

EDITURA PARALELA 45

colecția
COMPACT

Redactare: Daniel Mitran
Tehnoredactare: Iuliana Ene
Pregătire de tipar: Marius Badea
Design copertă: Mirona Pintilie

Sursă foto interior: © Zdenek Sasek | Dreamstime.com

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

Copyright © Editura Paralela 45, 2019

Prezenta lucrare folosește denumiri ce constituie mărci înregistrate,
iar conținutul este protejat de legislația privind dreptul de
proprietate intelectuală.

www.edituraparelela45.ro

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++	7
II. Algoritmi elementari.....	17
III. Fișiere text.....	33
IV. Tablouri unidimensionale (Vectori).....	35
V. Tablouri bidimensionale (Matrice).....	43
VI. Șiruri de caractere	47
VII. Structuri de date neomogene	53
VIII. Subprograme (Funcții)	57
IX. Funcții recursive.....	63
X. Elemente de combinatorică.....	67
XI. Elemente de teoria grafurilor.....	73

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 funcții utilizator
int main()
{
    //declarare variabile locale
    instrucțiuni
    return 0;
}
```

Citirea valorilor variabilelor de la tastatură / afișarea 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);
```

Instrucțiunea *if*

```
a) if (expresie)  
    instrucțiune_A;  
    else  
    instrucțiune_B;  
b) if (expresie)  
    instrucțiune_A;
```

Instrucțiunea *switch*

```
switch (expresie) {  
    case constantă_1: instrucțiuni_1  
        break;  
    case constantă_2: instrucțiuni_2  
        break;  
    .....  
    case constantă_n: instrucțiuni_n  
        break;  
    default: instrucțiuni  
}
```

Instrucțiunea *while*

```
while (expresie)
{
    Instrucțiuni
}
```

Instrucțiunea *do-while*

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

Instrucțiunea *for*

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

expresie1: expresie de inițializare;

expresie2: expresie de test;

expresie3: expresie de continuare.

Tipuri de date simple

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