenként a 3 számot, majd 2) egyetlen sztringként, ahol vesszővel választja el a számokat.

Input sztring ellenőrzött beolvasása (sztring → szám konverzió ellenőrzése):

```
int a=0, b=0, c=0;  
boolean ok;  
do {  
    System.out.print("Az egyenlet konstansai vesszővel elválasztva: ");  
    String inString = input.nextLine();  
    String[] strArray = inString.split(",");  
    ok = true;  
    try {  
        a = Integer.parseInt(strArray[0]);  
        b = Integer.parseInt(strArray[1]);  
        c = Integer.parseInt(strArray[2]);  
    }  
    catch (NumberFormatException ex) {  
        ok = false;  
    }  
} while (!ok);
```

Másodfokú egyenlet alakja: $ax^2 + bx + c = 0$

Diszkrimináns számítása: $D = b^2 - 4ac$

Diszkrimináns vizsgálata:

Valós gyök(ök) számítása:

$D = 0 \rightarrow$ Egy valós gyök
 $D > 0 \rightarrow$ Két valós gyök
 $D < 0 \rightarrow$ Két komplex gyök

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Tesztadatok:

$a = 2, b = 4, c = 2 \rightarrow D = 0 \rightarrow$ Egy valós gyök: -1.0

$a = 1, b = 6, c = 5 \rightarrow D = 16 > 0 \rightarrow$ Két valós gyök: -1.0 és -5.0

$a = 2, b = 2, c = 2 \rightarrow D < 0 \rightarrow$ Két komplex gyök

Figyelje meg a *Math* osztály metódusainak használatát! A *Math* osztályban csak static metódusok vannak, azaz az osztály nevével hívhatók. A *Math* osztály a *java.lang* csomagban van definiálva. Ez a csomag alapértelmezett, vagyis importálás nélkül használható. A *Math* osztály metódusainak elérése így: *Math.sqrt()*. Ha az osztálynév megadása nélkül szeretnénk hívni a metódusokat, akkor az **import static Math.*** utasítást kell használni, azaz a *Math* osztály minden static metódusát ismertté tesszük.