

## Uvod u Javu

**NAPOMENA:** Vežbe podrazumevaju da je student ovladao teorijom iz dela “Uvod u programski jezik Java” (11\_uvod-java.ppt)

## Java

Java instalacija dolazi u dva oblika: JDK (Java Development Kit) i JRE (Java Runtime Environment).

Provera da li je postoji java na računaru. Otvorite konzolu i ukucajte: **java -version**

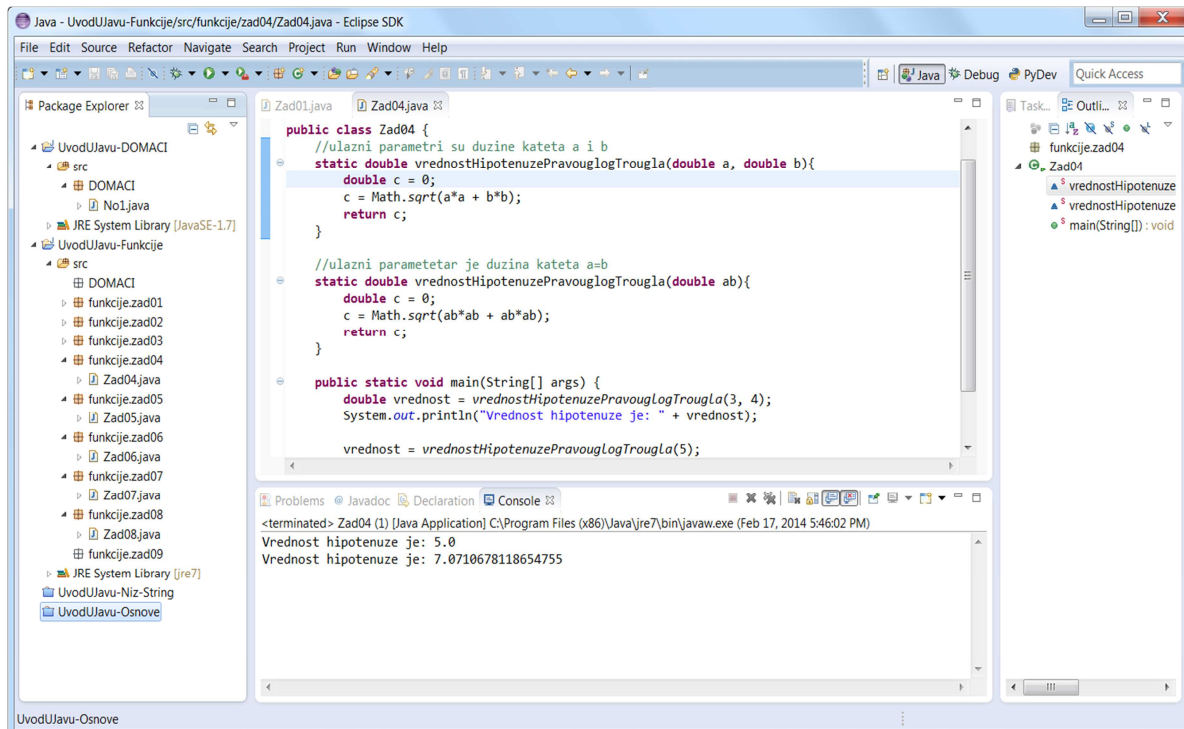
Provera da li je postoji java kompajler na računaru. Otvorite konzolu i ukucajte: **javac -version**

Kompajliranje java klasa iz konzole postiže se komandom **javac**. Kao parametri poziva pomenutoj komandi se prosleđuje ime java klase, pri čemu će rezultat biti odgovarajući **.class** fajl. Npr. izvršavanjem komande **javac Test.java** nastaje fajl **Test.class**. Pokretanje java programa postiže se upotrebom komande **java Test.class**. Za kompajliranje java fajlova i njihov poziv na vežbama koristimo alat Eclipse.

## Eclipse

Eclipse predstavlja open-source razvojnu platformu i skup aplikativnih okruženja za izgradnju softvera (nije ograničen samo na java aplikacije).

Pokretanjem okruženja otvara se radni prostor (workbench). Radni prostor je moguće organizovati na više načina, odnosno, perspektiva (perspective) koje uključuju određeni skup alata i specijalizovane su za različite namene (desni gornji ugao eclipse alata; perspektive: Java - razvoj standardnih java aplikacija, Java EE - prošireno standardno okruženje koje se npr. može koristiti za razvoj web aplikacija, PyDev - razvoj python aplikacija, Debug - okruženje za debugovanje/traganje grešaka).



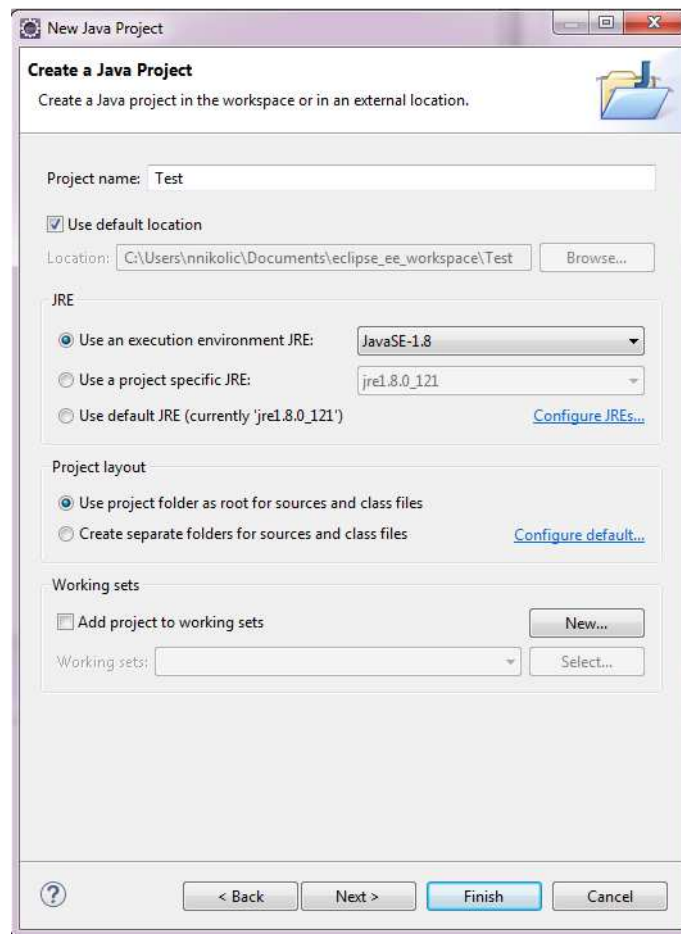
Radni prostor Java perspektive sastoji se od sekcija: **PackageExplorer** (prikazuje listu projekata koji se nalaze u radnom prostoru i hijerarhiju Java resursa u okviru projekata), **Outline** (prikazuje listu atributa i metoda klase koja se edituje), **Problems** (prikazuje tabelu grešaka i upozorenja vezanih za izvorni kod), **Console** (prikaz porukana ekran započet java program) i **Editor** (zauzima centralni deo površine radnog prostora).

## Rad sa projektima

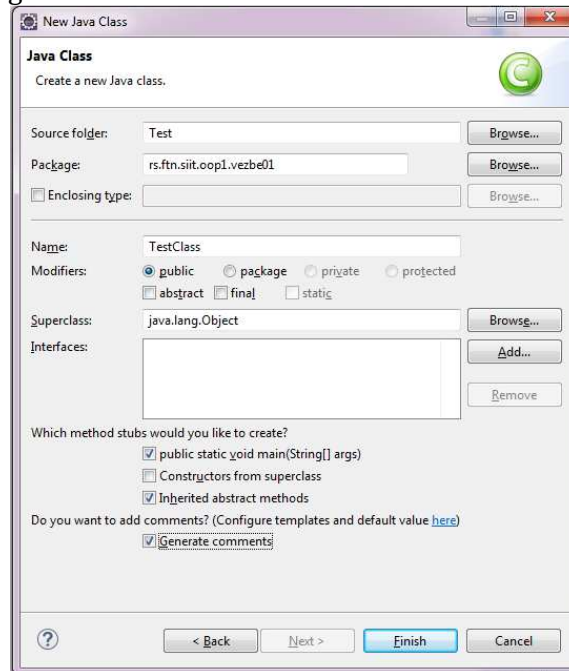
Programski kod se organizuje unutar projekata. Projekat se sastoji od hijerarhije klasa (u skladu sa pravilima definisanja paketa u Java programskom jeziku) i drugih datoteka (npr. slika, txt fajlova, konfiguracionih datoteka itd.). Radni prostor se mapira na odgovarajući direktorijum lokalnog sistema datoteka. U okviru radnog prostora može da bude definisano više projekata. Projekat može da bude otvoren (ikonica otvorena fascikla) ili zatvoren (ikonica plava zatvorena fascikla). Samo otvoreni projekti su aktivni, tj. nad njima je jedino dozvoljen rad u eklipsi. Zatvoreni projekti ne mogu da se menjaju jer nisu aktivni (ali se i dalje nalaze u lokalnom sistemu datoteka). Poželjno je zatvarati nepotrebne projekte jer se na taj način oslobađaju zauzeti memorijski resursi i podižu performanse okruženja.

## Kreiranje prvog projekta

Kreirati svoj prvi java projekat pokretanjem vizarda *File→New→Java Project*. Dodeliti vrednost Test kao ime projekta (pogledati sliku). Potom kliknuti na dugme *Next* pa na *Finish*.



Kreirati svoju prvu Java klasu pokretanjem vizarda *File→New→Class*. Dodeliti vrednost *Test* kao ime klase (pogledati sliku), vrednost *vezbe1* kao ime paketa i omogućiti stavku *public static void main*. Potom kliknuti na dugme *Finish*.



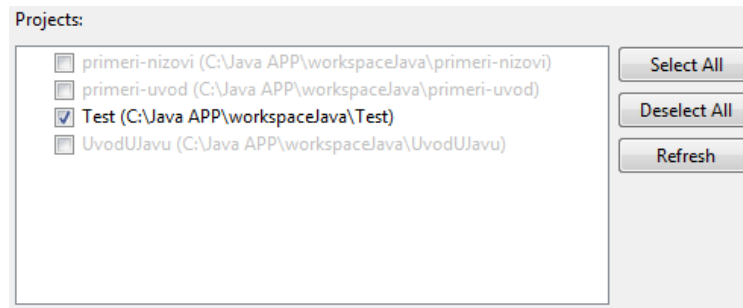
## Rad sa paketima u Javi

Paketi predstavljaju hijerarhijsku organizaciju fajlova po folderima i podfolderima. Paketi se kreiraju pokretanjem vizarda *File→New→Package* i zadavanjem njegovog imena. Kreirati nekoliko novih paketa pod nazivom *vezbe1.podpaket*, *vezbe2*, *vezbe3.podpaket*. Kroz operativni sistem otići na lokaciju projekta i istražiti sadržaj foldera *src*.

**Zatvaranje otvorenog projekta** - Desni klik na projekat pa stavka **Close Project**. Projekat je zatvoren ali ne i uklonjen iz okruženja.

**Uklanjanje projekta iz okruženja** - Projekte je moguće ukloniti logički i fizički iz radnog prostora. Uklanjanje se postiže desnim klikom na selektovani projekat pa stavka **Delete**.

**Otvaranje postojećeg projekta** - Prebaciti folder postojećeg projekta u radni direktorijum *WorkspaceJava*. Pozvati opciju *File→Import→General→Existing project into Workspace* i kliknuti na dugme *Next*. Na dugmetu *Browse* postaviti putanju do radnog direktorijuma *WorkspaceJava*. Iz liste ponuđenih projekata odabrati onaj koji se importuje. Klik na dugme *Finish*.



Pri otvaranju postojećih projekata može doći do greške ukoliko su podešavanja Jave različita na računaru na kome otvarate projekat od računara sa koga ste projekat preuzeli (neki od problema: vrezije java 6 ili 7, kompatibilnosti kompajlera...). Tada je potrebno prepodesiti podešavanja Jave za uvezeni projekat. Prethodno se postiže desnim klikom na projekat pa stavka **Properties**. **Podešavanje putanje do java biblioteke** postiže se odlaskom na *Java Build Path* → *Libraries*, uklanjanjem (dugme Remove) nepostojeće putanje (crveno obeleženo) i dodavanjem ispravne klikom na dugme *Add Library* → *JRE System Library*. **Podešavanje kompatibilnosti kompajlera** postiže se odlaskom na *Java Compiler* → *JDK Compliance*.

## Pravilno pisanje programskog koda

Glavni cilj prilikom pisanja programskog koga jeste da bude lak za čitanje i razumevanje. Dobro napisan programski kod je lakše održavati i ima manje greška. Zato je potrebno pridržavati se istih principa pisanja programskog koda. Na taj način će svakom inženjeru biti lakše da razume i održava programski kod koji je pisala neka druga osoba.

Opštia pravila pisanja koda su:

- Pišite kraće programe i metode kojim se može lakše upravljati.
- Koristite prednosti programskog jezika
- Formatirajte programski kod (auto-format u Eclipse: *Ctrl+Shift+F*)
- Koristite komentare

Postoje razne konvencije davanja naziva indentifikatora koje definišu pravila pisanja varijabli, tipova, funkcija i ostalih entiteta u programskom kodu i dokumentaciji. Prema sugestijama Java zajednice, prilikom pisanja Java koda se najčešće koristi *CamelCase* notacija. Nazivi indentifikatora prema *CamelCase* notaciji su sastavljeni od brojeva i slova bez razmaka, gde je svako pocetno slovo reci veliko (npr. *CamelCase*). Opšta pravila davanja naziva indentifikatora, prema *CamelCase* notaciji, su data u nastavku.

Identifikatori	Pravila	Primeri
Promenljive	Kraći smisleni nazivi koji čitaocu koda nagoveštavaju namenu varijable. Počinju malim slovom i svako prvo slovo naredne reči (ako postoji) je veliko.	width studentName numberOfStudents
Konstante	Koristite velika slova a reči razdvajati donjom crtom	MAX_HEIGHT DAYS_PER_WEEK
Klase	Nazivi bi trebalo da budu imenice gde je prvo slovo svake reči veliko.	Student ServiceGroup
Metode	Nazivi bi trebalo da budu glagoli koji opisuju šta metoda radi. Počinju malim slovom i svako prvo slovo naredne reči (ako postoji) je veliko.	create() run() findById()

Generalni principi davanja naziva identifikatora su:

- Koristite smislene nazive koja odgovaraju nameni odgovarajuće varijable. Takođe izbegavati tajnovite skraćenice. Npr. Umesto *pt*, *ptr* ili *povt* koristiti *povrsinaTrougla* za vrednost površine trougla.
- Nazivi *boolean* promenljivih i metoda trebaju biti nedvosmisleni. Npr. *isEmpty()*, *contains()*, *isDeleted* itd.
- Koristite kratke nazive za varijable kratkog zivotnog veka i brojače unutar petlji (npr. *i*, *j*, *k* itd.). Za varijable od većeg značaja koristite smislenije nazive.
- Izbegavati generička i beznačajna imena poput *foo*, *tmp*, *bla*, *asd*, *wow*, *lol* i itd. Koristiti terminologiju domena aplikacije za koju se pise programski kod.

## Osnove

### Primer 1. Upotreba programskih komentara

```
//Komentar naveden u okviru jednog reda

/*
Komentar
naveden u okviru
vise redova
*/
```

### Primer 2. Ispis na ekran.

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

    // opcija 1
    System.out.print("A");
    System.out.print("B\n");
    System.out.println("C");

    // opcija 2
    System.out.println("A");
    System.out.println("\tB");

    // koriscenje operatora + za konkatenciju teksta i brojeva
    System.out.println("Broj dana u nedelji je: " + 7);

}
```

**Primer 3. Korišćenje operatora dodele (=), deklaracija promenljivih i tipovi promenljivih**

```

public static void main(String[] args) {

    // deklaracija i inicijalizacija promenljivih
    // celobrojni tip, integer
    int a = 1;
    System.out.println("Vrednost a " + a );

    //deklaracija i inicijalizacija vise promenljivih
    int promenljiva1, promenljiva2 = 3, promenljiva3 = -1, rezultat;

    //greska - dodela bez prethodne inicijalizacije
    rezultat = promenljiva1;

    // decimalni tip, float, double
    float b= 5.55F;
    double d = 3.14;

}

```

**Primer 4. Korišćenje aritmetičkih operatora (+,-,\*,/,%)**

```

//koriscenje aritmetickih operatora
rezultat = promenljiva2 + a * promenljiva3;
System.out.println("Rezultat je: " + rezultat);

rezultat = 7 % 4;
System.out.println("Ostatak pri deljenju 7 sa 4 je: " + rezultat);

System.out.println("Vrednost b + c " + b + c );
System.out.println("Vrednost b + c sa zagradama " + (b + c) );
System.out.println("Vrednost b + d sa zagradama " + (b + d) );

```

**Primer 5. Prefiksni i sufiksni oblik korišćenja operatora uvećanja i smanjenja. Uočiti razlike.**

```

int a = 0;
// postfiksni oblik
System.out.println("3. Ispis promenljive a je: "+ a++);
System.out.println("4. Ispis promenljive a je: "+ a);

// prefiksni oblik
System.out.println("1. Ispis promenljive a je: "+ ++a);
System.out.println("2. Ispis promenljive a je: "+ a);

```

**Primer 6.** Kombinovanje aritmetičkih operatora i operatora dodele (+, -, \*, /, %, itd.), eksplicitna i implicitna konverzija, konstante.

```
int a = 1;
//deklaracija konstante b, rezervisana rec final
final int B = 5;
//greska prilikom dodele vrednosti konstanti
B = 4;

//koriscenjem kombinovanog operatora += i -=
a += 2; //isto kao i a=a+2
System.out.println("Vrednost uvecane promenljive "
    + "a=1 za 2 je:" + a);

a -= B; //isto kao i a=a-b
System.out.println("Vrednost umanjene promenljive "
    + "a=3 za konstantnu b=5 je:" + a);

// eksplicitna konverzija
float c = 1.3F;
// gubitak ostaka promenljive c
a = a + (int) c;
System.out.println("Uvecanje promenljive "
    + "a=-2 za c=1.3 je:" + a);

//implicitna konverzija
c = c + a;
```

**Primer 7.** Klasa Math, rad sa matematičkim formulama i konstantama.

```
//Klasa Math
/*
    oredjivanje površine kruga poluprečnika r=2,
    usvojeno je da je vrednost Pi preuzeta iz klase Math
*/
double površina;
int r = 2;
//opcija 1
površina = r * r * Math.PI;
System.out.println("Površina kruga je: " + površina);
//opcija 2
//koriscenje funkcije pow(x,y) za odredjivanje stepena r^2
površina = Math.pow(r, 2) * Math.PI;
System.out.println("Površina kruga je: " + površina);
```

**Primer 8.** Logički i relacioni operatori

Logički Operatori: operator "||" znaci "ili" disjunkcija, dok operator "&&" znaci "i" konjukcija.

Relacioni Operatori: "<" znači manje od, ">" znači veće od, "<=" znači manje ili jednako od, ">=" znači veće ili jednako od, "instanceof" znači promenljiva je tipa

Operatori jednakosti: operator "!=" znači razlicito od, a "==" jednako je sa.

```
// definicija promenljivih
boolean a = true, b = false, c = true;
int d = 2, e = 5;

//koriscenje logickih operator && i ||
System.out.println("Vrednost konjukcije a i b je: " + (a && b));
System.out.println("Vrednost konjukcije a i c je: " + (a && c));
System.out.println("Vrednost disjunkcije a ili b je: " + (a || b));
System.out.println("Vrednost slozenog izraza (a i b) ili c je: " + ((a && b) || c));

//koriscenje relacionih operatora >, <=, ==, !=
System.out.println("Vrednost relacije d vece od e je: " + (d > e));
System.out.println("Vrednost relacije d manje i jednako od e je: " + (d <= e));
System.out.println("Vrednost relacije d jednako e je: " + (d == e));
System.out.println("Vrednost relacije d razlicito e je: " + (d != e));

//kombinacija relacionih i logickih operatora u izrazu
boolean rezultat;
rezultat = (d > e) && a; // false konjukcija true
System.out.println("Vrednost rezultata konjukcije relacije "
    + "(d vece od e) i promenljive a je: " + rezultat);
```

**Primer 9. Kontrola toka programa if, if-else, if-elseif-else**

## a. IF naredba

```
/* if naredba
if ( logicki izraz) {
    naredbe koje se izvorsavaju ako i samo ako
    logicki izraz rezultuje sa true
}
*/
// proverava da li je godina tekuca
int godina = 2014;
if (godina==2014) {
    System.out.println("Godina je tekuca");
}
```

## b. IF-ELSE naredba



```
/* if-else naredba
if ( logicki izraz) {
    naredbe koje se izvršavaju ako i samo ako
    logicki izraz rezultuje sa true
}
else {
    naredbe koje se izvršavaju u svim ostalim slučevima
    tj. logicki izraz rezultuje sa false
}
*/
// proverava da li je a jednocifren ili dvocifren broj
int a = 3;
if (a>-10 && a<10) {
    System.out.println("Broj je jednocifren");
}
else {
    System.out.println("Broj je dvocifren");
}
```

## c. IF-ELSEIF-ELSE naredba

```
/* if-elseif-else naredba
if ( logicki izraz1) {
    naredbe koje se izvršavaju ako i samo ako
    logicki izraz 1 rezultuje sa true
}
else if ( logicki izraz2) {
    naredbe koje se izvršavaju ako i samo ako
    logicki izraz 2 rezultuje sa true
    N puta može da se ponovi else if deo
}
else {
    naredbe koje se izvršavaju u svim ostalim slučevima
    tj. logicki izraz 1 i 2 su false
}
*/
// proverava kretanje kursa eura - pada, raste ili stagnira,
// ako se pretpostavlja da ispod 110 pada,
// između 110 i 115 stagnira, i preko 115 raste
double kurs = 108.9;
if (kurs<110) {
    System.out.println("Euro pada");
}
else if (kurs>=110 && kurs <=115 ) {
    System.out.println("Euro stagnira");
}
else {
    System.out.println("Euro raste");
}
```

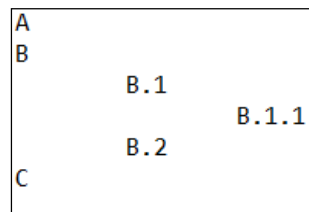
**Primer 10. Kontrola toka programa switch.** Switch kontrola toka radi samo sa primitivnim tipovima podataka byte, short, char i int. Takođe switch može da radi i sa enumeracijom. Od Java verzije 1.7 switch može da se koristi i String.

```
int mesec = 1;
switch (mesec) {
    case 1: System.out.println("Januar");
            break;
    case 2: System.out.println("Februar");
            break;
    case 3: System.out.println("Mart");
            // break;
    case 4: System.out.println("April");
            // break;
    case 5: System.out.println("Maj");
            // break;
    case 6: System.out.println("Juni");
            break;
    case 7: case 8: case 9: System.out.println("Letnji meseci");
            break;
    case 10: System.out.println("Oktobar");
            break;
    case 11: System.out.println("November");
            break;
    case 12: System.out.println("Decembar");
            break;
    default: System.out.println("Nepostojeci mesec");
            break;
}
```

## Zadaci



**Zadatak 1.** Napisati program koji omogućuje ispis na konzoli kao na slici.



**Zadatak 2.** Napisati program za računanje osobina geometrijskih figura. Deklaristati promenljive koja predstavljaju: površinu kvadrata 16, osnovicu jednakokrakog trougla dužine 4 i krak dužine 6. Izračunajte i ispišite (na konzoli) vrednost promenljive stranice kvadrata i površine jednakokrakog trougla, ako se zna da je:

$$h_a = \sqrt{b^2 - \frac{a^2}{4}} \quad P = \frac{a \cdot h_a}{2} = \frac{b \cdot h_b}{2}$$

- The year can be evenly divided by 4, is a leap year, unless:
- The year can be evenly divided by 100, it is NOT a leap year, unless:
- The year is also evenly divisible by 400. Then it is a leap year.

- ✓ **Zadatak 3.** Napisati program koji za danu godinu između (broj između 1538 i 10000) i utvrđuje da li je ona prestupna. Po gregorijanskom kalendaru prestupne godine određuju se na sledeći način:
- ako je godina deljiva sa 400, prestupna je (npr.2000 godina je prestupna)
  - ako godina nije deljiva sa 400, ali je deljiva sa 100, nije prestupna (npr.1900. godina nije prestupna)
  - ako godina nije deljiva sa 100, ali je deljiva sa 4, prestupna je (npr.2004 godina je prestupna)
  - ako godina nije deljiva sa 100 i nije deljiva sa 4, nije prestupna (npr.2001. nije prestupna)

- ✓ **Zadatak 4.** Napisati program koji za dato rastojanje u centimetrima (npr. 324) određuje koliko ima metara i decimetara?

- ✓ **Zadatak 5.** Napisati program koji za date stranice a=3, b=4 i c=5 računa površinu i zapreminu kvadra.

- ✓ **Zadatak 6.** Kolika je površina kupe prečnika 6 i visine 4? Ako nam je poznata formula, gde je s dužina izvodnice.

$$S = S_o + S_b = r s \pi + r^2 \pi = r \pi (s + r)$$

$$A = \pi r (r + \sqrt{h^2 + r^2})$$

- ✓ **Zadatak 7.** Napisati program koji na osnovu vrednosti x i y računa i prikazuje vrednost izlaznog podatka z. Izlazni podatak z se računa po sledećoj formuli:

$$z = \begin{cases} \frac{\max(x, y)}{1 + |\min(x, y)|}, & x < y \\ \frac{\max(x, y)}{1 + \min(x, y)}, & x \geq y \end{cases}$$

- ✓ **Zadatak 8.** Napisati program koji na osnovu unetih vrednosti parametara a, b i c kvadratne jednačine  $ax^2 + bx + c = 0$ , diskutuje prirodu njenih rešenja i prikazuje rešenja u slučaju da su realna.

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