

# Zapiski iz pouka Osnove programiranja II

## Programiranje Mikrokontrolerjev

Matej Blagšič

1. marec 2018

# Kazalo

1	Osnovno	2
2	Podatkovni tipi	2
3	Branje podatkov	3
4	Pisanje podatkov	3
5	Funkcije	3

# 1 Osnovno

Pri temu predmetu bomo obravnavali jezik C. Za uporabo lahko preneseš okolje Codeblocks z MinGW inštalacijo ali posebej MinGW compiler in poljubno okolje(Jetbrains).

Pomembno je, da imaš predznanje iz prejšnjega polletja pri Javascriptu, saj so tipi spremenljivk, sintaksa in drugo zelo podobno, tako da v detajle o stvarih, ki so enake ne bom šel.

Vsak dokument začnemo z `#include <stdio.h>` za standardne vhodne in izhodne ukaze. Vsaka koda se izvaja znotraj main funkcije:

```
int main(){
    printf("Hello!\n");
    return 0;
}
```

**Prav tako je pomembno uporabiti PODPIČJE za vsakim ukazom/vrstico!!!**

Če začnemo na začetku, opazimo `#include` ukaz. Ta se izvrši, preden se karkoli drugega. V temu primeru lahko vnesemo knjižnice. Te nam olajšajo programiranje s tem, da nam en ukaz izvede več ukazov, ki bi jih morali tipkati na roke. To datoteko/knjižnico navedemo lahko z "datoteka"navednicam. Če pa damo v `<datoteka>`, potem pa išče datoteke v sistemskih mapah okolja. Te datoteke so vrste **header** s končnico **.h**. V našem primeru je knjižnica za pisat in brat podatke - vhodne in izhodne podatke.

To je podobno kot v javascriptu: `<script src="datoteka">`

## 2 Podatkovni tipi

Si pogledjmo zgled:

```
int main(){
    int a;
    float b; //spremenljivka

    printf("Vprisi prvo vrednost");
    scanf("%d", &a);
    printf("Vpisi drugo vrednos");
    scanf("%f", &b);
    printf("%d + %f = %f\n", a, b, a+b);
    return 0;
}
```

C je občutljiv na tip podatkov. To pomeni, da moramo vrsto podatka navesti. To pomeni, da se moramo sami odločiti, kakšen tip podatka bo nosila spremenljivka.

Vemo, da v Javascriptu nismo rabili napisati tipa spremenljivke, le `var`.

## Tipi spremenljivk:

<b>int</b>	Intager: Celoštevilski tip. Uporabimo formatno določilo %d
<b>float</b>	Float: Realna števila. Uporabimo formatno določilo %f
<b>double</b>	Double: Double precision Float: "dvojni float"
<b>char</b>	Char: 1 byte velik karakter

## 3 Branje podatkov

Da nam program prebere podatek, uporabimo funkcijo:

```
scanf("formatno_dolocilo", &spremenljivka);
```

Vidimo, da moramo najprej deklarirati tip podatka, ki ga pričakuje operator Scanf. Potem pa določimo naslovni operator & in nato za njim spremenljivko, ki naj sprejme podatek.

## 4 Pisanje podatkov

Za pisanje podatkov uporabimo funkcijo:

```
printf("formatni_niz", izrazi)
```

Pomembne so tudi ubežne sekvence. To so `\r` `\n` `\t`, ki povejo, kaj se zgodi, ko se text izpiše. `\n` naredi novo vrstico(new line) po besedilu, `\t` je tabulator...

Tako v našem primeru, se a izpiše tam, kjer je njegov %d in b, kjer je %f ter vsota a+b tam, kjer je %f(glej izsek programske kode).

## 5 Funkcije

Funkcije deklariramo:

```
float imeFunkcije(){/*telo funkcije*/return 0;}
```

Opazimo, da funkcijo deklariramo kot float oz. funkcijo, ki vrne realno število. V resnici lahko funkcije definiramo kot karkoli hočemo, glede na to, kaj naj bi vrnila.

Prav tako vidimo, da glavna zanka, v kateri se koda izvaja, je `main`. v tej kodi se izvajajo vsi programi in funkcije. Tako se koda, ki je napisana tu notri, se prevede in spremeni v izvršilno kodo(executable).

Whatafak