

Dan GOTA  
Alexandra FANCA  
Adela POP  
Ovidiu STAN  
Honoriu VALEAN

---

# Programarea orientata pe obiecte în limbajul C#

---

**Toate drepturile rezervate autorilor & Editurii Risoprint**

*Editura RISOPRINT este recunoscută de C.N.C.S.  
(Consiliul Național al Cercetării Științifice).*  
*www.risoprint.ro*                      *www.cncs-uefiscdi.ro*



Opiniile exprimate în această carte aparțin autorilor și nu reprezintă punctul de vedere al Editurii Risoprint. Autorii își asumă întreaga responsabilitate pentru forma și conținutul cărții și se obligă să respecte toate legile privind drepturile de autor.

Toate drepturile rezervate. Tipărit în România. Nicio parte din această lucrare nu poate fi reprodusă sub nicio formă, prin niciun mijloc mecanic sau electronic, sau stocată într-o bază de date fără acordul prealabil, în scris, al autorilor.

All rights reserved. Printed in Romania. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the author.

ISBN: 978-973-53-2917-4

**Programarea orientata pe obiecte  
în limbajul C#**

**Autori:**

**Dan GOTA, Alexandra FANCA  
Adela POP, Ovidiu STAN, Honoriu VALEAN**

**Director editură: GHEORGHE POP**

## Cuprins

Capitolul 0: Introducere in C# .....	3
Capitolul 1: Setarea mediului de lucru C# in Visual Studio .....	3
I.    Componente de bază implicate în procesul de configurare a mediului în C# .....	3
I.I    .Net Framework.....	3
I.II   Visual Studio IDE.....	3
Capitolul 2: Clase si obiecte .....	3
I.    C# - Clasele.....	4
I.I   Definirea unei clase .....	4
II.   Funcțiile membru și încapsularea .....	4
III.  C# Constructorii .....	4
IV.   C# Destructori .....	4
V.    Membri Statici într-o clasa C#.....	4
Capitolul 3: Încapsularea in C#.....	4
Specificatorul de acces PUBLIC.....	4
Specificatorul de acces PRIVATE.....	4
Specificatorul de acces PROTECTED .....	4
Specificatorul de acces INTERNAL .....	5
Specificatorul de acces PROTECTED INTERNAL .....	5
Capitolul 4: Moștenirea in C#.....	5
Clase de baza si clase derivate.....	5
Inițializarea clasei de baza .....	5
Moștenirile multiple in C# .....	5
Capitolul 5: Polimorfismul in C#.....	5

Polimorfismul Static.....	6
Supraîncărcarea funcțiilor .....	6
Polimorfismul Dinamic.....	6
Capitolul 6: Supraîncărcarea Operatorilor in C# .....	6
Implementarea Supraîncărcării Operatorilor .....	6
Capitolul 7: Suprascrierea in C# .....	6
Capitolul 8: Abstractizarea in C# .....	6
Capitolul 9: Interfețe in C# .....	6
Bibliografie .....	6

# Capitolul 0: Introducere in C#

C# este unul din limbajele moderne de programare utilizate la scară mare. Acesta este de uz general fiind de asemenea orientat pe obiecte. Dezvoltatorii limbajului sunt de la Microsoft și, împreună cu echipa de dezvoltare, au obținut cu succes aprobarea de la Asociația Europeană a Producătorilor de Calculatoare (ECMA) și de la Organizația Internațională pentru Standardizare (ISO). Limbajul C# este unul dintre limbajele dezvoltate pentru Common Language Infrastructure, iar versiunea actuală C# este versiunea 10.0 lansată în 2021. Acest limbaj este foarte asemănător cu limbajul de programare Java din punct de vedere sintactic, fiind foarte ușor de învățat și înțeles de către utilizatorii care au minime cunoștințe de C, C++ sau Java.

## Capitolul 1: Setarea mediului de lucru C# in Visual Studio

### *I. Componente de bază implicate în procesul de configurare a mediului în C#*

*I.I .Net Framework*

*I.II Visual Studio IDE*

## Capitolul 2: Clase si obiecte

## I. C# - Clasele

### I.1 Definirea unei clase

### II. Funcțiile membru și încapsularea

### III. C# Constructorii

### IV. C# Destructori

### V. Membri Statici într-o clasă C#

## Capitolul 3: Încapsularea în C#

### Specificatorul de acces PUBLIC

Specificatorul de acces public permite unei clase să-și expună variabilele membre și funcțiile membre la alte funcții și obiecte. Orice membru public poate fi accesat din afara clasei.

### Specificatorul de acces PRIVATE

Specificatorul de acces privat permite unei clase să-și ascundă variabilele membre și funcțiile membre de alte funcții și obiecte. Numai funcțiile din aceeași clasă pot accesa membrii săi privați. Chiar și o instanță a unei clase nu poate accesa membrii săi privați.

### Specificatorul de acces PROTECTED

Specificatorul de acces protejat permite unei clase copil să acceseze variabilele membre și funcțiile membre ale clasei sale de bază. În acest fel, ajută la implementarea moștenirii. Vom discuta acest lucru în mai multe detalii în capitolul Moștenire.

### Specificatorul de acces INTERNAL

Specificatorul de acces intern permite unei clase să-și expună variabilele membre și funcțiile membre la alte funcții și obiecte din ansamblul curent. Cu alte cuvinte, orice membru cu specificator de acces intern poate fi accesat din orice clasă sau metodă definită în cadrul aplicației în care este definit membrul.

### Specificatorul de acces PROTECTED INTERNAL

Specificatorul de acces intern protejat permite unei clase să-și ascundă variabilele membre și funcțiile membre de alte obiecte și funcții ale clasei, cu excepția unei clase copil din cadrul aceleiași aplicații. Acest lucru este folosit și în timpul implementării moștenirii [5].

## **Capitolul 4: Moștenirea in C#**

### Clase de baza si clase derivate

### Inițializarea clasei de baza

### Moștenirile multiple in C#

## **Capitolul 5: Polimorfismul in C#**

Cuvântul polimorfism înseamnă a avea mai multe forme. În paradigma de programare orientată pe obiecte, polimorfismul este adesea exprimat ca „o interfață, funcții multiple”.

Polimorfismul poate fi static sau dinamic. În polimorfismul static, răspunsul la o funcție este determinat în momentul compilării. În polimorfismul dinamic, se decide în timpul execuției.

Polimorfismul Static

Supraîncărcarea funcțiilor

Polimorfismul Dinamic

## **Capitolul 6: Supraîncărcarea Operatorilor in C#**

Puteți redefini sau supraîncărca majoritatea operatorilor încorporați disponibili în C#. Prin urmare, programatorii pot folosi și operatori definiți de utilizator. Operatorii supraîncărcați sunt funcții denumite special, care sunt operatori cheie urmați de simbolul operatorului definit. Ca și alte funcții, operatorii supraîncărcați au un tip de returnare și o listă de parametri.

Implementarea Supraîncărcării Operatorilor

## **Capitolul 7: Suprascrierea in C#**

## **Capitolul 8: Abstractizarea in C#**

## **Capitolul 9: Interfețe in C#**

## **Bibliografie**



[1] Mark J. Price. C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development: Build applications with C#, .NET Core, Entity Framework Core, ASP.NET Core, and ML.NET using ... Studio Code, 4th Edition (English Edition), ISBN: 978-1788478120.

[2] Jamie Chan. C#: Learn C# in One Day and Learn It Well. C# for Beginners with Hands-on Project, ISBN:1518800270.

[3] Andrew Stellman, Jennofer Greene. Head First C Sharp #: A Learner's Guide to Real-World Programming with C# and .NET Core, ISBN: 101491976705.

[4] Joseph Albahari, Ben Albahari. C# 8.0 Pocket Reference: Instant Help for C# 8.0 Programmers, ISBN: ISBN-101492051217.

[5] Anne Boehm, Joel Murach. Murach's C#, ISBN:978-1890774943.

[6] Ben Sharp. Microsoft Visual C# Step by Step, 10th Edition, ISBN 978-0-13-761983-2.

[7] Joseph Albahari. C# 10 in a Nutshell: The Definitive Reference 1st Edition, , ISBN: 978-1098121952.