

# Увод в програмирането

3: Основни елементи на езика (част 1)

доц. Атанас Семерджиев

## Съдържание

- Коментари
- Идентификатори и ключови думи
- Основни типове
- Литерали и константи

```
/* Това е класическият тип коментар от C */
```


```
int x;
```

```
/* Той  
   може да бъде  
   на няколко реда... */
```

```
int y;
```

```
int /* ...и да се появи навсякъде */ z;
```

Въпреки, че  
е възможно,  
НЕ правете  
това в своя  
код!!!



3

```
// C++-style comments
```

```
// Тези коментари завършват в края на реда
```

```
int x;
```

```
// Ако искате да са на няколко реда  
// ще трябва да коментирате всеки от  
// от тях поотделно.
```

```
int y;
```

4

```
// По правило коментарите (независимо кой  
// стил използвате) се поставят или над това,  
// което коментирате...
```

```
int x;
```

```
int z; // ...или вдясно от него.
```

5

## Идентификатори

Идентификаторът е последователност от един или повече от следните:

- Букви (малки и големи)
- Цифри
- Долна черта

Идентификаторът не може да започва с цифра.

Въпреки, че е възможно в своите програми да изберете идентификатори започващи с долна черта, това е лоша практика.

6

```
// Дадените по-долу идентификатори са валидни:
int z;
int asdushdfh766_2222;
int arnold;
int _a_b_c_;

// Следващите обаче НЕ СА:
int 123;           // Започва с число
int a-b+c;         // Съдържа непозволени символи
int Two Words;     // Съдържа интервал
                  // (отново непозволен символ)
```

7

```
// ВНИМАНИЕ!!!
// С и С++ правят разлика между главни и малки
// букви!!!
int Age = 10;
int age = 20; // Това е различна променлива!

cout << "Age("
      << Age
      << "), age("
      << age
      << ")\n"; // извежда: Age(10), age(20)
```

8

```
// Конвенции за именуване на променливите
```

```
// Много, много лош стил:
```

```
int abfd_77 = x - y998; // 0_o
```

```
// Много по-ясно:
```

```
int MonthlyProfit = Income - Expenses;
```

```
// Унгарска конвенция:
```

```
int nMonthlyProfit = fpIncome - fpExpenses;
```

9

## Конвенции за именуване

Няколко прости правила:

- Името на една променлива трябва ясно да показва какво се съдържа в нея.
- Името на една функция трябва ясно да показва какво извършва тя.
- Не избирайте прекалено кратки идентификатори просто защото ви мързи да ги изписвате. Вместо това ги изберете веднъж и оставете средата да изписва вместо вас.

10

```
// Някои популярни конвенции за именуване  
  
int someverylongandstrangename; // Ko? He!  
  
int monthlyIncome; // Camel case  
  
int MonthlyIncome; // Pascal Case  
  
int monthly_income; // Долни черти  
  
int nMonthlyIncome; // Унгарска конвенция
```

11

## Ключови думи

Когато избирате идентификаторите за своите програми, не можете да използвате запазените (ключови) думи в езика.

Такива са например:

```
asm, bool, break, case, catch, char, class, const, continue, default,  
delete, do, double, else, enum, false, float, for, goto, if, int,  
long, new, operator, return, short, sizeof, switch, typedef, wchar_t,  
while
```

12

## Основни типове

Тип	Описание	Възможна ширина
bool	Булев (истина/лъжа)	8 бита
char	ASCII символ	8 бита
wchar_t	Multibyte символ	16 или 32 бита
short int (short)	Цяло число със знак	Поне 16 бита
int	Цяло число със знак	Поне 16 бита
long int (long)	Цяло число със знак	Поне 32 бита
long long int (long long)	Цяло число със знак	Поне 64 бита
float	Реално число	Поне 32 бита
double	Реално число	Поне 64 бита
long double	Реално число	Поне 64 бита

13

```
// Декларация на променлива
```

```
// Декларация на една променлива  
int Week;
```

```
// Декларация на няколко променливи  
int Hour, Minute, Second;
```

```
// Възрастта не може да бъде отрицателно число  
unsigned int Age = 10;
```

```
// signed се подразбира и може да се изпусне!  
signed int Balance;
```

14

```
// В C++ променливите могат да се декларират  
// на произволно място в кода  
  
int Week;  
  
cout << Week << endl; // Какво ще се случи тук?  
  
unsigned int Age = 10;  
  
cout << Age << endl;
```

15

## Декларация на променлива

char, short, long и int могат да бъдат със или без знак (signed или unsigned)

signed се подразбира и може да се изпусне

char прави изключение!

- char е отделен тип представящ един символ.
- Ако се използва в качеството на 8-битово число, задължително трябва да се укаже signed или unsigned.

16



## Декларация на променлива

Тип	Еквивалентен тип
short	short int
long	long int
long long	long long int
unsigned	unsigned int
signed	signed int

17

## Граници (цели числа)

Брой бита	Възможни стойности
8 бита	signed: -128 to 127 unsigned: 0 to 255
16 бита	signed: -32768 to 32767 unsigned: 0 to 65535
32 бита	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295

18

// Какво се случва при прехвърляне на границите?

```
short A = 32767;
```

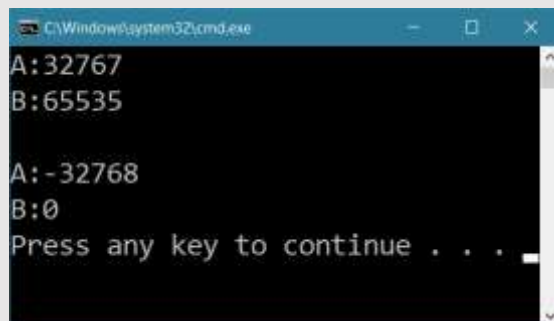
```
unsigned short B = 65535;
```

```
cout << "A:" << A << endl  
      << "B:" << B << endl  
      << endl;
```

```
A = A + 1;
```

```
B = B + 1;
```

```
cout << "A:" << A << endl  
      << "B:" << B << endl;
```



19

// Инициализация на променлива

// Декларация на променлива + инициализация

```
int Year = 2000;
```

// За същата цел можем да използваме скоби.

// Двата израза за еквивалентни

```
int Month(11);
```

// Тази променлива не е инициализирана

// Съдържанието ѝ е неопределено

```
int Day;
```

20

```
// ВНИМАНИЕ!!!  
// Опасно е да оставяте променливите си  
// неинициализирани и обикновено  
// НЕ БИВА да го правите.  
  
int Income = 1000; // ОК  
int Expenses; // Неопределено  
  
... // Някакъв код, който не променя Expenses  
  
// Какво ще съдържа Total???  
int Total = Income - Expenses;
```

21

## Литерали

Литералите са константи.

Основни типове:

- Булеви
- Цели числа
- Реални числа
- Символи
- Символни низове (стрингове)

22

## Литерали – Булеви

Има точно два булеви литерала:

- `true` – истина
- `false` – лъжа

```
bool TheEarthIsFlat = false;  
  
bool SomeBirdsFly = true;
```

23

// Литерали - цели числа

```
int A = 100; // Цяло число без знак
```

```
int B = -54; // Отрицателно число
```

```
int C = 023; // осмичен запис  
           // (десетично 19)
```

```
int D = 0xFF; // шестнадесетичен запис  
             // (десетично 255)
```

24

```
// Указване на типа
```

```
100 // int
```

```
100u // unsigned int
```

```
100l // long int
```

```
100ul // unsigned long int
```

25

```
// Литерали – реални числа
```

```
12.34 // 12,34
```

```
12e34 // 12 * 1034
```

```
12e-34 // 12 * 10-34
```

```
.5 // 0,5
```

```
5. // 5,0
```

26

```
// Литерали – символи
```

```
char Symbol1 = 'a'; // Малка латинска буква 'a'
```

```
char Symbol2 = '\n'; // Нов ред
```

```
char Symbol3 = '\x61'; // Код на символ в hex  
// Отново буквата 'a'
```

```
char Symbol4 = '\141'; // Код на символ в oct  
// Отново буквата 'a'
```

27

```
char Symbol5 = '''; // НЕВАЛИДНО!!!
```

```
char Symbol6 = '\''; // Коректно
```

```
char Symbol7 = '"'; // С кавичката  
// няма проблем
```

```
char Symbol8 = '\\'; // Обратна  
// наклонена черта
```

28

## Escape Codes

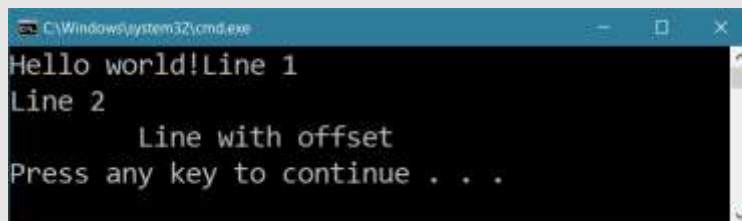
Тип	Описание
\n	Нов ред
\r	Carriage return
\b	Връщане на символ (backspace)
\t	Табулация
\a	Аларма/Сигнал
\'	Апостроф
\"	Кавичка
\?	Питанка
\\	Обратна наклонена черта

29

// Литерали – символни низове

```
cout << "Hello world!";
```

```
cout << "Line 1\nLine 2\n\tLine with offset\n";
```



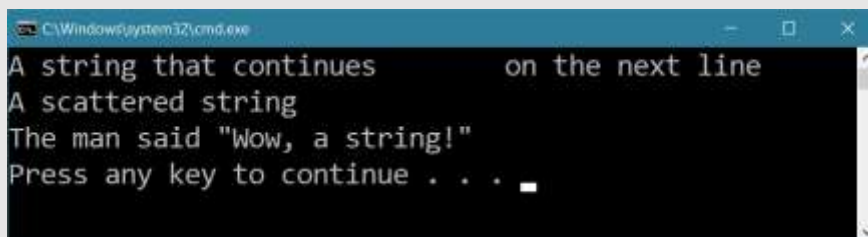
```
C:\Windows\system32\cmd.exe
Hello world!Line 1
Line 2
    Line with offset
Press any key to continue . . .
```

30

```
cout << "A string that continues \
      on the next line\n";

cout << "A " "scattered "
      "string\n";

cout << "The man said \"Wow, a string!\"\n";
```



```
C:\Windows\system32\cmd.exe
A string that continues      on the next line
A scattered string
The man said "Wow, a string!"
Press any key to continue . . .
```

31

```
// Променливи-константи

// Декларация на константа
const float PI = 3.14;

float Radius;
cout << "Enter Radius: ";
cin >> Radius;

float Area = PI * Radius * Radius;

PI = 3.1415926; // Грешка!!!
```

32



```
// Често срещана конвенция е имената на  
// константите да са ALL-UPPERCASE  
  
const float PI = 3.14;  
  
const int MAX_SIZE = 256;  
  
const unsigned int ERROR_CODE = 0xFFFFFFFF;
```

33