Redactare: Daniel Mitran
Tehnoredactare: Iuliana Ene
Pregătire de tipar: Marius Badea
Design copertă: Mirona Pintilie
Sursă foto interior: © Zdenek Sasek | Dreamstime.com

Editura Paralela 45

Descrierea CIP a Bibliotecii Naționale a României
GRECU, SILVIA
Memorator de informatică : limbajul C++ : clasele 9-12 : / Silvia Grecu, Lucia Miron, Mirela Țibu. - Pitești : Paralela 45, 2019
ISBN 978-973-47-3116-9

I. Miron, Lucia
II. Țibu, Mirela-Anca

004

SILVIA GRECU LUCIA MIRON
MIRELA ȚIBU

MEMORATOR DE INFORMATICĂ

LIMBAJUL C++

Clasele 9-12

Editura Paralela 45

Cuprins

I. ELEMENTE DE BAZĂ ALE LIMBAJULUI C++

Structura unui program C++

```
#include <iostream>
   //includerea altor librării necesare în
   program
using namespace std;
   //declarare variabile globale
   //declarare funcţii utilizator
int main()
{
   //declarare variabile locale
   instrucţiuni
   return 0;
}
```

Citirea valorilor variabilelor de la tastatură / afişarea valorilor expresiilor pe ecran

```
cin>>var1>>var2>>...>>varn;
cout<<exp1<<' '<<exp2<<' '<<...<<expn;
```

Atribuirea

```
variabila = expresie;
variabila <op> = expresie;
   //unde op ∈ {+,-,*,/,%,>>,<<,&,|,^}, este
   echivalentă cu
   //variabila = variabila <op> (expresie);
```

Instrucţiunea *if*

```
a) if (expresie)
        instrucţiune_A;
   else
        instrucţiune_B;
b) if (expresie)
        instrucţiune_A;
```

Instrucţiunea *switch*

```
switch (expresie) {
      case constantă_1: instrucţiuni_1
                        break;
      case constantă_2: instrucţiuni_2
                        break;
      ……
      case constantă_n: instrucţiuni_n
                        break;
     default: instrucţiuni
}
```


Instrucţiunea *while*

```
while (expresie)
  {
    Instrucţiuni
  }
```

Instrucţiunea *do-while*

```
do {
    Instrucţiuni
  } while (expresie);
```

Instrucţiunea *for*

```
for (expresie1; expresie2; expresie3)
{
   Instrucţiuni
}
```

`expresie1`: expresie de **iniţializare**;

`expresie2`: expresie de **test**;

`expresie3`: expresie de **continuare**.

Tipuri de date simple

Tip variabilă:
short int
Nr. octeți (bytes): 2 Domeniul de valori: $[-2^{15}, 2^{15} - 1]$
unsigned short int
Nr. octeți (bytes): 2 Domeniul de valori: $[0, 2^{16} - 1]$
int = long int
Nr. octeți (bytes): 4 Domeniul de valori: $[-2^{31}, 2^{31} - 1]$
unsigned int = unsigned long int
Nr. octeți (bytes): 4 Domeniul de valori: $[0, 2^{32} - 1]$
long long int
Nr. octeți (bytes): 8 Domeniul de valori: $[-2^{63}, 2^{63} - 1]$
unsigned long long int
Nr. octeți (bytes): 8 Domeniul de valori: $[0, 2^{64} - 1]$
Operatori:
Aritmetici: +, −, *, /, % Pe biți: ≫, ≪, &, |, ^, ~ Relaționali: <, <=, >=, > De egalitate: ==, !=
Funcții specifice:
<cmath.h> sqrt(x), x ≥ 0, pentru $\sqrt{x}$ abs(x) pentru |x| pow(a,b) pentru a^b

Tip variabilă:
float
Nr. octeți (bytes): 4 Domeniul de valori: $[-3.2 \cdot 10^{38}, 3.2 \cdot 10^{38}]$
double
Nr. octeți (bytes): 8 Domeniul de valori: $[-1.7 \cdot 10^{308}, 1.7 \cdot 10^{388}]$
Operatori:
Aritmetici: +, −, *, / Relaționali: <, <=, >=, > De egalitate: ==, !=
Funcții specifice:
<cmath.h> sqrt(x), x ≥ 0, pentru $\sqrt{x}$ abs(x) pentru |x| pow(a,b) pentru a^b

Tip variabilă:
char
Nr. octeți (bytes): 1 Domeniul de valori: $[-3.2 \cdot 10^{38}, 3.2 \cdot 10^{38}]$
unsigned char
Nr. octeți (bytes): 1 Domeniul de valori:
Operatori:
Aritmetici: +, −, *, /, % Relaționali: <, <=, >=, > De egalitate: ==, !=

Funcţii specifice:

islower(car) = { 1, dacă car e literă mică ; 0, altfel }

isupper(car) = { 1, car e literă majusculă ; 0, altfel }

tolower(car) = *car* în minusculă = car+'a'–'A'

toupper(car) = *car* în majusculă = car–'a'+'A'

Operatori

1. Prioritate maximă:

Operatori	*Semnificaţie*	*Asociativitate*
( ), [ ], → .	Apel de funcţie Expresie cu indici Selectori de membru la structuri	St-Dr

2. Operatori unari:

Operatori	*Semnificaţie*	*Asociativitate*
!	Negare logică	Dr-St
~, +, -	Negare bit cu bit (complementare cu 1) Plus şi minus unari	
++, --,	Incrementare/decrementare (pre şi post)	
&, *	Obţinerea adresei/indirectare	
sizeof(tip)	Dimensiune operand (în octeţi)	
(cast)	Conversie explicită de tip – cast	

3. Operatori aritmetici multiplicativi:

Operatori	*Semnificaţie*	*Asociativitate*
*, /, %	Înmulţire/împărţire/restul împărţirii întregi	St-Dr

4. Adunare, scădere:

Operatori	*Semnificaţie*	*Asociativitate*
+, -	Plus şi minus binari	St-Dr

5. Deplasări:

Operatori	*Semnificaţie*	*Asociativitate*
<<, >>	Deplasare stânga/dreapta pe biţi	St-Dr

6. Relaţionali:

Operatori	*Semnificaţie*	*Asociativitate*
<, <=, >, >=	Mai mic/mai mic sau egal/Mai mare/ mai mare sau egal	St-Dr

7. Egalitate:

Operatori	*Semnificaţie*	*Asociativitate*
==, !=	Egal/Diferit	St-Dr