
Tema 1

OSNOVNE INFORMACIJE O JEZIKU

- Istorijski
 - 1972 nastao
 - Napisan da bi se napravio ceo UNIX operativni sistem
- Za sta se koristi
 - Iz njega lako možemo da naučimo ostale popularne programske jezike, C++, Javu,..
 - Dosta je brzi od Pythona
 - Može da se koristi za pisanje operativnih sistema ili programa, ali i za pravljenje aplikativnog softvera
 - C ne koristi OOP paradigmu ali koristi neke druge bitne koncepte u programiranju koje ćemo koristiti u Javi ili C++.
- Koje okruženje koristimo
 - Može da se koristi bilo koji tekst editor ali da odvojeno imamo instaliran kompajler za C
 - Mi ćemo koristiti CodeBlocks (u kome možemo i C i C++ da kuckamo)
 - <http://www.codeblocks.org/>. Download **mingw-setup.exe** i instalirati (tekst editor i kompajler)
- Razlike - Python i C
 - Više se potencira na različitim tipovima podataka
 - C ignorise spejseve
 - Funkcionalno programiranje kao i Pythonu - pri pokretanju programa, pokreće se main() funkcija
 - Sintaksna pravila su drugačija
 - Nemamo dvotacke za blokove nego viticaste zagrade
 - Svaka naredba se završava sa tacka zarezom
 - Ostale naredbe ćemo u toku rada spominjati i upoređivati sa Pythonom
 - Komentari su umesto # naznaceni sa // za jedan red. Komentari za više redova su /* i */
 - Kompajljanje kao proces
 - Nakon napisanog koda, kod se **BUILDUJE** nakon čega se prave **OBJEKTNI I EXE fajlovi**
 - Nakon toga se program **RUN-uje**, čime se pokreće exe fajl
 - Obj i exe nisu postojali kod Pythona
 - U ovom slučaju sada imamo prevodjenje na masinski jezik

INSTALACIJA

- Potrebno je voditi učenika kroz instalacioni proces kod njega na računaru (online nastava)
- Potrebno samo opisati proces instalacije učenicima (nastava u učionici)

PRIMER 1:

```
#include <stdio.h>

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

OBJAŠNJENJA:

- include je kao u pythonu import, uz nova sintaksta pravila
- Svaki program mora biti napisan kroz funkciju main() koja se uvek pokrece
- Funkcija uvek ima povratnu vrednost koja je integer, zato imamo return deo
- Int je oznaka za integer brojeve kojima se, po sintaksi funkcije u C objasnjava da je povratni tip integer (to nismo imali u Pythonu)
- Ovo sa int (naglasavanje tipa povratne vrednosti funkcije) se koristi i u drugim popularnim jezicima
- Main fja moze da ima i argumente, ali o tome cemo nekada kasnije pricati i koristiti. Tada se argumenti prosledjuju pri samom pokretanju programa

PRIMER 2: PRINTF funkcija

```
#include <stdio.h>

int main() {
    printf("Hello!");
    printf("Danas pocinjemo da ucimo C.");
    return 0;
}
```

OBJAŠNJENJA:

- Printf ne stavlja kao u pythonu ono defaultno enter na kraju printa
- Ovde se printovi lepe ukoliko im enter nije naglasen
- Izmeniti liniju sa Hello: `printf("Hello World!\n");`

PRIMER 3: PROSTI TIPOVI PODATAKA I NJIHOV ISPIS

```
#include <stdio.h>

int main() {
    int broj = 15;
    float pi = 3.14;
    char slovo = 'p';
}
```

```

char rec[] = "Algoritmika";
printf("%d\n", broj);
printf("%f\n", pi);
printf("%.1f\n", pi); // show 1 digit
printf("%.2f\n", pi); // show 2 digits
printf("%.4f", pi); // show 4 digits
printf("%c\n", slovo);
printf("%s\n", rec)
printf("Prvi broj je %d, a drugi %f.", broj, pi);
return 0;
}

```

OBJAŠNENJA:

- Printfu mora da se naglasi u kom formatu zelimo da ispisemo koju promenljivu
- Bool ne mozemo da ispisemo
- Char je uvek za samo 1 znak jer ima 1 bajt, %c
- Int je 2 ili 4 bajta (%d ili %i)
- Float je 8 bajtova (%f)
- Double je 8 bajtova (%lf)
- String se pravi preko niza charova (%s)
- Kada zelimo da ubacimo vise promenljivih u string, samo ih dodajemo redom (kao u format funkciji u Pythonu)
- Primecujemo da se prilikom definisanja promenljive mora da naznaci kog je tipa
- Moguce je staviti i zajedno vise defiinicija
- Za imena promenljivih vaze ista pravila kao u Pythonu
- Preciznost za float brojeve

PRIMER 4: OPERATORI

```

#include <stdio.h>

int main() {
    int x = 5, y = 6, z = 50;
    printf("%d", x + y + z);

    int x = 5;
    int y = 2;
    int sum = 5 / 2;
    float rez = 5/2;

    printf("%d\n", sum); // Outputs 2
    printf("%.4f\n", rez); // Outputs 2
    float rez2 = (float) 5 / 2;

    printf("%f", rez2); // 2.500000
}

```

```
// Automatic conversion: float to int
int myInt = 9.99;

printf("%d", myInt); // 9

}
```

OBJAŠNJENJA:

- Paziti sa ovim promenama iz inta u float
- Probati i ostale operatore
- Probati += iz pythona
- Operator // ne postoji ovde jer je to znak za komentar
- Pored aritmetickih postoje i logicki i bitski operatori, koje cemo spominjati narednih casova

Operator	Name	Description	Example	Try it
+	Addition	Adds together two values	x + y	Try it »
-	Subtraction	Subtracts one value from another	x - y	Try it »
*	Multiplication	Multiplies two values	x * y	Try it »
/	Division	Divides one value by another	x / y	Try it »
%	Modulus	Returns the division remainder	x % y	Try it »
++	Increment	Increases the value of a variable by 1	++x	Try it »
--	Decrement	Decreases the value of a variable by 1	--x	Try it »

Operator	Example	Same As	Try it
=	x = 5	x = 5	Try it »
+=	x += 3	x = x + 3	Try it »
-=	x -= 3	x = x - 3	Try it »
*=	x *= 3	x = x * 3	Try it »
/=	x /= 3	x = x / 3	Try it »
%=	x %= 3	x = x % 3	Try it »

PRIMER 5: ZBIR CIFARA TROCIFRENOG BROJA

```
#include <stdio.h>

int main() {
    printf("Unesi jedan trocifren broj\n");
    int broj;
    scanf("%d", &broj);

    int brojJ = broj%10;
```

```
int brojS = broj / 100;
int brojD = (broj - brojS*100) / 10;
int sum = brojJ + brojS + brojD;
printf("%d\n", sum);
return 0;
}
```

OBJASNJENJE

- scanf funkcija za input - mora da se naglasi za sta se vezuje uneti broj. Upisuje se na adresu gde stoji deklarirana promenljiva
- Napraviti paralelu sa input() funkcijom u pythonu
- Kako bi se uneo string? Realan broj? Isprobati.

PRIMER 6: MIROROVANI TROCIFRENI BROJ

```
#include <stdio.h>

int main() {
    printf("Unesi jedan trocifren broj\n");
    int broj;
    scanf("%d", &broj);

    int brojJ = broj%10;
    int brojS = broj / 100;
    int brojD = (broj - brojS*100) / 10;
    int noviBroj = brojJ*100 + brojS + brojD*10;
    printf("%d\n", noviBroj);
    return 0;
}
```

OBJASNJENJE

- Slicno kao u prethodnom primeru
- Novi broj se pravi po matematickim pravilima
- Broj koji se završava nulom u ovom slucaju daje mirorovan dvocifren broj