



7

Programi u Javi

- Klasa i program
- Metoda `main`
- Izvršavanje Java programa iz komandnog prompta

Klasa i program

Sve što smo do sada radili svodilo se je na pisanje metoda, koje smo pozivali iz sučelja BlueJ razvojnog okruženja. Pomoću tog okruženja smo mogli metodi proslijediti parametre i na kraju vidjeti što metoda vraća. Nadalje smo naučili kako kreirati čitave klase. Iz tih klasa smo kreirali objekte i nad njima izvršavali metode, ali sve na razini metoda koje smo izvršavali iz BlueJ razvojnog okruženja.

Cilj programiranja je napraviti program koji će se moći samostalno izvršavati na računalu korisnika. Konkretno, htjeli bismo napisati program koji će se moći izvršavati izvan BlueJ razvojnog okruženja.

Program u Javi je isto tako jedna klasa. Da bi neka klasa postala program treba imati još jednu metodu koju nazivamo **main**. Dakle, da bi neka klasa bila program koji ćemo moći izvršavati neovisno o BlueJ razvojnog okruženju, ta klasa treba imati metodu **main**.

Kada želimo pokrenuti program napisan u Javi prvo se traži metoda **main** i izvršava se. Ukoliko želimo pokrenuti klasu koja nema metodu **main**, doći će do pogreške i klasa se naravno neće pokrenuti.

Za one koji su programirali u Pascalu, **main** je u stvari ekvivalent glavnom programu u Pascalu. U Pascalu smo isto tako mogli definirati svoje funkcije i procedure (metode), koje smo pozivali u glavnom programu. Pokretanjem programa se je u stvari izvršavao glavni program. Da nismo imali glavnom programa, isto tako bi došlo do greške.

Metoda `main`

Već smo rekli da će neka klasa biti program ako između ostalih ima metodu `main`.

Opći oblik definicije metode `main` je:

```
public static void main (String[] s)
{
    definicija_metode;
}
```

Ova metoda je izuzetno važna stoga ćemo ju detaljno objasniti. Osobito je bitno zaglavlje metode (prvi red), stoga ćemo objasniti značenje jedne po jedne riječi unutar konteksta metode `main`.

Riječ s kojom započinje definicija metode `main` je **public**. O njoj smo već govorili. Prisjetimo se: **public** općenito znači da je neki element klase javan i može mu se pristupiti i iz drugih klasa. Osim što može biti **public**, elementi klase mogu biti još i **private** i **protected**.

Ukoliko je neki element **private**, tada tom elementu ne možemo pristupiti izvan klase u kojoj se taj element nalazi.

Negdje između **public** i **private** elemenata su **protected** elementi. Ako je neki objekt **protected**, tada takvom elementu možemo pristupiti u klasi koja nasljeđuje dotičnu klasu, ali ne i u ostalim klasama.

U kontekstu metode `main`, bitno je da je ona **public** iz razloga što nju pokrećemo prvu prilikom pokretanja klase. Nju pokreće Java Virtual Machine.

Sljedeća riječ u nizu je **static**. Da je neka metoda **static** znači da ju je moguće pozivati na razini klase. Prisjetimo se, kada smo htjeli pozvati metodu `Povrsina ()` klase `Trokut`, morali smo kreirati instancu klase `Trokut` i nad tom instancom pozvati metodu `Povrsina ()`.

```
Trokut t = new Trokut (a, b, c);
double p = t.Povrsina ();
```

a nismo mogli pisati

```
double p = Trokut.Povrsina ();
```

s druge strane imamo klasu `Math` i npr. metodu `sqrt ()` koja je definirana nad tom klasom. Želimo li računati drugi korijen nekog broja, ne moramo kreirati instancu klase `Math` i nad njom pozivati metodu `sqrt ()`, već metodu `sqrt ()` pozivamo na razini klase `Math`.

```
double x = Math.sqrt (7);
```

U prvom slučaju metoda `Povrsina ()` nije bila **static** i stoga smo ju morali pozivati na razini objekta koji je bio tipa `Trokut`. U drugom slučaju je metoda `sqrt ()` **static** metoda i ona se može pozivati na razini klase.

void znači da metoda ništa ne vraća. Na kraju metode koja je tipa **void** nije potrebno pisati **return**.

`main` je ime metode.

String [] s se odnosi na parametre metode. Metoda **main** kao parametre ima niz stringova.

Primjer 7 – 1:

Napišimo program koji će zbrajati dva prirodna broja i ispisivati njihov zbroj.

Rješenje:

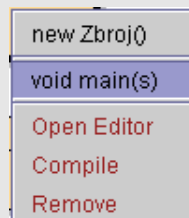
Budući da trebamo napisati program koji će zbrajati dva prirodna broja, napisat ćemo klasu koja će morati sadržavati metodu **main**. Brojeve ćemo unijeti kao parametre metode **main**.

```
public class Zbroj
{
    public static void main (String[] s)
    {
        int a = Integer.parseInt(s[0]);
        int b = Integer.parseInt(s[1]);
        System.out.println( a + b );
    }
}
```

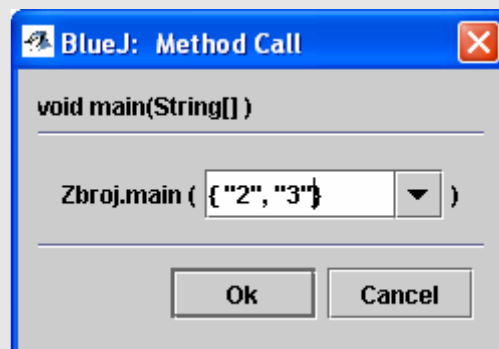
U s obzirom da su parametri metode **main** stringovi, trebamo ih pretvoriti u cijele brojeve i to smo napravili naredbom **Integer.parseInt ()**.

Rezultat smo ispisali naredbom **System.out.println ()**.

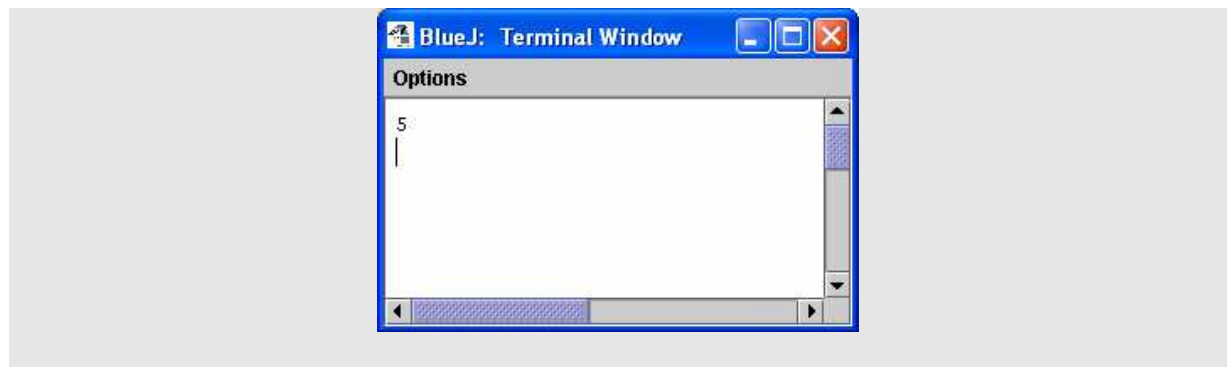
Nakon što smo definirali klasu **zbroj** te ju uspješno kompajlirali, preostaje nam još pokrenuti program. Program ćemo u BlueJ razvojnom okruženju pokrenuti pozivanjem metode **main**.



Nakon pozivanja metode **main** otvorit će nam se okvir gdje ćemo unijeti parametre koje proslijeđujemo metoda **main**. Budući da je parametar niz stringova, brojeve koje želimo zbrojiti zapisat ćemo unutar vitičastih zagrada, unutar navodnika, međusobno odvojene zarezom:



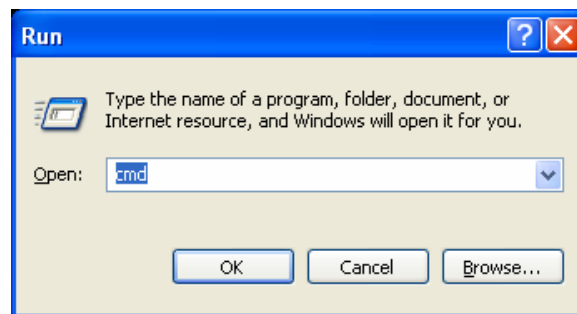
Klikom na gumb **OK** izvršit će se program i rješenje će biti ispisano u posebnom okviru:



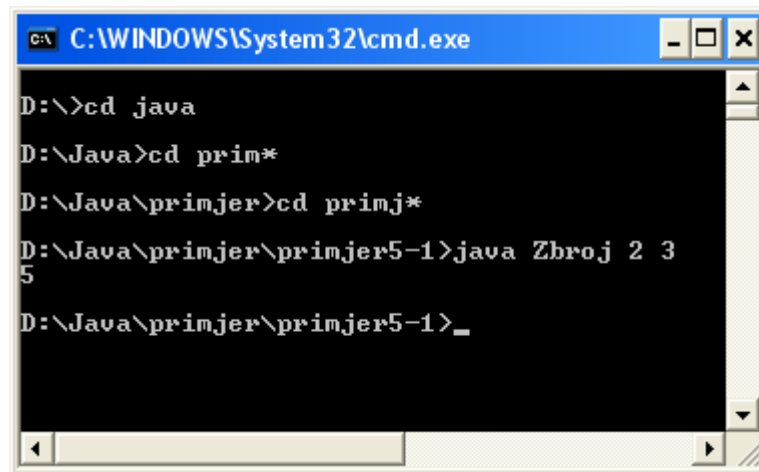
Izvršavanje Java programa iz komandnog prompta

U prošlom primjeru smo napravili program, međutim ponovo smo ga pozvali iz BlueJ razvojnog okruženja. Na samom početku ovog poglavlja rekli smo da nam je cilj pisati programe koji će se moći izvršavati neovisno o BlueJ razvojnom okruženju.

U nastavku ćemo pokazati kako napisani program pokrenuti neovisno o BlueJ okruženju. Prethodni program je bio primjer jednog programa pisan za operativni sustav DOS, tzv. **konzolni program**. Da bismo pokrenuli ovaj program pokrenut ćemo DOS-ov komandni ekran. To možemo napraviti tako da kliknemo na **Start** pa na **Run**, gdje upišemo **cmd** te kliknemo na gumb **OK**:



Otvorit će nam se prozor u kojem zadajemo naredbe upisivanjem. Premjestit ćemo se u mapu u kojoj se nalazi klasa **Zbroj**, koju smo kreirati u primjeru 5 – 1. te ćemo upisati: **java Zbroj 2 3**. Pritiskom na tipku *enter* na ekranu će se pojaviti 5 (zbroj brojeva 2 i 3).



Primijetimo da smo program pokretali naredbom: **java Zbroj 2 3**. Java ovdje znači da pozivamo Javin interpreter, zbroj je ime klase (dokument s nastavkom *class Zbroj.class*) dok su 2 i 3 parametri koje prosljeđujemo programu odnosno metodi **main**.

Napomena:

Da bismo mogli pokrenuti program iz komandnog prompta, on mora biti kompajliran tj. mora postojati dokument s nastavkom *.class*. Program kompajliramo u BlueJ-u.

Zadaci za vježbu

1. Što mora imati neka klasa da bismo ju mogli izvršavati kao program?
2. Zašto metoda `main` mora biti **public** i **static**?
3. Neke od klasa koje si kreirao u prošlim poglavljima pretvori u programe te ih izvrši.
4. Definiraj komandnu aplikaciju, koja će se moći pokretati iz komandnog prompta a aplikacija treba unositi prirodan broj n i :
 - a. ispisivati kvadrate svih prirodnih brojeva do n ;
 - b. zbrajati sve prirodne brojeve do n ;
 - c. ispisivati rastav broja n na proste faktore.