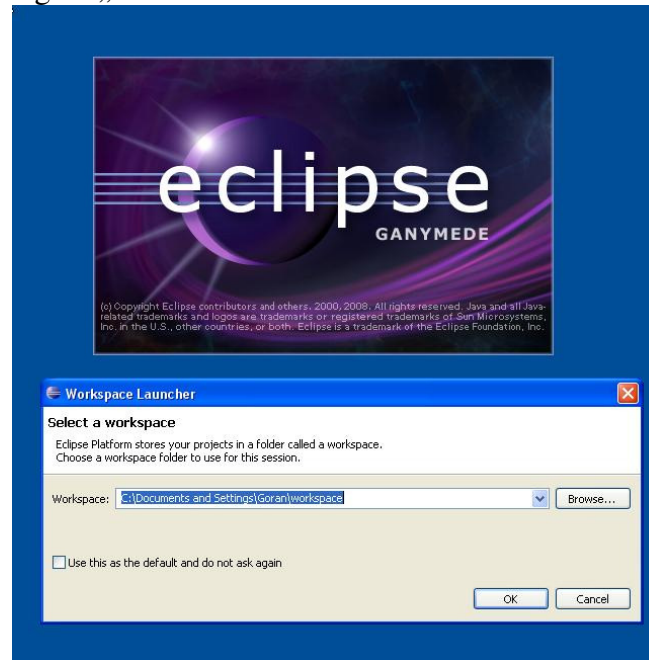
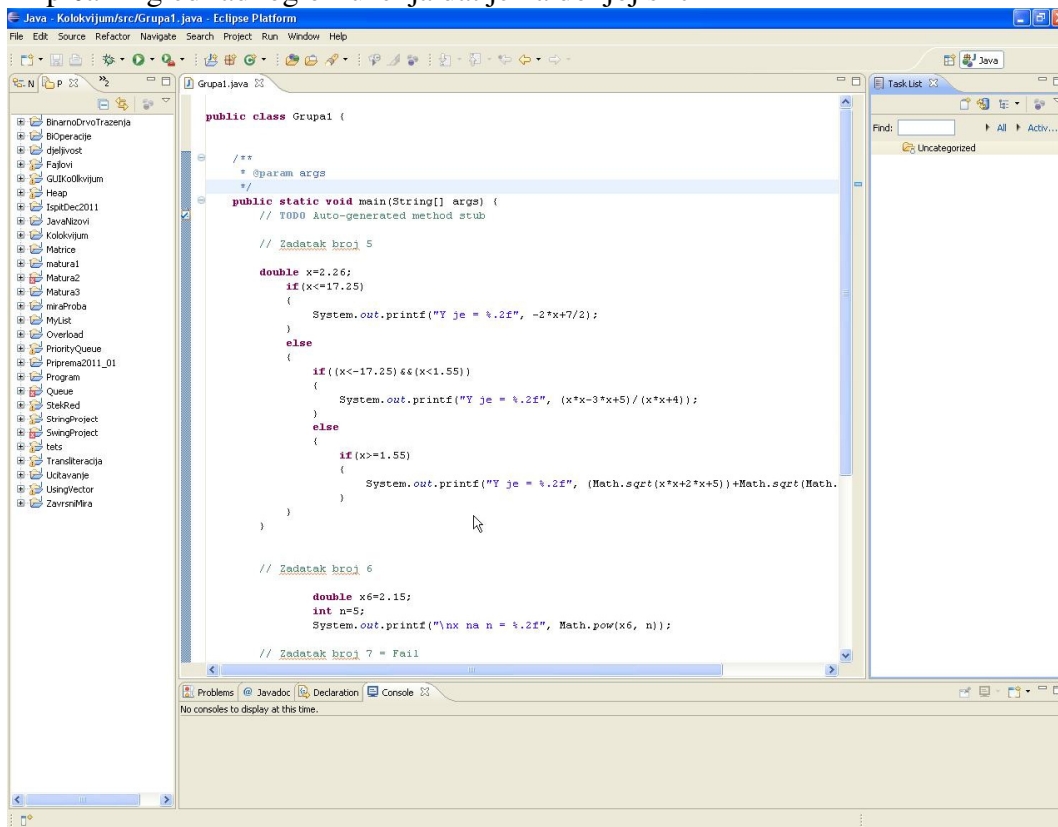


1. Radno okruženje Eclipse

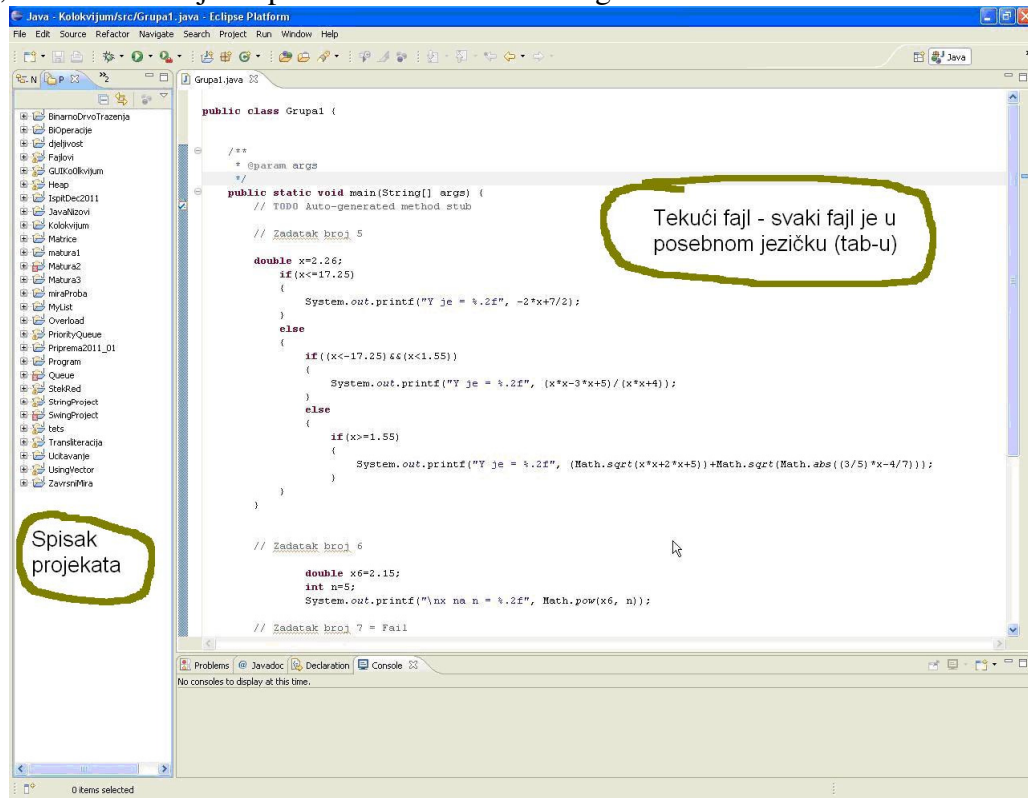
- a. Pokretanje Eclipse – dvostruki klik na eclipse.exe.
- b. Po pokretanju alata, traži se da unesete lokaciju za vaš radni prostor (engl. workspace),
Možete imati više radnih prostora koji su smješteni u različitim folderima ili diskovima vašeg računara. Predlažemo da kreirate folder sa vašim imenom (npr. AnaPeric) i da to bude mjesto na kojem će se čuvati vaši fajlovi. Izaberite za vaš radni prostor folder sa vašim imenom, klikom na dugme „Browse...“.



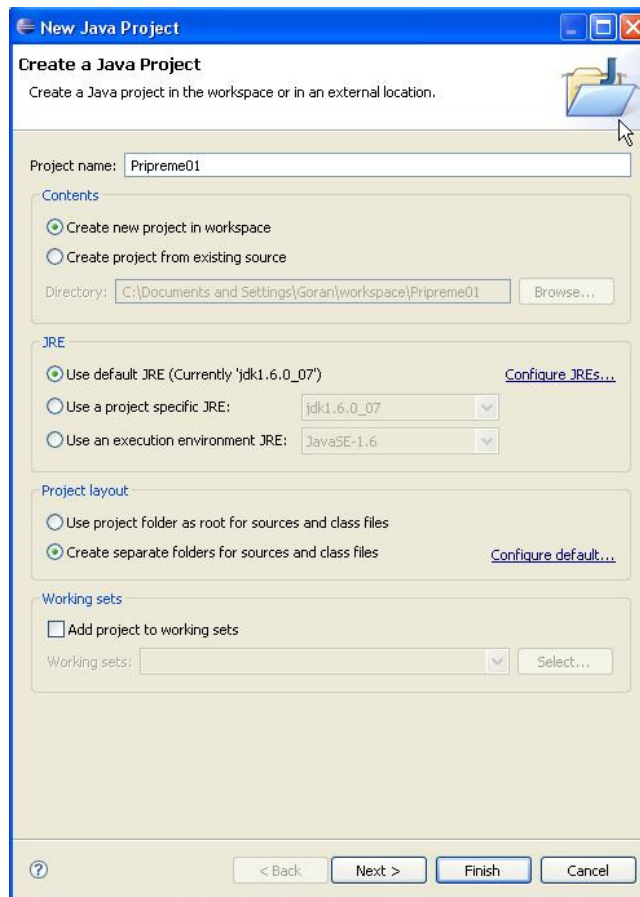
c. Tipičan izgled radnog okruženja dat je na donjoj slici



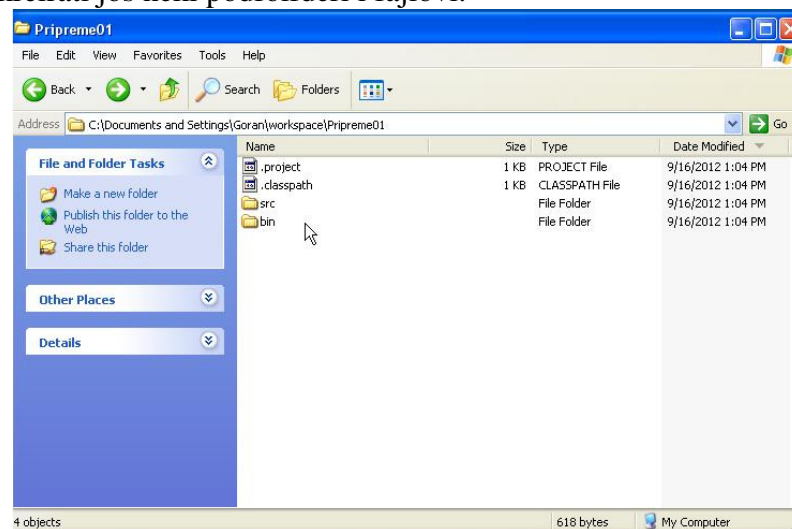
2. Pojedine elemente radnog okruženja možete uklanjati i ponovo prikazivati izborom stavke menija **Window**, podstavka **Show View**. Takođe, moguće je minimizirati pojedine elemente okruženja. Na primjer, radno okruženje na profesorskom računaru izgleda ovako:



3. Kreiranje projekta, klase sa metodom main i klase bez metoda main.
- Kreiranje novog projekta: stavka menija File → New → Java Project; otvara se pomoćnik (engl. Wizard):

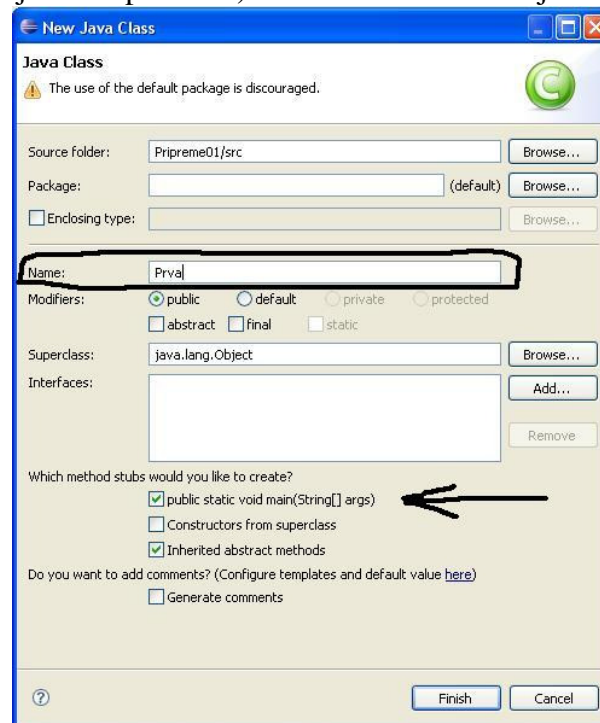


- b. Unesite ime projekta (Project name, u našem primjeru ime je Pripreme01) i kliknite na dugme Finish. Preporuka je da u imenu projekta nema naših slova niti blankova.
- c. U vašem radnom prostoru (workspace) kreiran je folder Pripreme01, a unutar njega podfolderi *src* i *bin* i fajlovi *.project* i *.classpath* (vidi donju sliku). Kasnije će se u folderu Pripreme01 kreirati još neki podfolderi i fajlovi.

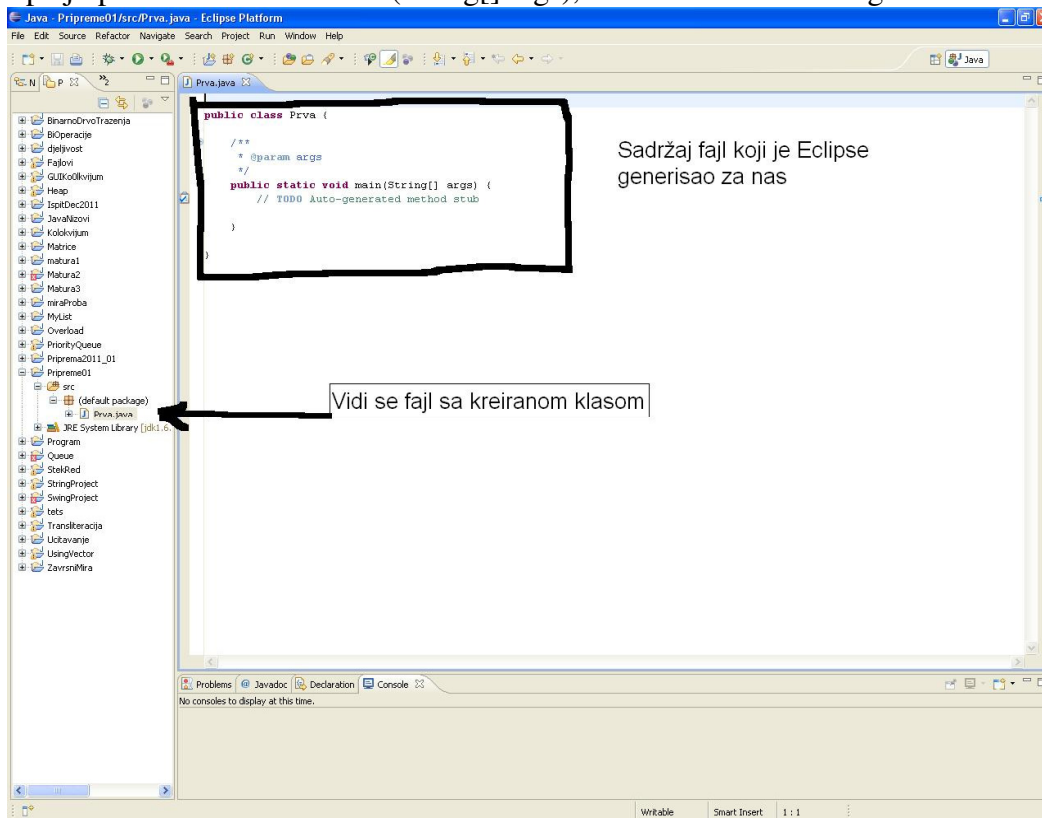


- d. Projekat se sastoji od jedne ili više klasa, pri čemu se svaka klasa nalazi u odvojenom fajlu sa ekstenzijom *.java*. Imena klasa se pišu sa početnim velikom slovom (npr. Prva, Zadatak1, StudentPrveGodine...). Ime klase mora početi velikim slovom, a zatim slijede slovo , cifra ili donja crta. Iako je moguće koristiti naša slova, to se ne preporučuje.

- e. Kreiranje nove klase: prvo se u spisku projekata selektuje projekat u koji želimo da dodamo klasu (kliknite na projekat Priprema01). Izaberite stavku menija File →New→Class:



- f. Popuniti polje za ime klase (polje Name, u našem primjeru ime klase je Prva) i čekirati opciju public static void main(String[] args), a zatim kliknuti na dugme Finish.

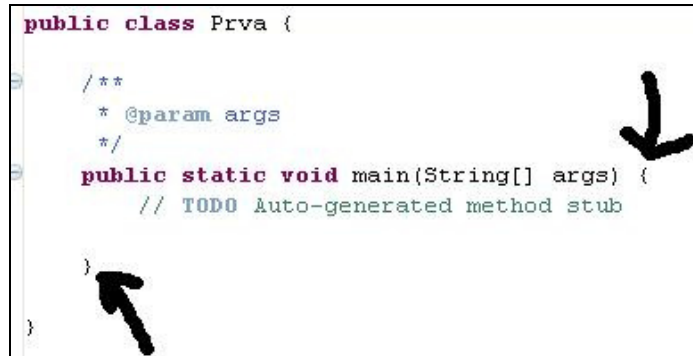


- g. Ponoviti ovaj postupak i kreirati klasu Druga, ali ovaj put **nemojte** čekirati opciju public static void main(String[] args).

4. Struktura Java-programa

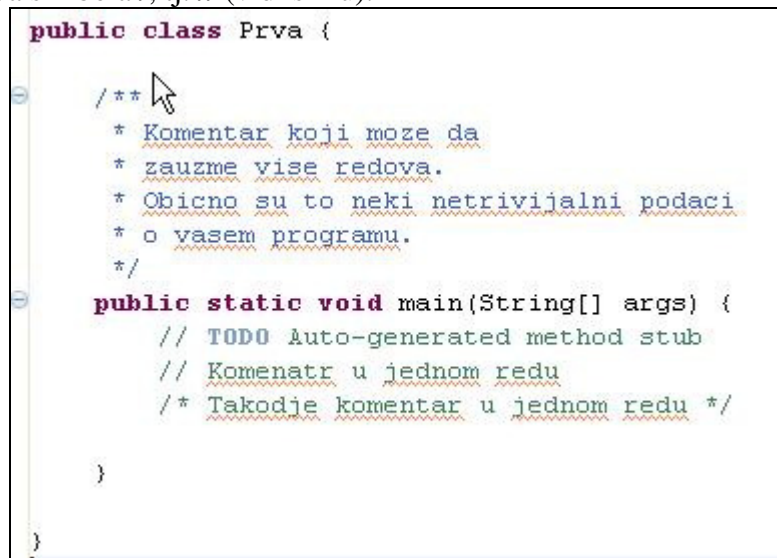
- a. Metod `main` u klasi je mjesto gdje se pokreće vaš program. Makar jedna klasa u projektu mora imati metod `main`. Za sada ćemo sve operacije izvoditi samo unutar metoda `main` (vidi sliku – sve naše operacije moraju biti između označenih zagrada). Kasnije će biti objašnjeno šta je to metod.

```
public class Prva {  
  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
    }  
}
```



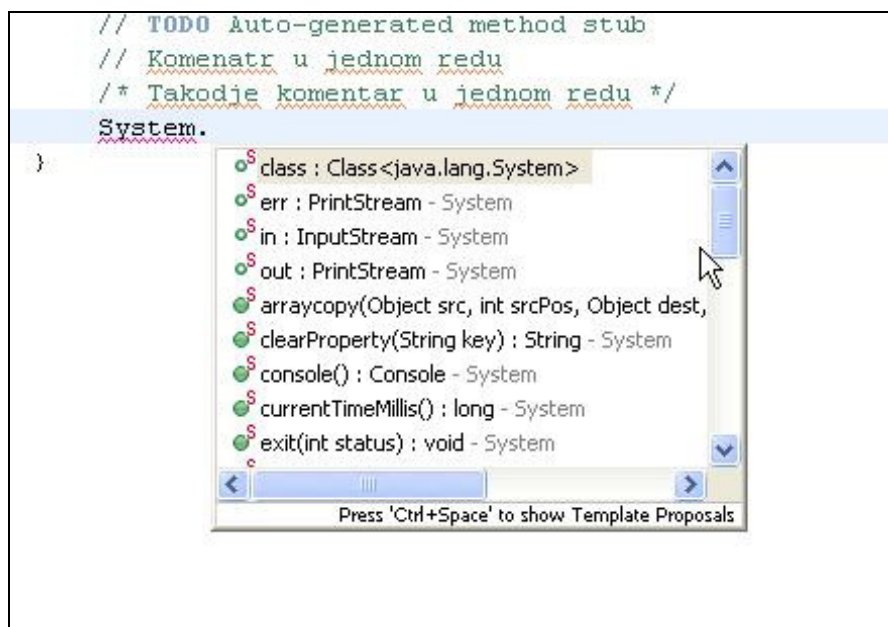
- b. Svi redovi koji se nalaze između simbola `/**` i `*/` su komentar, Komentar može zauzimati više redova (vidi sliku). Ako komentar zauzima samo jedan red, onda se taj red može označiti sa dva simbola `/`, tj. `//` (vidi sliku).

```
public class Prva {  
  
    /**  
     * Komentar koji može da  
     * zauzme više redova.  
     * Obično su to neki netrivialni podaci  
     * o vašem programu.  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        // Komentar u jednom redu  
        /* Takodje komentar u jednom redu */  
  
    }  
}
```

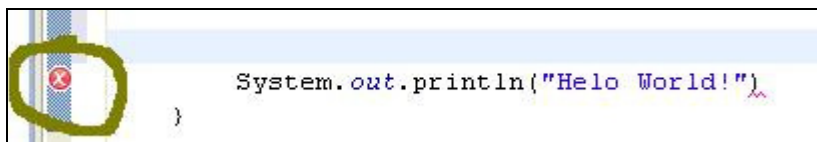


5. Štampanje rezultata: `print` i `println`

- a. Štampanje rezultata se obavlja preko sistemske klase `System` i njenog objekta `out`, primjenom metoda `print` i `println`. Obratite pažnju na razliku između `print` i `println`.
- b. Eclipse vam pruža pomoć pri kucanju teksta. Čim otkucate „System.“, otvara se pomoćna lista koji prikazuje sadržaj klase `System`. Ako otkucate jedno slovo, lista prikazuje samo one elemente koji počinju tim slovom, itd. Ako otkucate npr. `ca`, lista prikazuje samo one elemente koji počinju sa `ca`. Kada pronađete željeni element, pritisnite `Enter` i Eclipse će otkucati taj element umjesto vas. Ovo važi ne samo za klasu `System`, već i za sve druge klase, uključujući i one koji vi napišete.
- c. Koristićemo metode `print(String)` i `println(String)`. Oba metoda štampaju svoj argument u jednom redu, ali `println` po završetku štampanja pređe u sljedeći red, gdje se obavlja sljedeće štampanje. Pored ova dva metoda, postoji i tzv. formatirano štampanje, pomoću metoda `printf` (o ovom metodu govorićemo nešto kasnije).



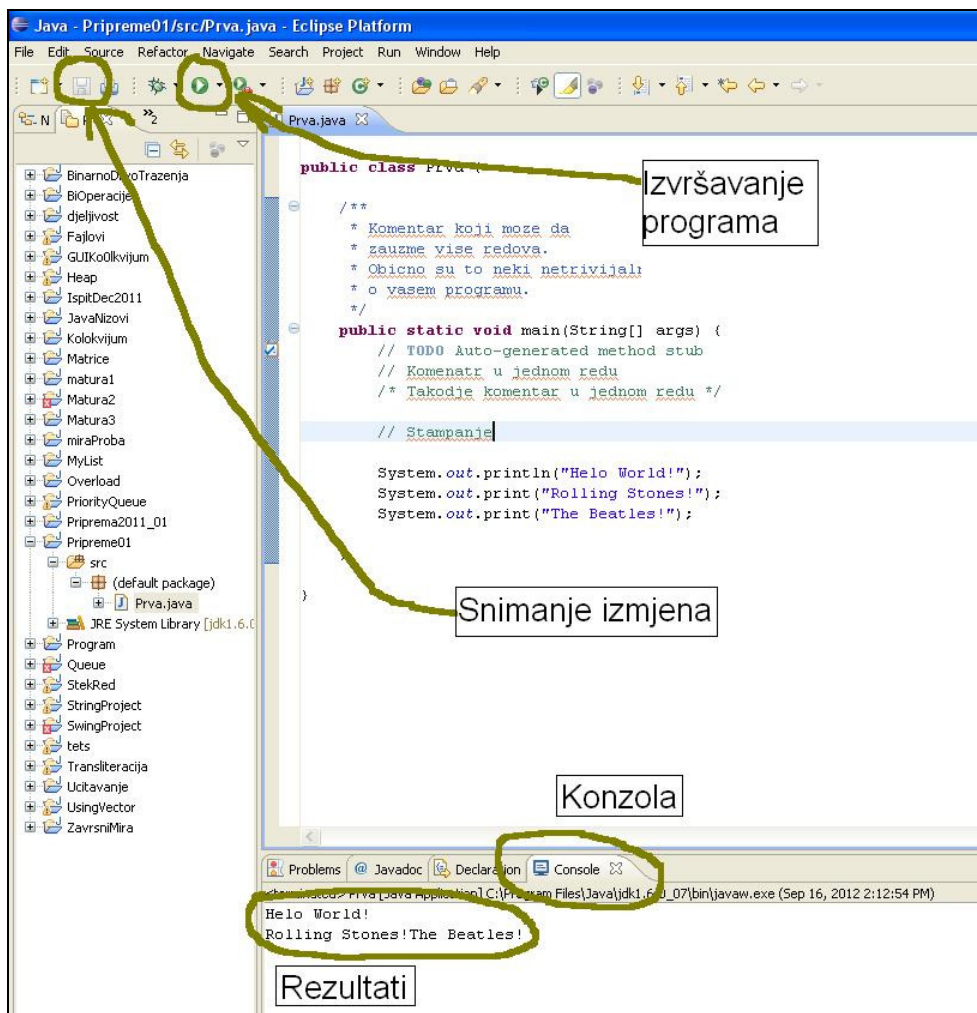
- d. Stringovi su nizovi karaktera i moraju biti pod dvostrukim navodnicima. Primjeri stringova su "Marko Perovic", "a 123 fg", "1245", "Petar\nJasna\nAna", "a", "A", "1"...
- e. Označeni simbol na slici označava da postoji greška u vašem programu. Klikom na simbol, dobijate opis greške. U ovom primjeru, morate na kraju reda unijeti simbol ; (tačka-zapeta).



- f. Otkucajte sljedeći tekst, snimite izmjene u fajlu i pogledajte na izlazu (konzoli) rezultate izvršavanja.

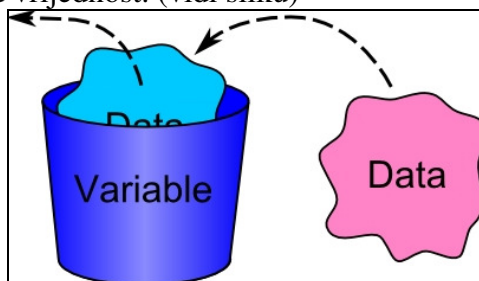
```
System.out.println("Helo World!");
System.out.print("Rolling Stones!");
System.out.print("The Beatles!");
```

- g. U prvom redu izlaza štampan je tekst "Helo World!", bez navodnika. Kako je prva naredba bila println, prešli smo u sljedeći red gdje je štampano "Rolling Stones!", takođe bez navodnika. Kako je to urađeno pomoću naredbe print, sljedeće štampanje nastavlja u istom redu, pa dobijamo da je u drugom redu "Rolling Stones!The Beatles".
- h. I print i println mogu da štampaju samo jedan string. Ako želimo da štampano više stringova, moramo koristiti više puta print i println, ili te stringove nadovezati u jedan primjenom operatora +. Npr, "The"+"Rolling Stones"+56 daje "TheRolling Stones56" (obratite pažnju na nedostatak razmaka).
- i. Donja slika prikazuje izgled okruženja poslije uspješnog pokretanja programa:



6. Dekaracija promjenljivih, tip promjenljive

- a. Promjenljiva je ime za neku memorijsku lokaciju. Sadržaj te lokacije, tj. vrijednost promjenljive se mijenja tokom vremena. Dodjeljivanje nove vrijednosti promjenljivoj uništava njenju postojeću vrijednost. (vidi sliku)



- b. Kada vam treba promjenljiva, morate zatražiti memoriju koja će čuvati vrijednost te promjenljive. Računaru nije potrebna ista količina memorije za npr. jedno slovo i jedna realan broj. Zbog toga morate navesti kakvog je tipa vrijednost koju želite čuvati u promjenljivoj. Dakle, morate navesti tip i ime promjenljive, i moguće zadati njenu početnu vrijednost.
- c. Java podržava sljedeće primitivne tipove podataka (tabela iz knjige „Java Programiranje“, autora Dejana Živkovića):

| Tip podataka | Veličina vrednosti | Opseg vrednosti |
|--------------|--------------------|-------------------------|
| byte | 1 bajt | [-128, 127] |
| short | 2 bajta | [-32768, 32767] |
| int | 4 bajta | $[-2^{32}, 2^{32} - 1]$ |
| long | 8 bajtova | $[-2^{64}, 2^{64} - 1]$ |
| float | 4 bajta | $\approx \pm 10^{38}$ |
| double | 8 bajtova | $\approx \pm 10^{308}$ |
| char | 2 bajta | — |
| boolean | 1 bit | — |
| void | — | — |

- d. Ovi tipovi su primitivni jer se njima predstavljaju proste vrijednosti, koj se ne mogu dalje dijeliti na manje cjeline.
- e. Vodite računa da je skup cijelih brojeva u računar konačan, za razliku od matematike, gdje je on beskonačan. Takođe, realni brojevi se u računar predstavljaju sa greškom. Realni brojevi se pišu sa decimalnom tačkom umjesto decimalne zapete.
- f. Primjeri deklarisanja promjenljivih:

```
int i, j, k = 32;
long n = 0L;
boolean indikator = false, flag = true, isPrime;
float a = 3.4f, b, c = 0.1234f;
double visina, sirina, m123, ugaoRotacije = 67.88;
```

7. Naredba dodjeljivanja

- a. Opšti oblik naredbe je: promjenljiva = izraz
- b. Prvo se izračuna izraz, pa se njegova vrijednost dodijeli promjenljivoj.
- c. Primjer:

```
k = 32;
p = visina * sirina;
m123 = k = 123;
k = j/10 + k%10;
a = (i+j+5.62)*(h-1/2) - a/2;
boolean t = (a+b>c) && (b+c>a) && (c+a>b);
```

- d. Aritmetički izrazi – sabiranje (+), oduzimanje (-), množenje (*), dijeljenje (/), moduo – ostatak pri dijeljenju (%)
- e. Relacioni operatori: manje (<), veće (>), manje ili jednako (<=), veće ili jednako (>=), jednako (==) i različito(!=).
- f. Logički veznici: && (konjunkcija, I, AND), || (disjunkcija, ILI, OR), ! (negacija, NE, NOT)

- g. Prioritet operatora prikazan je u sljedećoj tabeli (preuzeto sa sajta <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>)

| Prioritet operatora (operator precedence) | |
|---|--|
| Operator | Prioritet |
| Postfiksni (postfix) | <code>expr++ expr--</code> |
| Unarni (unary) | <code>++expr --expr +expr -expr ~ !</code> |
| Multiplikativni (multiplicative) | <code>* / %</code> |
| Aditivni (additive) | <code>+ -</code> |
| Pomjeranja (shift) | <code><< >> >>></code> |
| Relacioni (relational) | <code>< > <= >= instanceof</code> |
| Jednakost (equality) | <code>== !=</code> |
| Bit po bit I (bitwise AND) | <code>&</code> |
| Bit po bit XOR (bitwise exclusive OR) | <code>^</code> |
| Bit po bit XOR (bitwise inclusive OR) | <code> </code> |
| Konjunkcija (logical AND) | <code>&&</code> |
| Disjunkcija (logical OR) | <code> </code> |
| Ternarni operator (ternary) | <code>? :</code> |
| Dodjeljivanja (assignment) | <code>= += -= *= /= %= &= ^= = <<= >>= >>>=</code> |

- h. Neke od ovih operatora nismo obradili – govorićemo o njima u sljedećim sesijama.
- i. Svi operacije se izvršavaju slijeva udesno, osim operatora dodjeljivanja koji se izvršavaju zdesna ulijevo. Npr, $7+4+8$ se izračunava kao $(7+4)+8$.
- j. Vodite računa da ako su svi argumenti u izrazu cijeli brojevi, rezultat je također cio broj, čak i ako se koristi dijeljenje. Npr. $13/4$ daje rezultat 3, a ne 3.25. Ako želimo tačan rezultat, moramo pisati $13.0/4$ ili $13/4.0$ ili $13.0/4.0$. Obratite pažnju da je rezultat izraza $13/4*2.0$ jednak 6.0 a ne 6.5.
8. Klasa `Math`
- Konstanta `PI` – `Math.PI`
 - Metod `pow` – stepenovanje brojeva (a^b je `Math.pow(a, b)`).
 - Metod `sqrt` – kvadratni korijen (\sqrt{x} je `Math.sqrt(x)`)
9. Zadaci (svi zadaci su unutar metoda `main`)
- Šampati poruku „Hello World! Vase ime i prezime“ u jednom redu.
`System.out.println("Hello World! Ana Petrovic.");`
 - Šampati dan, mjesec i godinu rođenja, po jedan broj u redu.
`System.out.println(15);`
`System.out.println(4);`

```
System.out.println(1998);
```

- c. Štampati dan, mjesec i godinu rođenja, u jednom redu.

```
int dan = 15, mjesec = 4, godina = 1998;  
System.out.println(dan + " " + mjesec + " " + godina);
```

- d. Za date cijele brojeve a i b štampati njihov zbir, razliku, proizvod, količnik i ostatak pri dijeljenju.

```
int a = 13, b = 4;  
System.out.println("Zbir je " + (a+b));  
System.out.println("Razlika je " + (a-b));  
System.out.println("Proizvod je " + (a*b));  
System.out.println("Kolicnik je " + (a/b));  
System.out.println("Ostatak je " + (a%b));
```

- e. Izračunati površinu i obim pravougaonika ako su zadate cjelobrojne dužine njegovih stranica.

```
a = 5, b = 8;  
int obimPravougaonika = 2*(a+b);  
int površinaPravougaonika = a*b;  
System.out.println("Povrsina=" + površinaPravougaonika + "  
Obim=" + obimPravougaonika);
```

- f. Izračunati dužinu hipotenuze traougla ako su date dužine njegovih kateta a i b (definisani u prethodnom zadatku).

```
double c;  
c = Math.sqrt(a*a+b*b);
```

- g. Izračunati dužinu duži ako su date koordinate njenih krajnjih tačaka.

```
double x1 = 5, y1 = -1.25; // koordinate prve tacke  
double x2 = 5, y2 = -1.25; // koordinate druge tacke  
double duzina;  
// Koristimo Pitagorinu teoremu  
duzina = Math.sqrt((x2-x1)*(x2-x1)+ (y2-y1)*(y2-y1));  
System.out.println("Duzina=" + duzina);
```

- h. Napisati program koji za dati cio broj štampa tekst kao u primjeru. Razmaci, mala i velika slova su važni.

| Ulaz | Izlaz |
|------|--|
| 179 | The next number for the number 179 is 180. The previous number for the number 179 is 178. |

```
int n = 4713;  
System.out.println("The next number for the number " + n + "  
is " +(n+1) + ".");  
System.out.println("The previous number for the number " + n  
+ " is " +(n-1) + ".");
```

- i. Za dati četvorocifreni prirodni broj m odrediti njegovu cifru stotina.

```
int m = 4713;  
// Ovo je jedan način da se riješi zadatak  
int cifraStotina = (m%1000)/100;
```

- j. Za data tri pozitivna realna broja aa, bb i cc odrediti da li se može konstruisati trougao čije su stranice dati brojevi.

```
double aa = 3.0, bb = 4.0, cc = 5;  
// Koristimo nejednakost trougla  
boolean trougao = (aa+bb>cc) && (bb+cc>aa) && (cc+aa>bb);
```

Literatura:

1. Java tutorial: <http://docs.oracle.com/javase/tutorial/java/TOC.html>
2. <http://www.singipedia.com/content/3189-Osnove-Java-programiranja>
3. <http://www.singipedia.com/content/2194-Java-programiranje>