

Poštovane kolege i kolegice, umesto da vam posebno pišem teorijske odredbe vezane za konkretne primere i gradivo koje obrađujemo, postaviću na platformu skriptu koju sam svojevremeno, dok sam, kao i vi sada, ulazio u svet (Java) programiranja, pisao.

Skripta je pisana na osnovu sadržaja knjige „*Ivor Horton's Beginning: Java 2, JDK 5 Edition*“ i u potpunosti pokriva sadržaje koje ćemo obrađivati u prvom semestru.

Smatram da će vam ista biti od velike koristi i da će kao takva biti odličan dodatni izvor za učenje, a svakako da je kompletnija i sadržajnija od onoga što bi mogli na ovakav način da obradimo, naročito imajući u vidu tempo i raspoloživo vreme.

## VEŽBA 2: Promenljive, Operatori, Konverzija tipova, Kontrolne sekvence

### Primer 1 – Rad sa promenljivama osnovnih tipova podataka

*Zadatak:* Pojam i definisanje promenljivih osnovnih tipova podataka, literala i njihov prikaz.

1. Deklaracija klase i main() metoda:

```
package com.asss.uup;

// PROJEKAT SA PROMENLJIVAMA OSNOVNIH TIPOVA
/*
    Osnovni (prosti, primitivni) tipovi:
    - celobrojni: byte, short, int, double
    - u pokretnom zarezu: float i double
    - karakterni: char
    - logički: boolean
*/

public class PromenljiveOsnovnihTipova {

    public static void main(String[] args) {
```

2. Definisanje promenljivih svih osnovnih tipova podataka (obratiti pažnju na pravila i načine pisanja literala pojedinih tipova, kao i na njihovu međusobnu razliku):

```
// definisanje promenljivih osnovnih (prostih) tipova

// celobrojne vrednosti
byte b = 7;
short s = 9;
int i = 12;
long l = 27L; // ima "L" na kraju literala

// vrednosti u pokretnom zarezu (realni brojevi)
float f = 7.12F; // ima "F" na kraju literala
double d = 12.7;

// karakteri (samo jedan znak)
char c1 = 'n'; // malo slovo "n" u vidu literala
char c2 = 'S'; // veliko slovo "S" u vidu literala
char c3 = '7'; // broj "7" u vidu literala
char c4 = 78; // veliko slovo "N" u vidu decimalnog zapisa
char c5 = '\u0073'; // malo slovo "s" u vidu heksadecimalnog zapisa

System.out.println();

// logičke vrednosti
boolean b1 = true;
boolean b2 = false;
```

3. Narednim parom slika prikazan je kod koji prikazuje vrenosti prethodno definisnih promenljivih i rezultat izvršenja tog koda.

```
// prikaz celobrojnih vrednosti
System.out.println("Celobrojne vrednosti:");
System.out.println(b);
System.out.println(s);
System.out.println(i);
System.out.println(l);

System.out.println();

// prikaz vrednosti u pokretnom zarezu
System.out.println("Vrednosti u pokretnom zarezu:");
System.out.println(f);
System.out.println(d);

System.out.println();

// prikaz karaktera (različito definisanih)
System.out.println("Karakterne vrednosti:");
System.out.println(c1);
System.out.println(c2);
System.out.println(c3);
System.out.println(c4);
System.out.println(c5);

System.out.println();

// prikaz logičkih vrednosti (jedine dve koje postoje)
System.out.println("Logičke vrednosti:");
System.out.println(b1);
System.out.println(b2);
```

```
Celobrojne vrednosti:
7
9
12
27

Vrednosti u pokretnom zarezu:
7.12
12.7

Karakterne vrednosti:
n
s
7
N
s

Logičke vrednosti:
true
false
```

## Primer 2 – Operatori u Javi

*Zatatak:* Prikazati operatore u Javi, njihovu primenu i forme (indirektno rad sa promenljivama)

1. Deklaracija klase, main() metoda i promenljivih za rad:

```
package com.asss.uup;

// PROJEKAT SA OPERACIJAMA
/*
    Vrste operatora:
    - operator dodele
    - aritmetički operatori (ima ih 5, radjene još na prvim vežbama)
    - inkrement/dekrement (uvećavaju/umanjuju vrednost
        promenljive za 1)
    - "op=" operatori (pojednostavljuju upotrebu aritmetičkih)
    - relacioni operatori (upoređuju vrednosti operanada)
    - logički operatori (upoređuju logičke tačnosti operanada)
    - bulovi operatori (slični sa logičkim operatorima)
    - operatori binarnih operacija (nećemo raditi)
    - operacije pomeranja bitova (nećemo raditi)
    - kondicionalni operator (radićemo sa iskazima kontrole toka)
    - operator konkatencije (radićemo sa String objektima)
    - operator provere tipova (radićemo sa objektima)
*/

public class OperacijeNadPodacimaOsnovnihTipova {

    public static void main(String[] args) {

        // definicija/deklaracija promenljivih
        double a = 12, b = 7, c = 0;
        double zbir, razlika, proizvod, količnik, ostatak;
        boolean relOp, logOp;
```

## 2. Rad sa aritmetičkim operatorima:

```
// aritmetički operatori
System.out.println("***** Aritmetički operatori *****");

zbir = a + b;
razlika = a - b;
proizvod = a * b;
količnik = a / b;
ostatak = a % b;

System.out.println();
System.out.println("Za potrebe izračunavanja koristimo sledeće operande a:" + a + " i " + " b:" + b + ".");
System.out.println();

System.out.println("Rezultat operacije sabiranja nad operandima a i b je " + zbir + ".");
System.out.println("Rezultat operacije oduzimanja nad operandima a i b je " + razlika + ".");
System.out.println("Rezultat operacije množenja nad operandima a i b je " + proizvod + ".");
System.out.println("Rezultat operacije deljenja nad operandima a i b je " + količnik + ".");
System.out.println("Rezultat operacije ostatka pri deljenju nad operandima a i b je " + ostatak + ".");

System.out.println();
```

## 3. Rad sa inkrementatorima i dekrementatorima (obratiti pažnju na postojanje dve forma – prefiksna i postfiksna):

```
// inkrement i dekrement (postoje DVE FORME - prefiksna i postfiksna)
System.out.println();
System.out.println("***** Inkrement i dekrement *****");
System.out.println();

System.out.println("Primena inkrementa na zbir (postfiksna forma): " + zbir++);
System.out.println("Novi zbir je: " + zbir);

System.out.println("Primena inkrementa na razliku (prefiksna forma): " + ++razlika);
System.out.println("Nova razlika je: " + razlika);

System.out.println();

System.out.println("Primena dekrementa na proizvod (postfiksna forma): " + proizvod--);
System.out.println("Novi proizvod je: " + proizvod);

System.out.println("Primena dekrementa na ostatak (prefiksna forma): " + --količnik);
System.out.println("Novi ostatak je: " + količnik);

System.out.println();
```

## 4. Rad sa „op=“ operatorima:

```
// "op=" operatori (samo u kombinaciji sa aritmetičim operatorima, ostale nećemo raditi)
System.out.println();
System.out.println("***** 'op=' operatori *****");
System.out.println();

System.out.println("Trenuta vrednost c je : " + c);

c += a;
System.out.println("c += a : " + c);

c -= b;
System.out.println("c -= b : " + c);

c *= a;
System.out.println("c *= a : " + c);

c /= b;
System.out.println("c /= b : " + c);

c %= a;
System.out.println("c %= a : " + c);

System.out.println();
```

## 5. Rad sa relacionim operatorima:

```
// Relacioni operatori
System.out.println();
System.out.println("***** Relacioni operatori *****");
System.out.println();

System.out.println("Vrednosti operanada a i b su: " + a + " i " + b);

relOp = a < b;
System.out.println("Rezultat operacije a < b : " + relOp);
relOp = a > b;
System.out.println("Rezultat operacije a > b : " + relOp);
relOp = a <= b;
System.out.println("Rezultat operacije a <= b : " + relOp);
relOp = a >= b;
System.out.println("Rezultat operacije a >= b : " + relOp);
relOp = a == b;
System.out.println("Rezultat operacije a == b : " + relOp);
relOp = a != b;
System.out.println("Rezultat operacije a != b : " + relOp);

System.out.println();
```

## 6. Rad sa logičkim operatorima:

```
// Logički operatori
System.out.println();
System.out.println("***** Logički operatori *****");
System.out.println();

System.out.println("Trenutne vrednosti izračunatih rezultata su:");
System.out.println("- zbir = " + zbir);
System.out.println("- razlika = " + razlika);
System.out.println("- proizvod = " + proizvod);
System.out.println("- količnik = " + količnik);
System.out.println("- relop = " + relop);

System.out.println();

logOp = zbir > razlika && proizvod > količnik;
System.out.println("Rezultat operacije 'zbir > razlika && proizvod > količnik' je: " + logOp);
logOp = zbir < razlika & proizvod < količnik;
System.out.println("Rezultat operacije 'zbir < razlika & proizvod < količnik' je: " + logOp);
logOp = zbir == razlika || proizvod != količnik;
System.out.println("Rezultat operacije 'zbir == razlika || proizvod != količnik' je: " + logOp);
logOp = zbir >= razlika | proizvod <= količnik;
System.out.println("Rezultat operacije 'zbir >= razlika | proizvod <= količnik' je: " + logOp);
logOp = !relop;
System.out.println("Rezultat operacije '!relop' je: " + logOp);
```

## 7. Izvršenje programa:

```
***** Aritmetički operatori *****

Za potrebe izračunavanja koristimo sledeće operande a:12.0 i b:7.0.

Rezultat operacije sabiranja nad operandima a i b je 19.0.
Rezultat operacije oduzimanja nad operandima a i b je 5.0.
Rezultat operacije množenja nad operandima a i b je 84.0.
Rezultat operacije deljenja nad operandima a i b je 1.7142857142857142.
Rezultat operacije ostatka pri deljenju nad operandima a i b je 5.0.
```



```
***** Inkrement i dekrement *****
```

```
Primena inkrementa na zbir (postfiksna forma): 19.0
```

```
Novi zbir je: 20.0
```

```
Primena inkrementa na razliku (prefiksna forma): 6.0
```

```
Nova razlika je: 6.0
```

```
Primena dekrementa na proizvod (postfiksna forma): 84.0
```

```
Novi proizvod je: 83.0
```

```
Primena dekrementa na ostatak (prefiksna forma): 0.7142857142857142
```

```
Novi ostatak je: 0.7142857142857142
```

```
***** 'op=' operatori *****
```

```
Trenuta vrednost c je : 0.0
```

```
c += a : 12.0
```

```
c -= b : 5.0
```

```
c *= a : 60.0
```

```
c /= b : 8.571428571428571
```

```
c %= a : 8.571428571428571
```

```
***** Relacioni operatori *****
```

```
Vrednosti operanada a i b su: 12.0 i 7.0
```

```
Rezultat operacije a < b : false
```

```
Rezultat operacije a > b : true
```

```
Rezultat operacije a <= b : false
```

```
Rezultat operacije a >= b : true
```

```
Rezultat operacije a == b : false
```

```
Rezultat operacije a != b : true
```

```
***** Logički operatori *****
```

```
Trenutne vrednosti izračunatih rezultata su:
```

```
- zbir = 20.0  
- razlika = 6.0  
- proizvod = 83.0  
- količnik = 0.7142857142857142  
- relOp = true
```

```
Rezultat operacije 'zbir > razlika && proizvod > količnik' je: true
```

```
Rezultat operacije 'zbir < razlika & proizvod < količnik' je: false
```

```
Rezultat operacije 'zbir == razlika || proizvod != količnik' je: true
```

```
Rezultat operacije 'zbir >= razlika | proizvod <= količnik' je: true
```

```
Rezultat operacije '!relOp' je: false
```

### Primer 3 – Rad sa konverzijom tipova i kontrolama sekvenci

Zadatak: Pojam i prikaz automatske i eksplicitne konverzije

Zadatak: Pojam i prikaz osnovnih kontrolnih sekvenci

1. Deklaracija klase i main() metoda:

```
package com.asss.uup;  
  
// PROJEKAT SA KONVERZIJAMA TIPOVA (TYPE CASTING) + KONTROLNE SEKVENCE ("\  
  
public class KonverzijeOsnovnihTipova {  
  
    public static void main(String[] args) {
```

## 2. Automatska konverzija (Slučaj 1)

```
/*
    Automatska konverzija (slučajevi)
    Vršiti se samo u veći tip, nikada suprotno (boolean je izostavljen)
    byte -> short -> int -> long -> float -> double
*/
System.out.println("***** Automatska konverzija *****");

System.out.println();

/*
    Slučaj 1:
    Kod upotrebe operatora dodele desni operand menja tip u tip levog
    (samo u slučajevima kada ne može doći do gubitka podataka)
*/
int a1 = 12, b1 = 7;

/*
    iako je vrednost promenljive upisana kao celobrojna (int)
    prilikom kompajliranja će doći do konverzije tipa u tip double
*/
double zbir1 = 0;
System.out.println("Vrednost promenljive zbir1 je: " + zbir1);

/*
    iako vrednost je operacije sabiranja tip int (operandi su tipa int)
    prilikom kompajliranja će doći do konverzije tipa u tip double
*/
zbir1 = a1 + b1;
System.out.println("Zbir (double) brojeva a1 (int): " + a1
    + " i b1 (int): " + b1 + " je: " + zbir1);

/*
    kod je zakomentisan jer javlja gresku,
    a gresku javlja jer je rezultat operacije sabiranja tip double
    (operandi su tipa double) i kao takvog pokušava da ga smesti u
    promenljivu tipa zbir (int) pri čemu može doći do gubitka podataka
    zbog čega automatska konverzija ne može da se izvrši
*/

// double a2 = 12, b2 = 7;
// int zbir2;
//
// zbir2 = a2 + b2;
```

## 3. Automatska konverzija (Slučaj 2)

```
/*
    Slučaj 2:
    Kod aritmetičkih operacija u kojima se operandi razlikuju po tipu
    doći će do konverzije manjeg operanda u veći radi izvršenja operacije
    (u takvom slučaju nikada ne može doći do gubitka podataka)
*/
System.out.println();

int c = 28;
double d = 27.;
double količnik;

količnik = c / d;
System.out.println("Količnik (double) brojeva c (int): " + c
    + " i d (double): " + d + " je: " + količnik);
```

## 4. Automatska konverzija (Slučaj 3)

```
/*
    Slučaj 3:
    Sve rezultati operacija u kojima su operandi podaci tipa byte i/ili short,
    pošto se iste izvrše, se automatski konvertuju u tip int
    (nema gubitaka podataka jer je int veći tip od tipova byte ili short)
*/
System.out.println();

byte e = 9;
short f = 39;

System.out.print("Za operande e (byte): " + e + " i f (short): " + f);
System.out.println("dobijeni su sledeći zbrojevi:");

int zbir3 = e + e; // automatska konverzija u int
long zbir4 = e + f; // automatska konverzija u short pa u int pa u long
float zbir5 = f + e; // automatska konverzija u short pa u int pa u float
double zbir6 = f + f; // automatska konverzija u int pa u double

System.out.println("(int) = " + zbir3 + "; (long) = " + zbir4 +
    "; (float) = " + zbir5 + "; (double) = " + zbir6);
```

## 5. Automatska konverzija (Slučaj 4)

```
/*  
  Slučaj 4:  
  Prilikom rada sa "op=" operatorima uvek dolazi do automatske konverzije  
  tipa njihovog desnog operanda u tip njihovog levog operanda.  
  (u ovom slučaju MOŽE DOĆI DO GUBITKA PODATAKA i to bez ikakvog upozorenja)  
*/  
System.out.println();  
  
double g = 12.7;  
int h = 9;  
double i = 27.;  
  
System.out.println("Vrednosti promenljivih su:");  
System.out.println("g (double): " + g);  
System.out.println("h (int): " + h);  
System.out.println("i (double): " + i);  
  
i += g;  
h += g;  
  
System.out.println("Nakon 'op=' operacija 'i += g' i 'h += g' vrednosti"  
    + " i i h su i : " + i + " i " + " h: " + h);
```

## 6. Експлицитна конверзија (Случај 1)

```
System.out.println();           System.out.println();
/*
    Експлицитна конверзија
    Врши се у случајевима када је потребно тип променљиве већег типа
    променити у мањи, када логика задатка то налаже и може се писати за
    сваки од случајева који су обухваћени аутоматском конверзијом
    (УВЕК МОЖЕ ДОЋИ ДО ГУБИТКА ПОДАТАКА БЕЗ ИКАКАВИХ ОБАВЕШТЕЊА)
*/
System.out.println("***** Експлицитна конверзија *****");

System.out.println();
/*
    Случај 1:
    Промена променљиве из већег у мањи тип
    (закomentarисани случај аутоматске конверзије (линије 48-51))
*/
double a2 = 12.7, b2 = 7.9;
int zbir2;

System.out.println("За израчунавање се користе операнди " +
    "a2 (double): " + a2 + " и b2 (double): " + b2);

zbir2 = (int) (a2 + b2);    // резултат је 20,7 -> (int) -> 20

System.out.println("Пошто се резултат њихове операције (експлицитно) " +
    "конвертује у тип int, добија се збир (int): " + zbir2);
```

## 7. Експлицитна конверзија (Случај 2)

```
/*
    Slučaj 2:
    Primena eksplicitne konverzije radi zadovoljenja logike
    (u konkretnom slučaju se konvertuje u viši tip pa se podaci ne gube)
*/
System.out.println();

System.out.println("Operandi u operaciji deljenja su a1 (int): " + a1 +
    " i b1 (int): " + b1 + ", dok je promenljiva količnik tipa double.");

količnik = a1 / b1;

System.out.println("Količnik izračunat bez upotrebe konverzije u formuli" +
    " 'količnik = a1 / b1' ima vrednost: " + količnik);

/*
    da bi se dobio očekivni rezultat, tj. da ne bi došlo do
    gubitka podataka konverziju treba primeniti na jednu od
    promenljivih pri čemu će kompajler automatski konvertovati
    i drugu, a time će i rezultat biti novog tipa (double)
*/

double količnik1 = (double) a1 / b1;
double količnik2 = a1 / (double) b1;
// double količnik3 = (double) (a1 / b1);

System.out.println("Međutim, ukoliko upotrebimo eksplicitnu konverziju nad " +
    "nekim od operandi dobija se količnik: " + količnik1 + " ili " + količnik2);
```

## 8. Експлицитна конверзија (Slučaj 3)

```

/*
    Slučaj 3:
    Primena eksplicitne konverzije u slučaju koji bi svakako
    bio obuhvaćen automatskom konverzijom (Slučaj 3, linija 86)
*/
System.out.println();

System.out.print("Za operande e (byte): " + e + " i f (short): " + f);
/*
    Ovim redosledom kompajler vrši automatsku konverziju izraza 'zbir4 = e + f':
    1)      e          -> short
    2)      rezultat e + f  -> int (uvek se desava sa byte i short tipovima)
    3)      int oblike rezultata -> long
*/
zbir4 = (long)((int)((short)e + f));

System.out.println(", nakon izvršavanja (eksplicitnih) konverzija, " +
    "zbir je: " + zbir4);

```

## 9. Контролне секвенце

```

System.out.println();          System.out.println();
/*
    Kontrolne sekvence - "\"
    Koriste se u String objektima (znakovni nizovi, tekst) radi uporebe
    određenih karaktera i kontrole ispisa.
*/
System.out.println("***** Kontrolne sekvence - \"\\\" *****");

System.out.println();

System.out.println("Slovo S ima heksadecimalnu vrednost 0053: " + '\u0053');
System.out.println();
System.out.println("U dva reda se može pisati i ovako,\nbez novog println() metoda.");
System.out.println();
System.out.println("Kada je potrebno poravnati izlaze:");
System.out.println("a2 =\t\t" + a2);
System.out.println("(int) a2 =\t" + (int)a2);
System.out.println();
System.out.println("\"Kontrolna sekvenca\" dobro dodje kada nam trebaju posebni znaci.");

```



## 10. Извршење програма:

```
***** Automatska konverzija *****
```

```
Vrednost promenljive zbir1 je: 0.0
```

```
Zbir (double) brojeva a1 (int): 12 i b1 (int): 7 je: 19.0
```

```
Količnik (double) brojeva c (int): 28 i d (double): 27.0 je: 1.037037037037037
```

```
Za operande e (byte): 9 i f (short): 39 dobijeni su sledeci zbrovi:  
(int) = 18; (long) = 48; (float) = 48.0; (double) = 78.0
```

```
Vrednosti promenljivih su:
```

```
g (double): 12.7
```

```
h (int): 9
```

```
i (double): 27.0
```

```
Nakon 'op=' operacija 'i += g' i 'h += g' vrednosti i i h su i : 39.7 i h: 21
```

```
***** Eksplicitna konverzija *****
```

```
Za izračunavanje se koriste operandi a2 (double): 12.7 i b2 (double): 7.9
```

```
Pošto se rezultat njihove operacije (eksplicitno) konvertuje u tip int, dobija se zbir (int): 20
```

```
Operandi u operaciji deljenja su a1 (int): 12 i b1 (int): 7, dok je promenljiva količnik tipa double.
```

```
Količnik izračunat bez upotrebe konverzije u formuli 'količnik = a1 / b1' ima vrednost: 1.0
```

```
Medjutim, ukoliko upotrebimo eksplicitnu konverziju nad nekim od operandima dobija se količnik: 1.7142857142857142 ili 1.7142857142857142
```

```
Za operande e (byte): 9 i f (short): 39, nakon izvršavanja (eksplicitnih) konverzija, zbir je: 48
```

```
***** Kontrolne sekvence - "\" *****
```

```
Slovo S ima hexadecimalnu vrednost 0053: S
```

```
U dva reda se može pisati i ovako,  
bez novog println() metoda.
```

```
Kada je potrebno poravnati izlaze:
```

```
a2 =          12.7
```

```
(int) a2 = 12
```

```
"Kontrolna sekvenca" dobro dodje kada nam trebaju posebni znaci.
```