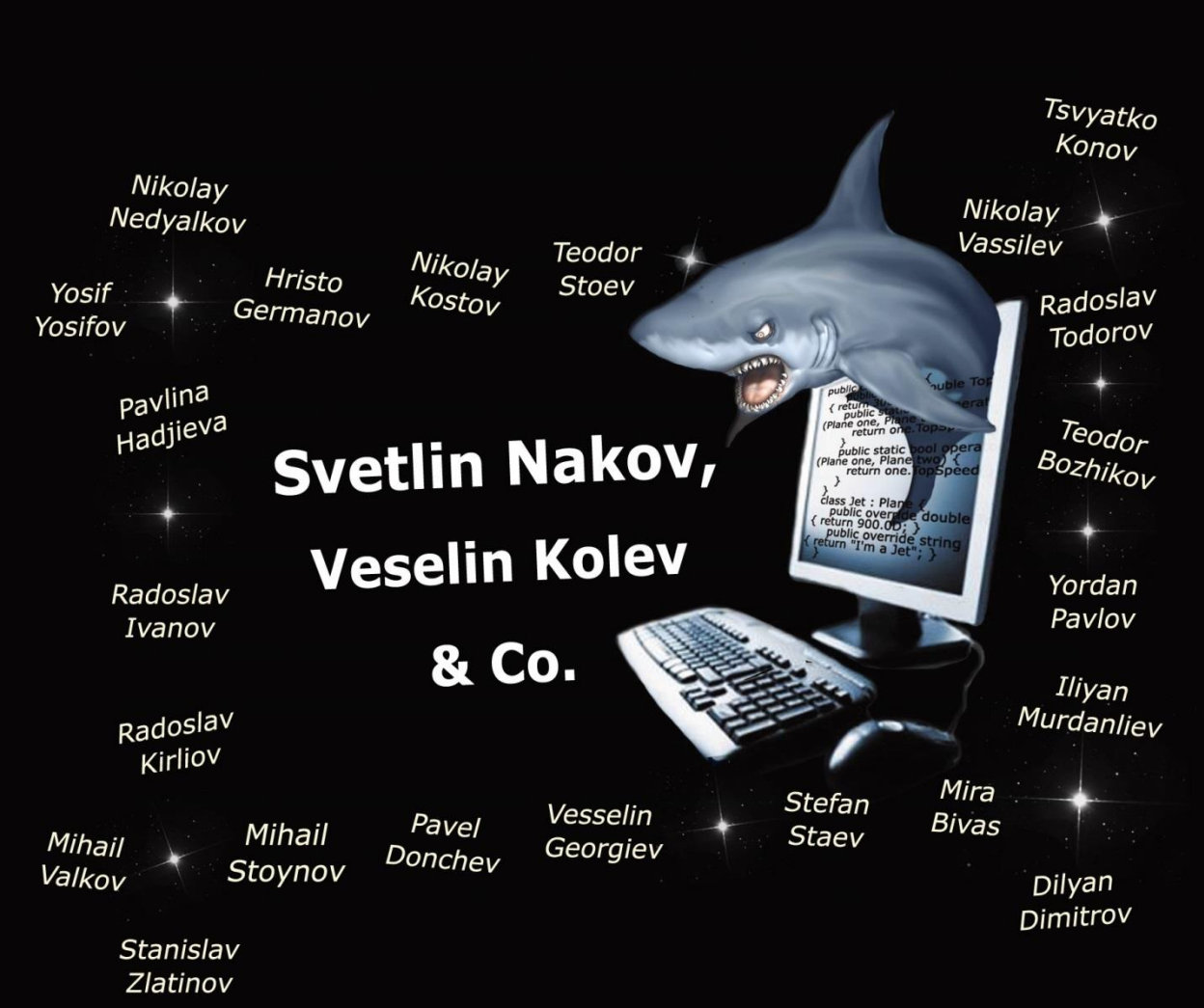

Fundamentals of Computer Programming with C#

The Bulgarian C# Book

Contents

Contents **2**
Detailed Table of Contents **5**
Preface **13**
Chapter 1. Introduction to Programming **69**
Chapter 2. Primitive Types and Variables **111**
Chapter 3. Operators and Expressions **139**
Chapter 4. Console Input and Output **165**
Chapter 5. Conditional Statements **195**
Chapter 6. Loops **211**
Chapter 7. Arrays **235**
Chapter 8. Numeral Systems **265**
Chapter 9. Methods **293**
Chapter 10. Recursion **351**
Chapter 11. Creating and Using Objects **385**
Chapter 12. Exception Handling **415**
Chapter 13. Strings and Text Processing **457**
Chapter 14. Defining Classes **499**
Chapter 15. Text Files **615**
Chapter 16. Linear Data Structures **641**
Chapter 17. Trees and Graphs **681**
Chapter 18. Dictionaries, Hash-Tables and Sets **727**
Chapter 19. Data Structures and Algorithm Complexity **769**
Chapter 20. Object-Oriented Programming Principles **807**
Chapter 21. High-Quality Programming Code **853**
Chapter 22. Lambda Expressions and LINQ **915**
Chapter 23. Methodology of Problem Solving **935**
Chapter 24. Sample Programming Exam – Topic #1 ..... **985**
Chapter 25. Sample Programming Exam – Topic #2 ..... **1041**
Chapter 26. Sample Programming Exam – Topic #3 ..... **1071**
Conclusion **1119**

Fundamentals of Computer Programming with C#

(The Bulgarian C# Programming Book)

Svetlin Nakov & Co.

Dilyan Dimitrov
Hristo Germanov
Iliyan Murdanliev
Mihail Stoynov
Mihail Valkov
Mira Bivas
Nikolay Kostov
Nikolay Nedyalkov
Nikolay Vasilev
Pavel Donchev
Pavlina Hadjieva
Radoslav Ivanov
Radoslav Kirilov
Radoslav Todorov
Stanislav Zlatinov
Stefan Staev
Svetlin Nakov
Teodor Bozhikov
Teodor Stoev
Tsvyatko Konov
Vesselin Georgiev
Veselin Kolev
Yordan Pavlov
Yosif Yosifov

Sofia, 2013

FUNDAMENTALS OF COMPUTER PROGRAMMING WITH C#

(The Bulgarian C# Programming Book)

Official Web Site:

http://www.introprogramming.info

ISBN 978-954-400-773-7

Detailed Table of Contents

Contents **2**
Detailed Table of Contents **5**
Preface **13**
About the Book 13
C# and .NET Framework 17
How To Read This Book? 22
Why Are Data Structures and Algorithms Emphasized? 25
Do You Really Want to Become a Programmer? 26
A Look at the Book's Contents 29
History: How Did This Book Come to Be? 38
Authors and Contributors 40
The Book Is Free of Charge! 53
Reviews 53
License 63
Resources Coming with the Book 65
Chapter 1. Introduction to Programming **69**
In This Chapter 69
What Does It Mean "To Program"? 69
Stages in Software Development 71
Our First C# Program 75
The C# Language and the .NET Platform 79
Visual Studio IDE 93
Alternatives to Visual Studio 104
Decompiling Code 104
C# in Linux, iOS and Android 107
Other .NET Languages 107
Exercises 108
Solutions and Guidelines 108
Chapter 2. Primitive Types and Variables **111**
In This Chapter 111
What Is a Variable? 111
Data Types 111
Variables 123
Value and Reference Types 128
Literals 131

Exercises....135
Solutions and Guidelines....136
Chapter 3. Operators and Expressions....139
In This Chapter....139
Operators....139
Type Conversion and Casting....152
Expressions....158
Exercises....160
Solutions and Guidelines....161
Chapter 4. Console Input and Output....165
In This Chapter....165
What Is the Console?....165
Standard Input-Output....169
Printing to the Console....169
Console Input....183
Console Input and Output – Examples....190
Exercises....192
Solutions and Guidelines....193
Chapter 5. Conditional Statements....195
In This Chapter....195
Comparison Operators and Boolean Expressions....195
Conditional Statements "if" and "if-else"....200
Conditional Statement "switch-case"....206
Exercises....208
Solutions and Guidelines....209
Chapter 6. Loops....211
In This Chapter....211
What Is a "Loop"?....211
While Loops....211
Do-While Loops....216
For Loops....221
Foreach Loops....225
Nested Loops....226
Exercises....231
Solutions and Guidelines....233
Chapter 7. Arrays....235
In This Chapter....235
What Is an "Array"?....235
Declaration and Allocation of Memory for Arrays....235
Access to the Elements of an Array....238
Reading an Array from the Console....241

Printing an Array to the Console 243
Iteration through Elements of an Array 244
Multidimensional Arrays 246
Arrays of Arrays 253
Exercises 257
Solutions and Guidelines 259
Chapter 8. Numeral Systems 265
In This Chapter 265
History in a Nutshell 265
Numeral Systems 266
Representation of Numbers 276
Exercises 289
Solutions and Guidelines 290
Chapter 9. Methods 293
In This Chapter 293
Subroutines in Programming 293
What Is a "Method"? 293
Why to Use Methods? 294
How to Declare, Implement and Invoke a Method? 295
Declaring Our Own Method 295
Implementation (Creation) of Own Method 300
Invoking a Method 301
Parameters in Methods 303
Returning a Result from a Method 328
Best Practices when Using Methods 345
Exercises 347
Solutions and Guidelines 348
Chapter 10. Recursion 351
In This Chapter 351
What Is Recursion? 351
Example of Recursion 351
Direct and Indirect Recursion 352
Bottom of Recursion 352
Creating Recursive Methods 352
Recursive Calculation of Factorial 353
Recursion or Iteration? 355
Simulation of N Nested Loops 356
Which is Better: Recursion or Iteration? 362
Using Recursion – Conclusions 378
Exercises 378
Solutions and Guidelines 380
Chapter 11. Creating and Using Objects 385

In This Chapter 385
Classes and Objects 385
Classes in C# 387
Creating and Using Objects 390
Namespaces 405
Exercises 410
Solutions and Guidelines 412
Chapter 12. Exception Handling 415
In This Chapter 415
What Is an Exception? 415
Exceptions Hierarchy 424
Throwing and Catching Exceptions 426
The try-finally Construct 432
IDisposable and the "using" Statement 437
Advantages of Using Exceptions 439
Best Practices when Using Exceptions 445
Exercises 453
Solutions and Guidelines 454
Chapter 13. Strings and Text Processing 457
In This Chapter 457
Strings 457
Strings Operations 462
Constructing Strings: the StringBuilder Class 480
String Formatting 488
Exercises 491
Solutions and Guidelines 496
Chapter 14. Defining Classes 499
In This Chapter 499
Custom Classes 499
Usage of Class and Objects 502
Organizing Classes in Files and Namespaces 505
Modifiers and Access Levels (Visibility) 508
Declaring Classes 509
The Reserved Word "this" 511
Fields 512
Methods 518
Accessing Non-Static Data of the Class 519
Hiding Fields with Local Variables 522
Visibility of Fields and Methods 524
Constructors 531
Properties 549
Static Classes and Static Members 559

Structures 580
Enumerations 584
Inner Classes (Nested Classes) 590
Generics 594
Exercises 610
Solutions and Guidelines 613
Chapter 15. Text Files 615
In This Chapter 615
Streams 615
Reading from a Text File 620
Writing to a Text File 628
Input / Output Exception Handling 630
Text Files – More Examples 631
Exercises 636
Solutions and Guidelines 638
Chapter 16. Linear Data Structures 641
In This Chapter 641
Abstract Data Structures 641
List Data Structures 642
Exercises 676
Solutions and Guidelines 678
Chapter 17. Trees and Graphs 681
In This Chapter 681
Tree Data Structures 681
Trees 681
Graphs 714
Exercises 722
Solutions and Guidelines 723
Chapter 18. Dictionaries, Hash-Tables and Sets 727
In This Chapter 727
Dictionary Data Structure 727
Hash-Tables 735
The "Set" Data Structure 760
Exercises 765
Solutions and Guidelines 767
Chapter 19. Data Structures and Algorithm Complexity 769
In This Chapter 769
Why Are Data Structures So Important? 769
Algorithm Complexity 770
Comparison between Basic Data Structures 779
When to Use a Particular Data Structure? 779

Choosing a Data Structure – Examples 786
External Libraries with .NET Collections 801
Exercises 803
Solutions and Guidelines 804
Chapter 20. Object-Oriented Programming Principles 807
In This Chapter 807
Let's Review: Classes and Objects 807
Object-Oriented Programming (OOP) 807
Fundamental Principles of OOP 808
Inheritance 809
Abstraction 824
Encapsulation 828
Polymorphism 830
Cohesion and Coupling 836
Object-Oriented Modeling (OOM) 842
UML Notation 844
Design Patterns 847
Exercises 851
Solutions and Guidelines 852
Chapter 21. High-Quality Programming Code 853
In This Chapter 853
Why Is Code Quality Important? 853
What Does Quality Programming Code Mean? 854
Why Should We Write Quality Code? 854
Identifier Naming 857
Code Formatting 866
High-Quality Classes 874
High-Quality Methods 878
Proper Use of Variables 883
Proper Use of Expressions 890
Use of Constants 891
Proper Use of Control Flow Statements 894
Defensive Programming 898
Code Documentation 900
Code Refactoring 904
Unit Testing 905
Additional Resources 912
Exercises 912
Solutions and Guidelines 913
Chapter 22. Lambda Expressions and LINQ 915
In This Chapter 915
Extension Methods 915