

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 (Linux Terminal ili cmd u Windowsu) i ukucajte:
`java -version`

Provera da li je postoji java kompajler na računaru

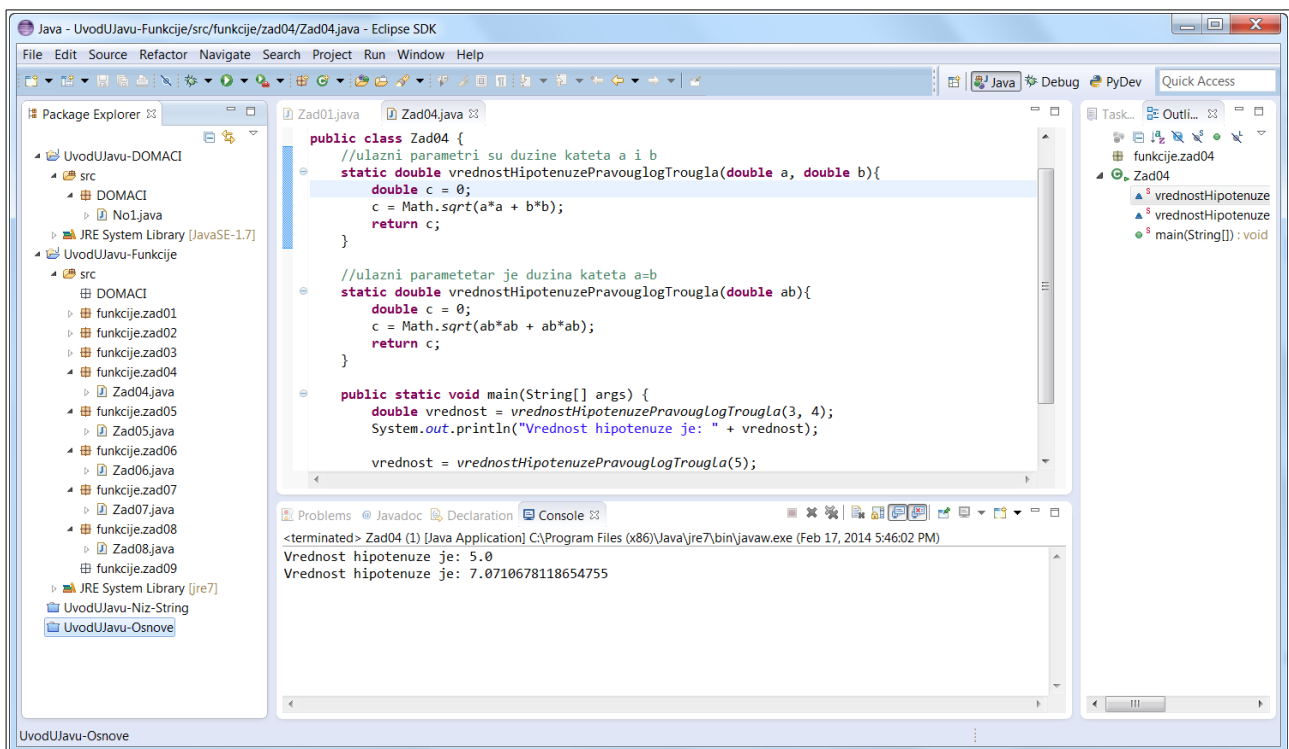
- Otvorite konzolu i ukucajte:
`javac -version`

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 koristićemo 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/traženje 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 poruka na ekran za pokrenuti java program) i *Editor* (zauzima centralni deo površine radnog prostora).

Korisne Eclipse prečice

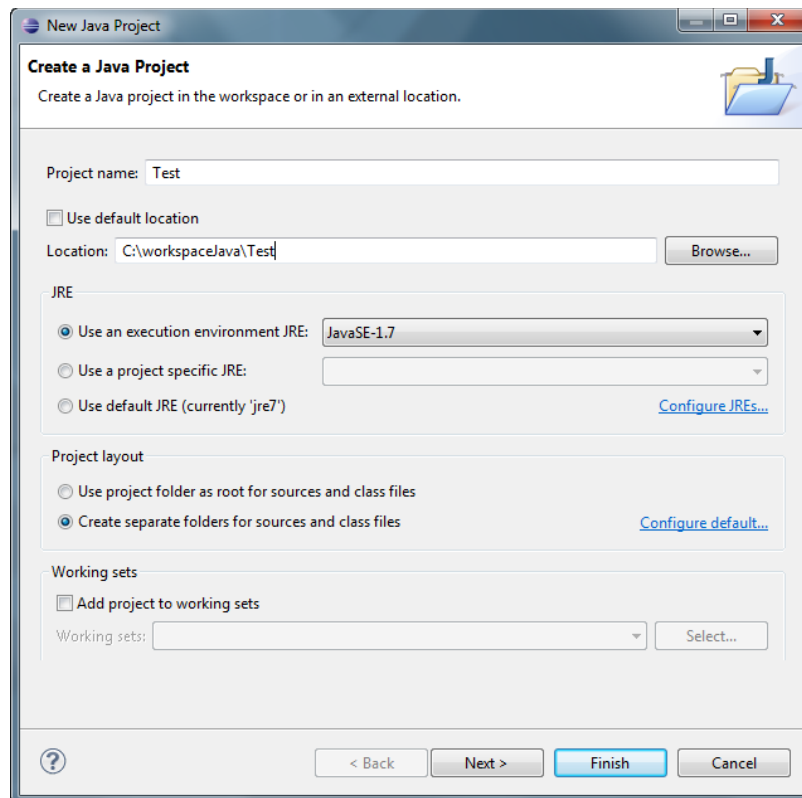
<code>ctrl + space</code>	Auto complete funkcija. Dopunjava započetu ključnu reč. Vrlo korisna jer smanjuje vreme kucanja koda i verovatnoću pojave štamparskih grešaka.
<code>ctrl + d</code>	Brisanje cele trenutne linije.
<code>ctrl+shif+f</code>	Automatsko formatiranje koda. Ova prečica će popraviti nepravilno indentiran kod i prelomiti predugačke linije.
<code>ctrl + /</code>	Komentarisanje trenutne linije ili selektovanih linija

Rad sa projektima

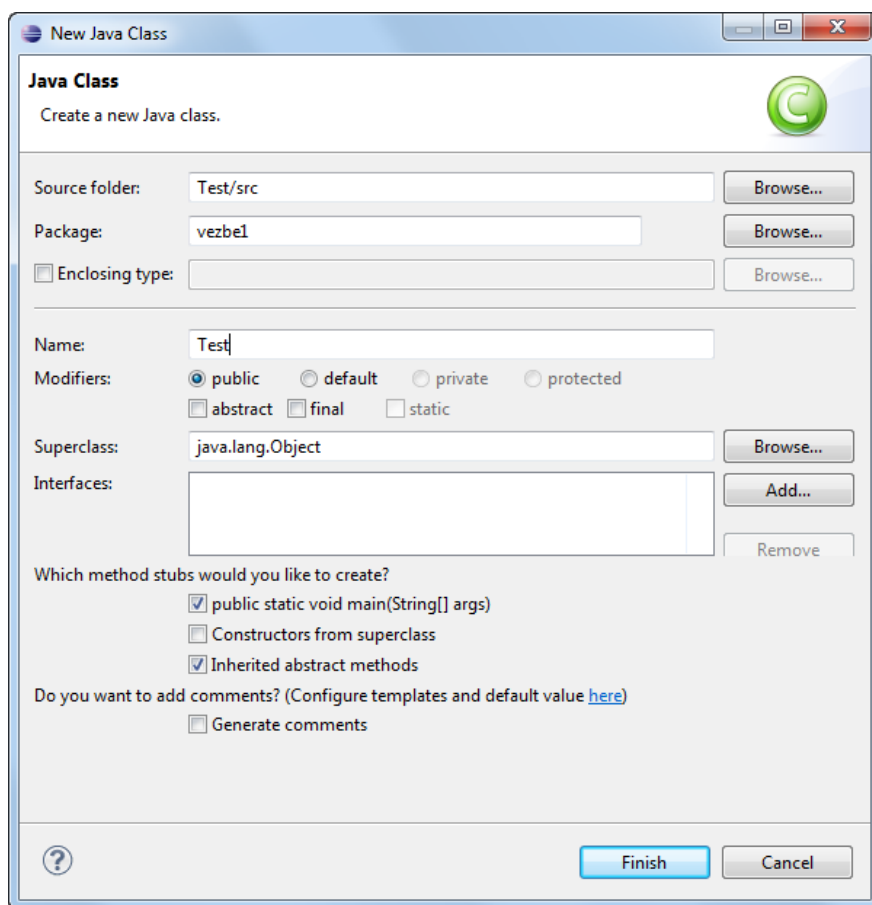
Programski kod se organizuje unutar projekata. Projekat se sastoji od hijerarhije klasa (u skladu sa pravilima definisanja paketa u Java programskom jeziku) i drugih datoteka (npr. slika, txt fajlova, konfiguracionih datoteke itd.). Radni prostor se mapira na odgovarajući direktorijum lokalnog sistema 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. jedino nad njima je 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*.



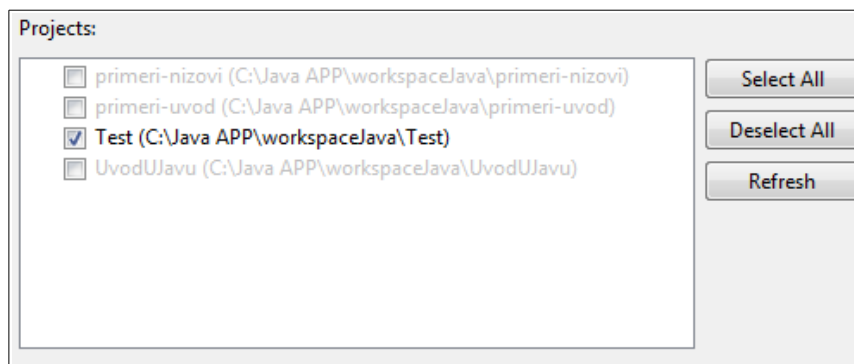
Rad sa paketima u Javi

Paketi predstavljaju hijerarhijsku organizaciju fajlova po folderima i podfolderima unutar projekata. 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. Pozvati opciju *File* → *Import* → *General* → *Existing project into Workspace* i kliknuti na dugme *Next*. Na dugmetu *Browse* postaviti putanju do radnog direktorijuma. Iz liste ponuđenih projekata odabrati onaj koji se importuje (slika ispod). 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 na kojem je projekat kreiran (neki od problema: vrezije jave 6 ili 7, kompatibilnosti kompajlera...). Tada je potrebno prepodesiti podešavanja Jave za uvezani projekat. Prethodno se postiže desnim klikom na projekat pa na stavku *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*.

Pokretanje Java programa

Java program je moguće pokrenuti ako je napisan u klasi koja poseduje main metodu (može se napisati ručno ili kreirati automatski prilikom kreiranja nove klase odabirom stavke *public static void main*).

Pokretanje se vršim desnim klikom na klasu i odabirom stavke *Run As* → *Java Application*.

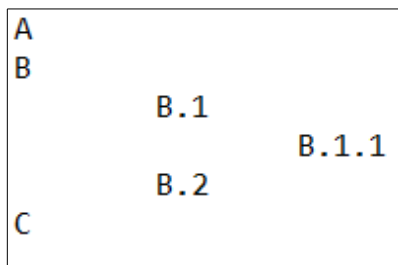
Upotreba programskih komentara:

```
// Komentar naveden u okviru jednog reda

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

Zadaci

1. U Eclipse okruženju kreirati novi Java projekat pod nazivom *JavaUvod* i u njemu paket *osnove*. Kreirati u okviru paketa *osnove.zad01* novu klasu *Zad01*. U okviru main funkcije napišite programski kod koji omogućuje ispis na konzoli kao na slici.



2. Kreirati u okviru paketa *osnove.zad02* novu klasu *Zad02* koju ćemo koristiti za računanje osobina geometrijskih figura. Deklaristati promenljive koja predstavlju: površinu kvadrata 16, stranice pravougaonika dužine 5 i 6, poluprečnik kruga dužine 1,954905637, osnovicu jednakokrakog trougla dužine 4 i krak dužine 6. Izračunajte i ispišite (na konzoli) vrednost promenljivih: obim pravougaonika, površina pravougaonika, obim kruga, površina kruga i obim trougla. Usvojiti da je vrednost za Pi 3.14. Da li možemo izračunati stranicu kvadrata?
3. Klasa *Math*, rad sa matematičkim formulama i konstantama.

```

public static void main(String[] args) {
    //Klasa Math
    /*
        oredjivanje povrsine kruga poluprecnika r=2,
        usvojeno je da je vrednost Pi preuzeta iz klase Math
    */
    double povrsina;
    int r = 2;
    //opcija 1
    povrsina = r * r * Math.PI;
    System.out.println("Povrsina kruga je: " + povrsina);
    //opcija 1, koriscenje funkcije pow(x,y) za odredjivanje stepena r^2
    povrsina = Math.pow(r, 2) * Math.PI;
    System.out.println("Povrsina kruga je: " + povrsina);
}
  
```

Kreirati u okviru paketa *osnove.zad03* novu klasu *Zad03* koju ćemo koristiti za računanje osobina geometrijskih figura korišćenjem klase *Math*. Deklaristati promenljive koja predstavlju: 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:

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

4. Prefiksni i sufiksni oblik korišćenja operatora uvećanja i smanjenja, postoji razlika. Primer uvećanja za jedan *promenljiva++*, *++promenljiva*

```

public static void main(String[] args) {
    int a = 0;
    // prefiksni oblik
    System.out.println("3. Ispis promenljive a je: "+ a++);
    System.out.println("4. Ispis promenljive a je: "+ a);

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

```

5. Logički i relacioni operatori

Logički Operatori: operator "||" znači "ili" (disjunkcija), dok operator "&&" znači "i" (konjunkcija).

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);

```

6. Kontrola toka programa if, if-else, if-else if-else

1. if naredba

```

public static void main(String[] args) {
    /* if naredba
    if ( logicki izraz ) {
        naredbe koje se izvrsavaju 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");
    }
}

```

2. if else naredba

```
public static void main(String[] args) {  
    /* if-else naredba  
    if ( logicki izraz) {  
        naredbe koje se izvrsavaju ako i samo ako logicki izraz rezultuje sa true  
    }  
    else {  
        naredbe koje se izvrsavaju u svim ostalim slucajevima 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");  
    }  
}
```

3. if-else if-else naredba

```
public static void main(String[] args) {  
    /* if-elseif-else naredba  
    if ( logicki izraz1) {  
        naredbe koje se izvrsavaju ako i samo ako logicki izraz 1 rezultuje sa true  
    }  
    else if ( logicki izraz2) {  
        naredbe koje se izvrsavaju ako i samo ako logicki izraz 2 rezultuje sa true  
        N puta moze da se ponovi else if deo  
    }  
    else {  
        naredbe koje se izvrsavaju u svim ostalim slucajevima tj. logicki izraz 1 i 2 su false  
    }  
    */  
    // proverava kretanja kursa eura - pada, raste ili stagnira,  
    // ako se pretpostavlja da ispod 110 pada, izmedju 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");  
    }  
}
```

7. Kreirati u okviru paketa osnove.zad04 novu klasu Zad04. Napisati program koji za datu 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)
- U svim ostalim slučajevima, godina nije prestupna

8. 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*.

```
public static void main(String[] args) {  
  
    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;  
    }  
    //menjati vrednost meseca 1..13  
}
```

Zadaci za domaći

1. Zadaci koji se ne stignu uraditi na vežbama ostaju za domaći.
2. Napisati program koji dato rastojanje u centimetrima (npr. 324) ispisuje u metrima, decimetrima i centimetrima (npr. 3 metra, 2 decimetra i 4 centimetra).
3. Napisati program koji za date stranice $a=3$, $b=4$ i $c=5$ računa površinu i zapreminu kvadra.

$$P=2 \cdot (a \cdot b + a \cdot c + b \cdot c) \quad V=a \cdot b \cdot c$$

4. Kolika je površina kupe prečnika 6 i visine 4? Ako nam je poznata formula

$$P=r^2 \cdot \pi + r \cdot s \cdot \pi$$

gde je s dužina osnovice.