

Razvoj programa, Code::Blocks, struktura programa, printf, scanf,  
konverzioni karakteri

predavač: Nadežda Jakšić

# Programiranje programski jezik C

# Razvoj programa

- projektni zadatak
- projektovanje programa (algoritmi)
- pisanje programskog koda, izvorni kod, **source code**
  - sintaksne i semantičke greške
- kompajliranje, prevođenje izvornog koda u nule i jedinice; dobija se objektni kod; **nešto.OBJ**
- linkovanje, povezivanje; na izlazu se dobija izvršna datoteka; **nešto.EXE**
- izvršavanje programa
- testiranje
- održavanje i modifikacija programa

# Vrste programskih jezika

- mašinski jezik

**0011010110010110110110** naredba

- assembler – uvedene su mnemotehničke skraćenice za operacije i simboličke oznake podataka

**ADD a, b** naredba u assembleru

- viši programski jezici – jednoj naredbi odgovara više naredbi simboličkog jezika

**c = a + b;**

# Viši programski jezici

- **proceduralni** – zadaje se algoritam, tj. kompletan skup instrukcija za rešavanje problema

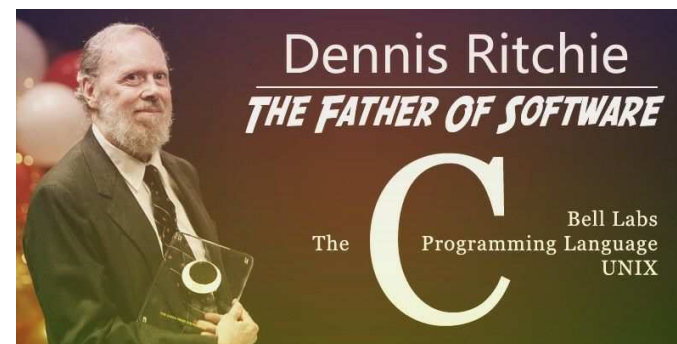
**Basic, Pascal, C**

- **deklarativni** – sastoje se od opisa problema na osnovu koga prevodilac deklarativnog jezika generiše rešenje
- posebnu klasu čine **objektno-orijentisani** jezici kod kojih su prisutni i proceduralni i deklarativni elementi

**C++, Java, Smalltalk, C#, Python, PHP**

# Programski jezik C

- razvio ga je **Dennis Ritchie** u Belovim laboratorijama za potrebe operativnog sistema UNIX
- jezik opšte namene iako je razvijen za sistemsko programiranje
- poseduje upravljačke strukture, ali podržava i manipulaciju bitovima, korišćenje procesorskih registara i pristup podacima pomoću adrese



# Struktura programa

```
#include <stdio.h> //pretprocesorske direktive
...
globalne promenljive
int main () //glavna funkcija u programu, može da bude samo jedna
{ //OS prvo poziva main funkciju; između zagrada { } je telo funkcije
  deklaracije i naredbe //na kraju svake naredbe obavezno ;
...
return 0; //ako program bude uspešno završen vratiće nulu
}
```

# Code::Blocks

- integrisano razvojno okruženje (IDE) otvorenog koda (free)
- C, C++, Fortran
- instalacija [www.codeblocks.org/downloads](http://www.codeblocks.org/downloads)

izabrati **Download the binary release**

izabrati **codeblocks -17.12 mingw-setup.exe**

# Komentari

- ako se pišu u jednom redu onda //
- ako se pišu u više redova onda počinju oznakom /\*, a završavaju se oznakom \*/
- **C** nije pozicioni jezik, forma je slobodna, ali je program nečitak (identifikatori i izrazi između znakova navoda se ne smeju razdvojiti u više redova)



# Izlazna konverzija printf

- ispisuju se podaci na ekranu
- ono što se prikazuje stavlja se u znake navoda

## za početak:

```
#include <stdio.h>
int main()
{
printf ("Danas je cetvrtak"); //ubaciti komentar, ispis u novom redu i
                             tabulaciju
printf ("Juce je bila sreda"); //ako želimo ispis znakova navoda onda \"
printf ("Sutra je petak");    //npr. printf (\"sutra je petak\")

    return 0;
}
```

- pišemo programe koji se izvršavaju u konzolnom okruženju

# Tipovi podataka

- **promenljiva** – ime memorijske lokacije u koju se smešta podatak koji može da menja vrednost u toku izvršavanja programa
- **konstanta** – ne menja vrednost u toku izvršavanja programa
- **tip** podatka je određen skupom mogućih vrednosti koje može da uzme podatak i skupom mogućih operacija koje mogu da se izvode nad tim podacima

# Osnovni tipovi podataka

- **char** – bilo koji karakter, tj. kôd tog karaktera; dužina jedan bajt
- **int** – celobrojni podatak koji po veličini najviše odgovara hardveru datog računara; obično dva bajta ili 4 bajta
- **float** – realan broj u jednostrukoj tačnosti; 4 bajta
- **double** – realan broj u dvostrukoj tačnosti; 8 bajtova

ako se ne navede nijedan tip, podrazumeva se **int**

# Osnovni tipovi podataka

svim navedenim tipovima mogu da se dodaju oznake:

- **signed** – označen broj (i pozitivan i negativan)
- **unsigned** – neoznačen broj (samo pozitivan)
- **long** – duži opseg brojeva
- **short** – kraći opseg brojeva

**C** propisuje samo minimalnu preciznost celobrojnih tipova, tako da tip **int** mora imati najmanju dužinu od 16 bitova, a tip **long** 32 bita

# Opsezi pojedinih tipova

primer računara na kom tip **short** zauzima 2 bajta, tip **int** 4 bajta, a tip **long** 8 bajtova

- **unsigned short:**  $0, 1, \dots, 2^{16} - 1 = 65535$
- **short:**  $-2^{15} = -32768, \dots, -1, 0, 1, \dots, 2^{15} - 1 = 32767$
- **unsigned:**  $0, 1, \dots, 2^{32} - 1 = 4294967295$
- **int:**  $-2^{31} = -2147483648, \dots, -1, 0, 1, \dots, 2^{31} - 1 = 2147483647$
- **unsigned long:**  $0, 1, \dots, 2^{64} - 1 = 18446744073709551615$
- **long:**  $-2^{63}, \dots, -1, 0, 1, \dots, 2^{63} - 1$

# Opsezi pojedinih tipova

primer računara na kom tip **int** zauzima dva bajta, a tip **long** 4 bajta

- **char** – znak od -128 do 127 tj. od  $-2^7$  do  $2^7-1$
- **signed char** – isto
- **unsigned char** – od 0 do 255, samo pozitivni
- **int** – 2 bajta, od  $-2^{15}$  do  $2^{15}-1$ , tj. od -32768 do 32767
- **long int** – 4 bajta, od  $-2^{31}$  do  $2^{31}-1$
- **unsigned int** – 2 bajta, od 0 do 65535, samo pozitivni

umesto **short int** može samo **short**

# Promenljive, deklaracija

- promenljivoj se obavezno dodeljuje tip i eventualno početna vrednost; ako se ne navede tip, podrazumeva se da je **int**
- **operator dodele** je znak jednakosti; izraz sa desne strane se dodeljuje izrazu sa leve strane znaka jednakosti; npr. **a=5**; ne može **5=a**

deklaracija: **tip identifikator = početna vrednost**

npr:

int a;

int a=5;

int a,b,c;

int a, b=8,c;

float precnik;

char m;

double porez;

identifikator – ime koje dodeljuje programer  
sadrži slova engleskog alfabeta, cifre i znak\_  
prvi znak mora da bude slovo  
pravi se razlika između malih i velikih slova  
ne mogu da budu službene reči

dve notacije za pisanje identifikatora:

porez\_na\_promet ili porezNaPromet

# printf – prikaz iz memorije

```
#include <stdio.h>
int main()
{
char a='n';
int b=5;
float pi=3.14159265358;
float PI=3.14159265358;
printf ("karakter je %c\n",a);
printf ("ceo broj je %d\n",b);
```

//%10d u polju širine deset poravnat sa desne strane, %-10d poravnat sa leve strane

```
printf ("realan broj u jednostrukoj tacnosti je%f\n",pi); //%.2f prikazuje broj sa dva decimalna mesta
```

```
printf ("realan broj u dvostrukoj tacnosti je %e\n",PI); //%.15.2f i %-15.2f
return 0;}
```

**eksponencijalna notacija** – za prikazivanje veoma velikih i veoma malih brojeva

**m x Ee** **m** je mantisa, **E** je osnova brojnog sistema, **e** je eksponent

**konverzioni karakteri za printf**

**%c** za karakter

**%d** za cele brojeve

**%f** za realne podatke

**%s** za string

**%u** za neoznačeni decimalni broj

**%l** prefiks koji se koristi uz %d, %u da bi se definisao dugačak – long broj



# Deklaracija konstanti

- daje se ime konstanti i onda se u programu koristi njeno ime, a ne vrednost (simbolička konstanta)
- naredba mora da se piše u jednom redu
- obično se identifikatori konstanti pišu velikim slovima

```
# define PI 3.1415
```

```
# define MIN 0
```

```
# define MAX 100
```

# Zadatak

```
#include <stdio.h>

int main()
{
    int a=5,b=8,c;
    c=a+b;
    printf ("Prvi broj je %d\n",a);
    printf ("Drugi broj je %d\n",b);

    printf ("Zbir ova dva broja je %d\n",c);

    return 0;
}
```

# Ulazna konverzija scanf

```
#include <stdio.h>
int main()
{
    int a,b,c;
    printf ("Unesite prvi broj\n");
    scanf ("%d", &a);
    printf ("Unesite drugi broj\n");
    scanf ("%d", &b);
    c=a+b;
    printf ("Zbir ova dva broja je %d\n",c);
    return 0;
}
```

# Konverziji karakteri za scanf

**%d** za int

**%i** za int

**%l** za long

**%u** za unsigned

**%f** za float

**%e** za float u eksponencijalnom obliku

**%lf** za double

**%s** za string

**%c** za char

# Obim i površina kruga

```
#include <stdio.h>
int main ()
{
float r,o,p;
printf ("Unesite duzinu poluprecnika\n");
scanf ("%f",&r);
o=2*r*3.1415;
p=r*r*3.1415;
printf ("Obim kruga je %f\n", o);
printf ("Povrsina kruga je %f\n", p);
return 0;
}
```

- deklarirati konstantu PI      **#define PI 3.1415**

# Kalkulator

```
# include <stdio.h>
int main ()
{
int prvi, drugi, zbir, razlika, proizvod;
float kolicnik;
printf ("Unesite dva broja\n");
scanf ("%d%d", &prvi,&drugi);
zbir=prvi+drugi;
razlika=prvi-drugi;
proizvod=prvi*drugi;
kolicnik=(float)prvi/drugi; //eksplicitna konverzija tipova
printf ("zbir ova dva broja je %d\n", zbir);
printf ("razlika ova dva broja je %d\n", razlika);
printf ("proizvod ova dva broja je %d\n", proizvod);
printf ("kolicnik ova dva broja je %f\n", kolicnik);
return 0;
}
```