

# Osnove programskega jezika C++

1

**OSNOVE PROGRAMIRANJA V C++  
UPORABA PROGRAMA CODE::BLOCKS**

## 2. Prvi program v C++

2

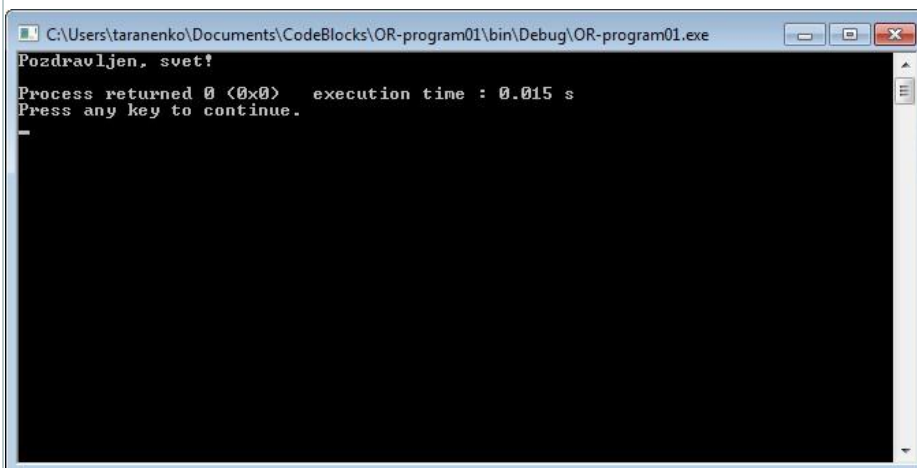
## Prvi program v C++

3

```
/*  
    Moj prvi program.  
    Avtor: Andrej Taranenko.  
*/  
  
#include <iostream>  
using namespace std;  
  
int main() {  
    // Na ekran izpišimo sporočilo  
    cout << "Pozdravljen, svet!" << endl;  
  
    return 0;  
}
```

## Prvi program v C++ ... rezultat

4



```
C:\Users\taranenko\Documents\CodeBlocks\OR-program01\bin\Debug\OR-program01.exe  
Pozdravljen, svet!  
Process returned 0 (0x0)   execution time : 0.015 s  
Press any key to continue.  
_
```

## Prvi program v C++

5

```

/*
    Moj prvi program.
    Avtor: Andrej Taranenko.
*/

#include <iostream>
using namespace std;

int main() {
    // Na ekran izpišimo sporočilo
    cout << "Pozdravljen, svet!" << endl;

    return 0;
}

```

## Prvi program v C++

6

The diagram shows the same code as slide 5, but with annotations explaining its parts:

- Annotation 1:** Points to the multi-line comment block (`/* ... */`) with the text "več vrstični komentar".
- Annotation 2:** Points to the `#include <iostream>` line with the text "vključujemo zunanje vire".
- Annotation 3:** Points to the `using namespace std;` line with the text "lažji dostop do nekaterih objektov".
- Annotation 4:** Points to the `int main() {` line with the text "funkcija z imenom *main()*".
- Annotation 5:** Points to the `return 0;` line with the text "tukaj se program prične izvajati".
- Annotation 6:** Points to the closing brace of the `main()` function (`}`) with the text "konča izvajanje funkcije *main()*, s tem zaključi program".

## Programi v C++

7

- Preprost program v C++ se prične tako:

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

int main() {
```

- in zaključi na naslednji način:

```
    return 0;
}
```

## Zapis programa v C++

8

- prevajalnik strogo loči med malimi in velikimi črkami
- sprejme se poljubna oblika zamikov in praznin
- program zapišemo tako, da je enostavno čitljiv
  - { postavimo takoj za stavek, h kateremu pripada
  - } postavimo v svojo vrstico in poravnamo z začetkom stavka, ki ga zaključuje
  - v eno vrstico zapišemo en stavek
  - stavke, ki “spadajo skupaj” zamaknemo za enako št. znakov
- stavki (ne nujno vrstice) se zaključijo s podpičjem

## Zapis programa v C++

9

`#include <iostream>`

- ✦ pove prevajalniku, kje išče informacije, ki jih uporabljamo v programu
- ✦ s tem vključujemo knjižnico `iostream`
- ✦ `iostream` vsebuje definicije objektov `cout` in `cin`

`using namespace std;`

- ✦ v katerem imenskem prostoru (predstavljajmo si skupino) naj išče informacije, če jih ne najde prej

## Zapis programa v C++

10

`int main() {`

- ✦ je začetek glavne funkcije
- ✦ program se prične izvajati tukaj

`return 0;`

`}`

- ✦ zaključek funkcije `main()`
- ✦ sporoči operacijskemu sistemu, da pri izvajanju ni bilo napak (vračamo vrednost nič – o napak)

## Zapis programa v C++

11

/\*

Moj prvi program.

Avtor: Andrej Taranenko.

\*/

- ✦ je komentar, ki zavzema več vrstic
- ✦ prične se s simbolom /\*
- ✦ zaključí se s simbolom \*/
- ✦ vse vmes šteje za komentar
- ✦ komentarji niso del izvedljivega programa
- ✦ komentarji so namenjeni tistim, ki delajo z izvornim programom

## Zapis programa v C++

12

// Na ekran izpišimo sporočilo

- ✦ je enovrstični komentar
- ✦ prične se s simbolom //
- ✦ zaključí se s skokom v novo vrstico
- ✦ vse od simbola // naprej in do konca vrstice šteje za komentar
- ✦ običajno z njimi opišemo stavek oz. skupino stavkov, ki sledijo

## Zapis programa v C++

13

```
cout << "Pozdravljen, svet!" << endl;
```

- ✦ z objektom cout prikažemo informacijo na ekranu (konzoli)
- ✦ operator << lahko beremo kot “pošlji” v smeri puščic
- ✦ tekst v C++ zapišemo znotraj dvojnih narekovajev
- ✦ endl predstavlja skok v novo vrstico
- ✦ podpičje zaključuje stavek, ki se je pričel s cout


## Prevajanje, povezovanje, zagon...

14

- izvorni program oz. izvorno kodo pišemo v navadnem urejevalniku besedil
- shranjen je v .cpp datoteki – navadna tekstovna datoteka
- prevajalnik pretvori izvorno kodo v objektno kodo
- povezovalnik združi vse objektne kode v izvedljivi program

## Prevajanje, povezovanje, zagon...

15

- kodo prevedemo in povežemo v izvedljiv program
  - pritisnemo ikono za ustvarjanje izvedljivega programa 
  - pritisnemo kombinacijo tipk **Ctrl** in **F9**
  - izberemo meni Build / Build
- odpravimo napake, na katere opozori prevajalnik
  - po odpravljanju napak moramo program znova prevesti iz zgraditi

slovarček:

- build ... zgradi
  - prevede program in zgradi izvedljivi program

## Prevajanje, povezovanje, zagon...

16

- izvedljivi program zaženemo
  - pritisnemo ikono za zagon programa 
  - pritisnemo kombinacijo tipk **Ctrl** in **F10**
  - izberemo meni Build / Run
- oba koraka lahko združimo z Build and Run
  - pritisnemo ikono za zgradi in zaženi 
  - pritisnemo tipko **F9**
  - izberemo meni Build / Build and Run

slovarček:

- run ... zaženi
  - zažene izvedljivi program
- build and run
  - prevede program in tvori izvedljivi program, ki ga nato še zažene



## Napake, napake, napake ...

17

- **sintaktične napake**
  - so napake, ki kršijo pravila sintakse jezika C++
  - prepozna jih prevajalnik
  - prevajalnik jih opiše (pove, kaj ga moti) in pogosto prikaže, kje v izvorni kodi je napaka
- **logične napake**
  - težko odkriti
  - so napake v algoritmu reševanja
  - računalnik jih ne prepozna kot napake
- **napake v času izvajanja programa**
  - napake, do katerih pride, ko izvajamo program (npr. napačen tip podatkov)

## 3. Osnovni pojmi

18

## Osnovni pojmi

19

1. žetoni
  1. ključne besede
  2. identifikatorji
  3. literali
  4. operatorji
  5. ločila
2. znaki programskega jezika C++
3. nevidni znaki
4. komentarji

## 3. 1. Ključne besede

20

**BESEDE V C++, KI SO IZKLJUČNO  
REZERVIRANE, IMAJO STROGO DOLOČEN  
POMEN IN JIH NE SMEMO UPORABLJATI NA  
NOBEN DRUG NAČIN.**

## Ključne besede v C++

21

asm	auto	bool	break	case
catch	char	class	const	
const_cast	continue	default	delete	do
double	dynamic_cast	else	enum	explicit
export	extern	false	float	for
friend	goto	if	inline	int
long	mutable	namespace	new	
operator	private	protected	public	register
reinterpret_cast		return	short	signed
sizeof	static	static_cast	struct	switch
template	this	throw	true	try
typedef	typeid	typename	union	
unsigned	using	virtual	void	volatile
wchar_t	while			

Osnove računalništva in informatike, predavanja

dr. Andrej Taranenko

## Ključne besede v C++

22

asm	auto	bool	break	case
catch	char	class	const	
const_cast	continue	default	delete	do
double	dynamic_cast	else	enum	explicit
export	extern	false	float	for
friend	goto	if	inline	int
long	mutable	namespace	new	
operator	private	protected	public	register
reinterpret_cast		return	short	signed
sizeof	static	static_cast	struct	switch
template	this	throw	true	try
typedef	typeid	typename	union	
unsigned	using	virtual	void	volatile
wchar_t	while			

Osnove računalništva in informatike, predavanja

dr. Andrej Taranenko

## Ključne besede v C++

23

asm	auto	bool	break	case
catch	char	class	const	
const_cast	continue	default	delete	do
double	dynamic_cast	else	enum	explicit
export	extern	false	float	for
friend	goto	if	inline	int
long	mutable	namespace	new	
operator	private	protected	public	register
reinterpret_cast	static	return	short	signed
sizeof	this	static_cast	struct	switch
template	typeid	throw	true	try
typedef	using	typename	union	
unsigned	while	virtual	void	volatile
wchar_t				

Osnove računalništva in informatike, predavanja

dr. Andrej Taranenko

## 3. 2. Identifikatorji

24

**BESEDE V JEZIKU C++, KI JIM MI SAMI  
DOLOČIMO POMEN.**

Osnove računalništva in informatike, predavanja

dr. Andrej Taranenko

## Identifikatorji

25

- zaporedje črk, števk in podčrtajev
  - ne sme se pričeti s števk
  - črke angleške abecede!!!
- ločimo med malimi in velikimi črkami
- običajno neka smiselna imena
  - eno ali dvoznakovne identifikatorje le, ko je očitno
  - lahko poljubno dolgi
    - ✖ nekateri sistemi ločijo le prvih 31 znakov

## Identifikatorji - primeri

26

veljavni	neveljavni
stevec	for
i	3a
cout	vsota števil
stevilo1	števil01
stevilo_1	stevilo 1

## 3. 3. Literali

27

SO KONSTANTNE VREDNOSTI, KI JIH  
PREVAJALNIK PREPOZNA – OBRAVNAVA JIH  
KOT EDEN OD OSNOVNIH PODATKOVNIH  
TIPOV

Osnove računalništva in informatike, predavanja

dr. Andrej Taranenko

## Literali

28

- so konstantne vrednosti
- prevajalnik jih prepozna

celoštevilski	znakovni	nizi	števila s plavajočo vejico
5	'5'	"besedilo"	0.123
05	'A'	"še en tekst"	7.
0x5	'a'		7.0
	'\n'		0.7E1
	'\t'		700E-2
3	'3'	"3"	3.0

Osnove računalništva in informatike, predavanja

dr. Andrej Taranenko

## 3. 4. Operatorji in ločila

29

## Operatorji in ločila

30

- operatorji, ločila, komentarji in nevidni znaki ločujejo elemente jezika C++
- operatorji
  - uporabljamo jih v izrazih
  - pomen imajo šele z ustreznimi argumenti
  - isti simboli lahko imajo različne pomene
- ločila
  - uporabljamo za strukturiranje elementov programa

## Operatorji in ločila

31

### Operatorji v C++ (nekateri)

<code>+</code> <code>-</code> <code>*</code> <code>/</code> <code>%</code>	aritmetični operatorji
<code>-&gt;</code> <code>-&gt;*</code>	operator za kazalec in kazalec na člana
<code>&amp;&amp;</code> <code>  </code> <code>!</code>	logični operatorji
<code>=</code> <code>+=</code> <code>*=</code>	operatorji prirejanja
<code>::</code>	operator dosega
<code>new</code> <code>delete</code>	operatorja prostega pomnilnika

## Operatorji in ločila

32

### Ločila v C++

<code>(</code> <code>)</code>	oklepaji
<code>{</code> <code>}</code>	zaviti oklepaji
<code>,</code>	vejica
<code>;</code>	podpičje



