




**PROGRAMIRANJE**  
Predavanje 02 – Varijable i tipovi podataka

Ishod  
učenja 1

1

**OSNOVNO IZVRŠAVANJE  
PROGRAMA**

Strana • 2



2

## Osnovni C++ program

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;

    return 0;
}
```

Strana \* 3



3

## Izvršavanje osnovnog C++ programa

Computer model 001

CPU



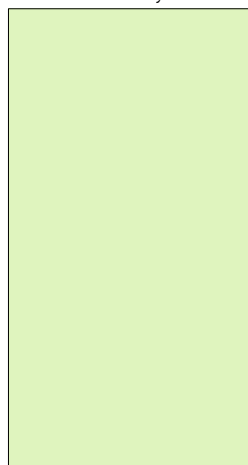
Program code

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;

    return 0;
}
```

Memory



Console (cout)



Keyboard (cin)



Strana \* 4



4

## Izvršavanje osnovnog C++ programa

Sljedeće za  
odraditi...



```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```

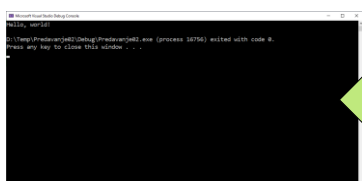
- Na početku, CPU detektira prvu liniju koju treba izvršiti
  - To je uvijek prva linija u mainu

Strana • 5



5

## Što se dešava u prvoj liniji?



- Tekst "Hello, world!" se šalje u konzolu
- Nakon toga, znak za prijelaz u novu liniju (new line character) se šalje u konzolu
  - Znak za prijelaz u novu liniju se označava kao `\n` ili `endl`

Strana • 6



6

## Izvršavanje osnovnog C++ programa

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```

Upravo sam  
odradio ovu  
liniju



- Prva linija je uspješno izvršena

Strana • 7



7

## Izvršavanje osnovnog C++ programa

Sljedeće...

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```



- Linija „return 0” vraća operacijskom sustavu uspjeh

Strana • 8



8

## Izvršavanje osnovnog C++ programa

Nema više ništa za izvršiti...



```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```

- Nema više ništa za odraditi pa program završava
  - (primijetite da u ovom zadatku nismo uopće koristili memoriju)

Strana • 9



9

## VARIJABLE I TIPOVI PODATAKA

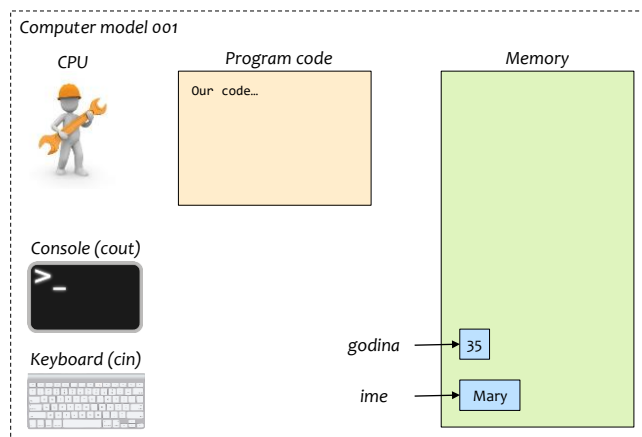
Strana • 10



10

## Varijable

- Varijabla je naziv za dio memorije koji može čuvati vrijednost



Strana • 11



11

## Varijable

- Svaka varijabla ima ime
- Svaka varijabla može čuvati točno jedan podatak koji se naziva vrijednost
  - Ako želimo čuvati više od jedne vrijednosti, moramo koristiti više varijabli
- Vrijednost varijable možemo mijenjati prema želji

Strana • 12

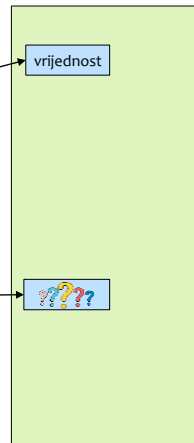


12

## Deklaracija varijable

- Deklaracija kreira varijablu
  - Rezervira memoriju za varijablu i (opcionalno) upisuje vrijednost u nju

**tip\_podataka naziv = vrijednost;**



- ili

**tip\_podataka naziv;**

Strana • 13



13

## Deklaracija varijable

- Pravila imenovanja varijabli:
  - Naziv može sadržavati sljedeće znakove:
    - Mala i velika slova engleske abecede: a... z i A... Z
    - Znamenke: 0... 9
    - Podvlaku: \_
  - Ne smije započeti sa znamenkom
- Ime je osjetljivo na velika i mala slova
  - Number, number, nuMber, NUMBER, ... su različiti nazivi

Strana • 14



14

## Tipovi podataka

- Svaka varijabla ima svoj tip podataka koji određuje operacije koje se mogu raditi s vrijednosti varijable
  - Npr. možemo pomnožiti dva broja, ali ne dva slova
- Postoje četiri ugrađena/primitivna/fundamentalna tipa:
  - `int` – za cijele brojeve: 5, -82, 0, 98, 152, ...
  - `double` – za decimalne brojeve: 3.14, 299793.0, -2.2, ...
  - `char` – za znakove: 'a', 'R', '!', '\n', ' ', '9', ...
  - `bool` – istina ili laž (true ili false)
- Koristit ćemo i posebni tip podataka za tekst:
  - `string` – "hello", "what a nice day!", "", ...

Strana • 15



15

## Tipovi podataka

- Varijabla nekog tipa podataka troši određenu količinu memorije

Tip podataka	Količina prostora u našem kompajleru
<code>bool</code>	1 bajt
<code>char</code>	1 bajt
<code>int</code>	4 bajta
<code>double</code>	8 bajtova
<code>string</code>	1 bajt za svaki znak

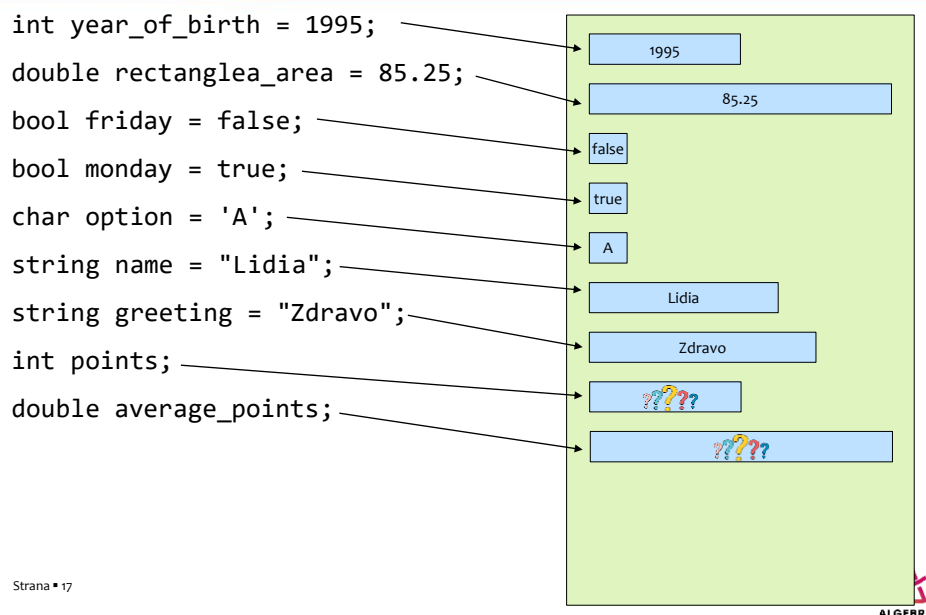
Strana • 16



16

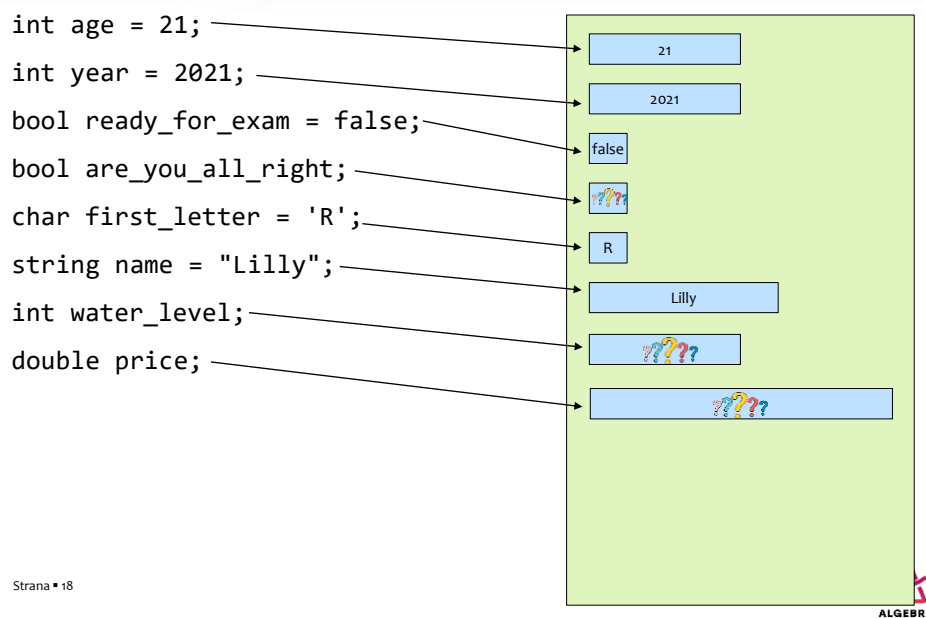


## Primjeri



17

## Primjeri



18

# PISANJE U VARIJABLU I ČITANJE IZ VARIJABLE

Strana • 19



19

## Pisanje u varijablu

- Vrijednost u varijablu možemo upisati korištenjem operatora dodjele vrijednosti operator= (*assignment operator*)
- Izračuna vrijednost na desnoj strani i kopira je u varijablu na lijevoj strani

```
varijabla = izracunata_vrijednost;
```

Strana • 20



20

## Čitanje iz varijable

- Vrijednost varijable čitamo tako da napišemo naziv varijable i što želimo napraviti s tom vrijednosti

```
cout << varijabla;
```

```
int varijabla_a = varijabla_b;
```

Strana • 21



21

## Primjeri

```
int a = 10;  
int b = 5;  
a = a + b + 2;  
cout << a << endl;  
cout << b << endl;
```

Strana • 22



22

## Složeniji primjer

Počijem  
tu...



```
#include <iostream>
using namespace std;

int main() {
    int total_points = 17;
    int new_points = 2;

    cout << "Before: " << total_points << endl;

    total_points = total_points + new_points;

    cout << "After: " << total_points << endl;

    return 0;
}
```

- Prva linija deklarira novu cjelobrojnu varijablu s vrijednosti 17
  - (rezervira se prostor od 4 bajta)

Strana • 23



23

## Složeniji primjer

Sljedeće...



```
#include <iostream>
using namespace std;

int main() {
    int total_points = 17;
    int new_points = 2;

    cout << "Before: " << total_points << endl;

    total_points = total_points + new_points;

    cout << "After: " << total_points << endl;

    return 0;
}
```

- Druga linija kreira još jednu cjelobrojnu varijablu s vrijednosti 2
  - Potrošili smo još 4 bajta

Strana • 24



24

## Složeniji primjer

Sljedeće...



```
#include <iostream>
using namespace std;

int main() {
    int total_points = 17;
    int new_points = 2;

    cout << "Before: " << total_points << endl;

    total_points = total_points + new_points;

    cout << "After: " << total_points << endl;

    return 0;
}
```

### ▪ Treća linija:

- Ispiše „Before:”
- Pročita vrijednost varijable i ispiše je
- Ispiše znak za prelazak u novi red

Strana \* 25



25

## Složeniji primjer

Sljedeće...



```
#include <iostream>
using namespace std;

int main() {
    int total_points = 17;
    int new_points = 2;

    cout << "Before: " << total_points << endl;

    total_points = total_points + new_points;

    cout << "After: " << total_points << endl;

    return 0;
}
```

- ### ▪ Četvrta linija čita vrijednosti obje varijable, zbraja ih i dobiveni zbroj upisuje u varijablu total\_points

Strana \* 26



26

## Složeniji primjer

Sljedeće...



```
#include <iostream>
using namespace std;

int main() {
    int total_points = 17;
    int new_points = 2;

    cout << "Before: " << total_points << endl;

    total_points = total_points + new_points;

    cout << "After: " << total_points << endl;

    return 0;
}
```

- Na kraju, ispisuje „After:” pa vrijednost varijable pa znak za novi red

Strana • 27



27

## Zadaci

1. Napišite program koji može čuvati tri vrijednosti: 19.2, 4.875 i 11.25. Izračunajte prosječnu vrijednost i spremite je u memoriju. Na kraju ispišite prosječnu vrijednost.
2. Napišite program koji može čuvati imena i starost dva studenta. Zatim ispišite sve podatke o prvom studentu pa o drugom.

Strana • 28



28

## Napomena

- Za stringove, moramo uključiti zaglavlje `<string>`

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string name = "Hannu";
    int age = 45;

    cout << name << " is of age " << age << endl;

    return 0;
}
```

Nemojmo zaboraviti

Dvostruki navodnici

Strana \* 29



29

## Literali

- Literal je vrijednost napisana izravno u programu

```
int a = 55;
```

Tip podataka    Naziv varijable    Literal

```
string name = "Annie";
```

Strana \* 30



30

# PISANJE U KONZOLU

Strana \* 31



31

## Pisanje u konzolu

- Konzolu predstavlja **cout** (*console output stream*)
- Za pisanje u konzolu se koristi operator umetanja << (insertion operator)

```
cout << "HELLOO!";
```

- Posebni znakovi se piše pomoću obrnute kose crte:

- \n = newline, enter, znak za prelazak u novi red
- \t = tab
- \" = dvostruki navodnik
- \\ = ovo ispisuje jednu obrnutu kosu crtu

- Što će ovo ispisati:

```
cout << "\tA\tB" << endl << "ONE\n" << "TWO" << endl;
```

Strana \* 32



32



# ČITANJE S TIPKOVNICE

Strana • 33



33

## Čitanje s tipkovnice

- Tipkovnicu predstavlja **cin** (*console input stream*)
- Osnovni tipovi se čitaju s operatorom vađenja >> (extraction operator)

```
int x;
cin >> x;
```

- Kako radi druga linija:

1. Program je blokiran dok korisnik ne pritisne tipku enter
2. Operator uzima sve utipkane znakove prije entera
3. Od njih napravi vrijednost osnovnog tipa
4. Upozorenje: enter ostaje u cin! ← Uff...

Strana • 34



34

## Čitanje s tipkovnice

- String vrijednosti se čitaju pomoću ugrađene funkcije `getline()`

```
string prezime;
getline(cin, prezime);
```

- Kako radi druga linija:
  1. Program je blokiran dok korisnik ne pritisne tipku enter
  2. Funkcija stavlja sve utipkane znakove prije enteru u varijablu
  3. Funkcija uklanja enter iz `cin`

Jako pristojno!



Strana • 35



35

## Problem

- Kad imamo sljedeću situaciju, dešava se problem:

```
string ime;
int godina;

cout << "Koliko imate godina: ";
cin >> godina;

cout << "Kako se zovete: ";
getline(cin, ime);

cout << "Pozdrav, " << ime << " imate " << godina << endl;
```

- Objasnite što izaziva problem!

Strana • 36



36

## Rješenje

- Koristimo `cin.ignore()` kako bismo ukloniti enter

```
string ime;
int godina;

cout << "Koliko imate godina: ";
cin >> godina;
cin.ignore();

cout << "Kako se zovete: ";
getline(cin, ime);

cout << "Pozdrav, " << ime << " imate " << godina << endl;
```

Strana • 37



37

## Strategija rješavanja

- Za puno programa možemo koristiti strategiju:
  1. Kreiramo sve potrebne varijable
  2. Pročitamo vrijednosti s tipkovnice
  3. Izračunamo što treba
  4. Ispišemo na konzolu
- Ponekad ćemo neke korake spojiti ili preskočiti

Strana • 38



38

## Zadaci

3. Napišite program koji učitava temperaturu u stupnjevima Fahrenheita i ispisuje temperaturu u stupnjevima Celzijusa. Formula je:  $\text{TempUC} = (\text{TempUF} - 32) * 5/9$ 
  - Npr. 30 C je 86 F
4. Napišite program koji učitava radijus kružnice te ispisuje njezinu površinu i opseg ( $P=r^2*PI$ ,  $O=2r*PI$ ).
5. Učitajte od korisnika godinu rođenja, ime, prezime, je li redovan student ili ne, te prvo slovo naziva smjera. Ispišite učitane vrijednosti.

Strana \* 39



39

## Zadaci

6. Formula za skalarni produkt 3D vektora je:  $\mathbf{AB} = a_x b_x + a_y b_y + a_z b_z$ , gdje je  $\mathbf{A} = a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}$ ,  $\mathbf{B} = b_x \mathbf{i} + b_y \mathbf{j} + b_z \mathbf{k}$ . Učitajte dva vektora na način da učitajte njihove komponente ( $a_x, a_y, \dots$ ) i ispišite njihov skalarni produkt.
7. Strijelac gađa lukom i strijelom i pogađa sljedeće vrijednosti na meti: 2, 4, 4, 4, 5, 5, 7, 9. Izračunajte i ispišite prosječnu pogođenu vrijednost i standardnu devijaciju.
  - Napomena: zbog jednostavnosti računanja, svi brojevi tipovi podataka neka budu double
  - Napomena: uključite zaglavlje **cmath** i koristite ugrađene funkcije `pow(broj, potencija)` i `sqrt(broj)`.

Strana \* 40



40

## Zadaci za sljedećih 7 dana

- Prije sljedećeg predavanja je obavezno:
  1. Pročitati sljedeća poglavlja iz knjige *Demistificirani C++*:
    - 2.7 Upravljanje slijedom izvođenja programa
    - 3.3 Operator pridruživanja
    - 3.4 Tipovi podataka i operatori
    - 4.1 Blokovi naredbi
    - 4.2 Naredbe za grananje toka
  2. Pogledati sljedeće:
    - W03-1 Logical expressions
      - <https://youtu.be/zefodJMIY7M>
    - W03-2 if, else if, else statements
      - <https://youtu.be/yJUuGqV9lFM>
    - W03-3 switch statement
      - <https://youtu.be/MEEjUho-EZE>

Strana • 41

