Java SE 1
Java alapok JShell kornyezetben

Szabo Daniel
daniel.szabo99@outlook.com

2021. május 29.

Kivonat

Ebben a feladatsorban a Java programozas alapjaival fogsz megismerkedni a Java konzolos kornyezetenek hasznalataval, a JShell-el. A feladatsor fontos alapokat fektet le valtozok kezelesevel, muveletek irasaval es a linearis programozas elso lepeseivel kapcsolatban, amik a legelso programozasi koncepciok amiket megtanulunk, igy a legfontosabbak is. Minden kovetkezo programozasi tananyag erre a tudasra fog epiteni. A feladatsor elvegzesehez szukseged lesz a JShell kornyezetre, amit az `jshell` paranccsal tudsz elohivni, ha a JDK-t korabban feltelepitetted a szamitogepdre. Ha ez nem tortent meg, a Java hivatalos honlapjan megtalalod a telepitesi utmutatot.

Tartalomjegyzék

1. Valtozok

Bevezetes Ebben a feladatban megismerkedunk a programozas egyik legalapvetobb koncepciojaval, a valtozokkal. Megtanuljuk oket letrehozni, modositani, muveletekkel felhasznalni es megjeleniteni oket.

1.1. Adattipusok

A Java nyelvben minden valtozonak kotelezo adattipust meghatarozni. Rengeteg beepitett adattipus van a nyelvben es sajat tipusokat is tudunk letrehozni, de egyelore nehany alapveto tipussal ismerkedunk meg:

Adattipus	**Tarolt adat**	**Alapertek**	**Pelda ertekek**
int	egesz szam	0	-10, 0, 2021
double	tort szam	0.0	-1.25. 0.0, 3.14
boolean	logikai ertek	false	true, false
String	szoveg (karakterlanc)	null	"", "Hello!"

1. táblázat: Alap Java adattipusok

1.2. Valtozo letrehozasa

Egy valtozo letrehozasa jellemzoen ket reszbol all, egy deklaraciobol es egy inicializaciobol. A deklaracio letrehoz egy ures valtozot a megadott nevvel, az inicializacio pedig megadja a valtozo kezdoerteket amit kesobb ha szeretnenk tudunk valtoztatni.

```
1 int i = 12;
```

Kodreszlet 1. Int valtozo deklaralasa inicializacioval. Ennek a valtozonak 12 a kezdoerteke

```
1 double d = 4.5;
```

Kodreszlet 2. Double valtozo deklaralasa inicializacioval. Ennek a valtozonak 4.5 a kezdoerteke

```
1 boolean b = true;
```

Kodreszlet 3. Boolean valtozo deklaralasa inicializacioval. Ennek a valtozonak true a kezdoerteke

```
1 String s = "Hello!";
```

Kodreszlet 4. String valtozo deklaralasa inicializacioval. Ennek a valtozonak "Hello!" lesz a kezdoerteke

1.3. Valtozok modositasa

Egy valtozot korabban letrehoztunk, akkor annak az erteket felulirhatjuk, de a tipusat nem valtoztathatjuk meg. Egy mar korabban letrehozott valtozo hivasakor az adattipust nem kell ujbol meghatarozni. Ha megis ujbol beirjuk az adattipust, Java azt fogja hinni, hogy megint egy valtozot deklaralunk.

```
1 int a = 12;
2 a = 5;
```

Kodreszlet 5. Egy 12 kezdoerteku int valtozot modositunk.

```
1 String s = "Hello!";
2 s = "Hello␣World!";
```

Kodreszlet 6. Egy "Hello" kezdoerteku String valtozot modositunk.

```
1 boolean b = false;
2 b = true;
```

Kodreszlet 7. Egy false kezdoerteku boolean valtozot modositunk.

1.4. Feladatok

1.4.1. Feladat

Hozz letre minden adattipussal 2 valtozot kulonbozo ertekekkel.

1.4.2. Feladat

Mi tortenik, ha

- deklaralsz egy erteket de nem inicializalod,
- olyan erteket adsz egy valtozonak ami mas tipushoz tartozik,
- int valtozoba egy nagyon nagy szamot (nagyobb mint 2147483647) mentesz?

2. Operatorok

Bevezetes Ebben a feladatban nehany alapveto operator (muveleti jel) hasznalatat targyaljuk.

2.1. Alap operatorok

Operator	Adattipus	Muvelet	Hasznalat
=	Barmilyen tipus	Ertekadas	a = 5
+	Szam/String	Osszeadas/Osszefuzes	a = 1 + 2 hello = "Hel"+ "lo"
-	Szam	Kivonas	a = 2 - 1
*	Szam	Szorzas	a = 2 * 3
/	Szam	Osztas	a = 10 / 5
%	Szam	Maradekkepzes	a = 10 % 3 (= 1)
!	Boolean	Negacio	b = !false (= true)
&&	Boolean	Es	false && false (= false) false && true (= false) true && true (= true)
||	Boolean	Vagy	false ||false (= false) false ||true (= true) true ||true (= true)
==	Barmilyen bemenet Boolean kimenet	Egyenloseg	5 == 3 (= false)
!=	Barmilyen bemenet Boolean kimenet	Egyenlotlenseg	5 != 3 (= true)
>	Szam	Nagyobb mint	1 >2
<	Szam	Kisebb mint	1 <2
>=	Szam	Nagyobb vagy egyenlo mint	2 >= 2
<=	Szam	Kisebb vagy egyenlo mint	2 <= 2

2. táblázat: Alap Java operatorok

Fontos! A == egyenloseg vizsgalo operator kizarolag primitiv tipusokkal (pl.: int, double, boolean) mukodik. String-eket es mas nem primitiv tipusokat a reverzibilis `.equals()` fuggvennyel hasonlitunk ossze. A == operator ezeken a tipusokon csak akkor fog igazat adni eredmenyuk, ha egy valtozot onmagaval hasonlitasz ossze.

```
1 String s1 = "abcd";
2 String s2 = "abc";
3 boolean egyenloseg = s1.equals(s2);
4 boolean egyenloseg2 = s2.equals(s1);
```

Kodreszlet 8. String egyenloseg vizsgalata helyesen

2.2. Operatorok hasznalata

```
1 double pi = 3.14;
2 int R = 10;
3 double terulet = pi * R * R;
4 String eredmeny = "A␣kor␣terulete␣" + terulet + ".";
```

Kodreszlet 9. Pelda muvelet integer

```
1  int kor = 22;
2
3  boolean gyermek = kor < 12;
4  boolean serdulo = kor >= 12 && kor < 19;
5  boolean felnott = kor >= 19 && kor < 60;
6  boolean idos = kor >= 60;
7
8  boolean kotelezoSisak = !felnott;
9  boolean kotelezoFelugyelet = !(felnott);
10 boolean ingyenJegy = gyermek || idos;
```

Kodreszlet 10. Pelda muvelet boolean valtozokkal (kalandpark)

2.3. Feladatok

2.3.1. Feladat

Minden operatorral vegezz legalabb 2 muveletet. Talalj ki praktikus, valos problemakat amiket ezekkel az operatorokkal lehet megoldani.

2.3.2. Feladat

Mi tortenik, ha

- String-et osszefuzol egy masik, nem String valtozoval,
- boolean operatort hasznalsz int valtozokkal vagy forditva,
- ket osszeadott int eredmenye nagyobb mint 2147483647),
- int es double valtozokkal vegyesen vegzel muveleteket?

2.3.3. Feladat

A megfelelo operatorok hasznalataval vegezz maradekos osztast.

```
a = 20
b = 4
hanyados = ?
maradek = ?
```

2.3.4. Feladat

Mi lesz a boolean B erteke az alabbi muveletek utan? Eloszor probald meg fejben megoldani, majd ellenorizd magad JShell-ben.

```
1 int a = 4;
2 int b = 3;
3 boolean B = a > b;
```

Kodreszlet 11. Muvelet 1.

```
1 int a = 4;
2 int b = 3;
3 boolean B = a <= b || a != 3;
```

Kodreszlet 12. Muvelet 2.

```
1 int a = 4;
2 int b = 3;
3 boolean c = !(a > b || a < b);
4 boolean d = a % 2 == 0;
5 boolean B = c || d;
```

Kodreszlet 13. Muvelet 3.

3. Kiiras konzolra

A JShell engedelyezi, hogy egy erteket megtekintsunk egyszeruen azzal, hogy beirjuk egy valtozo nevet vagy egy muveletet, de a konzolra kiirasnak kulon parancsa van: `System.out.println("Szoveg")`. Ez egy beepitett fuggveny ami egy `String` tipusu objektumot kap parameterkent es azt megjeleniti. Gyakran ha nem `String` tipusu objektumot kap akkor megprobalja automatikusan String-ge atalakitani.

```
1 String hello = "Hello␣";
2 String world = "World!";
3 System.out.println(hello + world);
```

Kodreszlet 14. Konzolra kiiras 1.

```
1 String nev = "Juliska";
2 int kor = 22;
3 System.out.println("Hello,␣a␣nevem␣" + nev + "␣es␣"
4     + kor + "␣eves␣vagyok.");
```

Kodreszlet 15. Konzolra kiiras 2.

4. If-Else

Az If-Else egy a sok vezerloszerkezet kozul, amit a Java nyelvben hasznalunk. Alapvetoen a program szabalyosan parancsrol parancsra, felulrol lefele ugrik, de vannak helyzetek amikor egy kondiciotol fuggoen szeretnenk valtoztatni, hogy mit csinaljon a programunk. Az If-Else vezerloszerkezet a legegyszerubb ilyen megoldas, amivel a valosagban is gyakran talalkozunk:

Ha [kondicio] akkor [tortenjen valami]
Peldaul:

```
1 int a = 4;
2 if(a > 2) {
3     System.out.println("a␣nagyobb␣mint␣2");
4 }
```

Kodreszlet 16. If hasznalata

Ha [kondicio] akkor [tortenjen valami], kulonben [tortenjen valami mas]
Peldaul:

```
1 int a = 4;
2 if(a > 2) {
3     System.out.println("a␣nagyobb␣mint␣2");
4 } else {
5     System.out.println("a␣nem␣nagyobb␣mint␣2")
6 }
```

Kodreszlet 17. If hasznalata

Ha [kondicio] akkor [tortenjen valami], kulonben ha [masodik kondicio] akkor [tortenjen valami mas], kulonben ha [masodik kondicio] akkor [tortenjen valami mas] , kulonben [tortenjen valami mas].

```
1  int a = 4;
2  if(a > 2) {
3      System.out.println("a␣nagyobb␣mint␣2");
4  } else if (a > 1) {
5      System.out.println("a␣nem␣nagyobb␣mint␣2␣de␣nagyobb␣mint␣1");
6  } else if (a >= 0) {
7      System.out.println("a␣nem␣nagyobb␣mint␣2␣es␣nem␣nagyobb␣mint␣1,
8  ␣␣␣␣␣␣␣␣de␣nagyobb␣vagy␣egyenlo␣0-val");
9  } else {
10     System.out.println("a␣nem␣nagyobb␣mint␣2,␣nem␣nagyobb␣mint␣1
11 ␣␣␣␣␣␣␣␣es␣nem␣nagyobb␣vagy␣egyenlo␣0-val");
12 }
```

Kodreszlet 18. If hasznalata

4.1. Feladatok

4.1.1. Feladat

Mit fognak kiirni a konzolra a kovetkezo kodreszletek? Eloszor probald meg fejben megoldani, majd ellenorizd magad JShell-ben.

```
1 int a = 4;
2 int b = 3;
3 int c = 0;
4 if(a > b) {
5     c = a;
6 } else {
7     c = b;
8 }
9 System.out.println(c);
```

Kodreszlet 19. Muvelet 1.

```
1 int a = 4;
2 int b = 3;
3 if(a == b) {
4     System.out.println(a);
5 } else if (a > b){
6     System.out.println(a);
7 } else {
8     System.out.println(b);
9 }
```

Kodreszlet 20. Muvelet 2.

```
Strin nev = "Julcsika";
int kor = 7;
if(nev.equals("Julcsika") && kor <= 7) {
    System.out.println("Szia␣Julcsika!");
} else if(kor <= 7) {
    System.out.println("Szia!");
} else {
    System.out.println("Udvozlet!");
}
```

Kodreszlet 21. Muvelet 3.

```
int a = 4;
int b = 3;
int c = 5;

if(a < b || a < c) {
    a = b;
    if(a >= c) {
        System.out.println(a);
    } else {
        a = c;
        System.out.println(a);
    }
} else {
    System.out.println(a == b);
}
```

Kodreszlet 22. Muvelet 4.