

## PRIMER UPORABE VNOSA IN PRIKAZA

### Program za izračun povprečne hitrosti vozila, ki je v določenem času prepotovalo določeno razdaljo:

Za naš računalnik bomo napisali programsko aplikacijo, v kateri bomo podali čas vožnje in prevožene kilometre, računalnik pa nam bo izračunal in prikazal povprečno hitrost naše vožnje. V fiziki ta problem rešimo z enačbo

$$v = s / t \quad (s - \text{pot, } t - \text{čas, } v - \text{povprečna hitrost}).$$

Čas in pot predstavljata vhodne podatke. Enačbo bomo vključili v naš algoritem, v katerem pa moramo poleg izračuna poskrbeti še za vnos in prikaz podatkov. Vnos se mora zgoditi pred izračunom, prikaz pa po njem.

Primer bomo rešili v vseh treh programskih jezikih.

#### Jezik C#:

```
using System;                                     //knjižnica za razred Console
namespace PovprecnaHitrost
{
    class Program
    {
        static void Main(string[] args)
        {
            //vnos podatkov:
            Console.Write("Vnesi pot v kilometrih (brez decimalk): ");
            int pot = Convert.ToInt32(Console.ReadLine()); //vnos poti
            Console.Write("Vnesi čas v kilometrih (lahko tudi decimalke): ");
            double cas = Convert.ToDouble(Console.ReadLine()); //vnos časa
            //izračun povprečne vrednosti:
            double hitrost = pot / cas;
            //izpis povprečne hitrosti:
            Console.WriteLine("Povprečna hitrost vožnje je " + hitrost + ".");
            Console.ReadLine(); //zaustavitev izvajanja programa
        }
    }
}
```

Vse tri spremenljivke, ki smo jih potrebovali, smo ustvarili neposredno pred uporabo.

Decimalne vrednosti v konzolnem oknu vnašamo z decimalno vejico.

## Jezik C++:

```
using namespace std;           //standardni imenski prostor
#include <iostream>             //knjižnica za vhodni in izhodni tok

int main()
{
//priprava spremenljivk za vnos podatkov:
    int pot;                    //kilometri bodo le cela števila
    double cas;                 //pri urah lahko vnašamo decimalke
//vnos podatkov:
    cout << "Vnesi pot v kilometrih (brez decimalk): ";
    cin >> pot;                 //vnos poti
    cout << "Vnesi čas v urah (lahko tudi decimalke): ";
    cin >> cas;                //vnos časa
//izračun povprečne hitrosti:
    double hitrost = pot / cas;
//izpis povprečne hitrosti:
    cout << "Povprečna hitrost vožnje je " << hitrost << ".";
    cin >> pot;                //zaustavitev izvajanja programa
    return 0;
}
```

Spremenljivko *hitrost*, v katero smo shranili povprečno hitrost, smo ustvarili neposredno pred uporabo, lahko pa bi jo pripravili že na začetku programa skupaj s spremenljivkami za vhodne podatke.

Pri ukazu *cin>>* za zaustavitev lahko uporabimo eno od obstoječih spremenljivk. Njene vrednosti v nadaljevanju ne bomo več potrebovali.

Decimalne vrednosti v konzolnem oknu vnašamo z decimalno piko.

## Jezik Java:

```
package povprecnaHitrost;      //vključimo paket
import java.io.*;              //vključimo knjižnico z metodo za izpis
import java.util.Scanner;      //vključimo knjižnico z metodo za vnos

public class PovprecnaHitrost
{
    public static void main(String args[]) {
        //vnos podatkov:
        Scanner vnos= new Scanner(System.in);
        System.out.println("Vnesi pot v kilometrih (brez decimalk): ");
        int pot= vnos.nextInt(); //vnos poti
        System.out.println("Vnesi čas v urah (lahko tudi decimalke): ");
        double cas= vnos.nextDouble(); //vnos časa
    }
}
```

```
//izračun povprečne hitrosti:  
double hitrost= pot / cas;  
//izpis povprečne hitrosti:  
System.out.println("Povprečna hitrost vožnje je " + hitrost + ".");  
}  
}
```

Če se naš program nahaja v paketu, moramo v zaglavju vključiti paket, v katerem se nahaja. V našem primeru je to paket z imenom *povprecnaHitrost*.

Decimalne vrednosti v izhodnem oknu vnašamo z decimalno vejico.

Pri vnosu podatkov smo najprej uporabili ukaz za izpis, v katerem smo uporabniku podali navodila. Brez teh navodil bi bilo konzolno okno ob zagonu aplikacije prazno in uporabnik ne bi vedel, kaj računalnik od njega pričakuje.

Da se konzolno okno ne zapre preden preberemo rezultat izračuna, moramo v jeziku C# in C++ izvajanje programa zaustaviti z ukazom za vnos podatkov, da konzolno okno ostane odprto. Pri jeziku Javi to ni potrebno, ker se izhodno okno ne zapre.