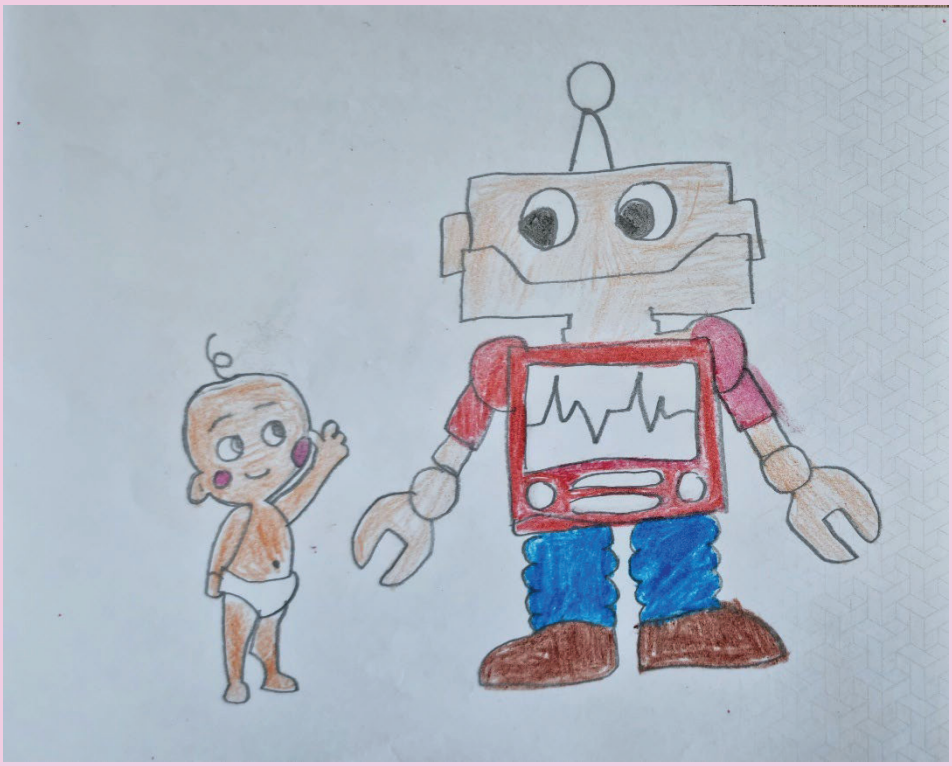
Alfa BK Univerzitet

Lecture Notes on Fundamentals of Programming

Exploring fundamental programming concepts using Java Language



Milan Đorđević

Izdavač
ALFA BK UNIVERZITET
Palmira Toljatija 3, Beograd

Za izdavača
prof. dr Jovan Veselinović

Recenzenti
prof. Dr Veljko Milutinović
Dr Jelena Stojanović
prof. Dr Mamoun Alazab
Dr Fadi Safieddine

CIP - Каталогизација у публикацији Народна библиотека Србије, Београд

004.438JAVA
004.42.045

ЂОРЂЕВИЋ, Милан, 1982-
Lecture Notes on Fundamentals of Programming : exploring fundamental programming concepts using Java Language / Milan Đorđević. - Beograd : Alfa BK University, 2024 (Beograd : 3D+). - 164 str. : ilustr. ; 24 cm

Autorova slika. - Tiraž 50. - About the author: str. 6-7.

ISBN 978-86-6461-076-6

a) Програмски језик "Java" b) Објектно оријентисано програмирање

COBISS.SR-ID 148030985

Lecture Notes on Fundamentals of Programming

Exploring fundamental programming concepts using Java Language

Milan Đorđević

CONTENTS

ABOUT THE AUTHOR

Dr. Milan Đorđević, an academic and researcher, holds the position of Associate Professor in Computer Science, where he brings his wealth of expertise in technology to institutions renowned for their commitment to excellence, such as American University of the Middle East (AUM) in Kuwait, and the Higher Colleges of Technology (HCT) in the United Arab Emirates (UAE) and the Alfa BK University in Serbia. His academic journey culminated in the attainment of a Ph.D. from the University of Primorska in Slovenia, where he delved into the intricate realm of grafted genetic algorithms.

In addition to his role as an educator, Dr. Đorđević is immersed in research, with a specific focus on optimization algorithms and harnessing the power of artificial intelligence to combat the dissemination of misinformation across online platforms. His dedication to fostering learning extends beyond conventional boundaries, as evidenced by his active involvement in initiatives like the Microsoft Imagine Cup. Here, he not only guides and mentors aspiring students but also showcases pioneering projects that push the boundaries of technological innovation.

Throughout his career, Dr. Đorđević has garnered widespread acclaim for his invaluable contributions to academia and society at large. Notably, he has been nominated for the esteemed Best Faculty Award and has received certificates of appreciation from both AUM and HCT, underscoring his exceptional commitment and dedication to advancing knowledge and empowering students. His research characterize interdisciplinary collaboration, aimed at enhancing the efficiency of technology through the application of cutting-edge computational techniques.

As a proud recipient of the prestigious Microsoft Imagine Cup award, Dr. Đorđević serves as a beacon of innovation, epitomizing an unwavering commitment to leveraging technology for the betterment of society. His roles at esteemed institutions not only underscore his dedication to shaping the next generation of technologists but also highlight his pivotal role in nurturing problem solvers capable of tackling the most pressing challenges of our time.

The courses taught by Dr. Đorđević are designed to equip students with both theoretical knowledge and practical skills essential for navigating the dynamic landscape

of modern technology. Through engaging hands-on projects and real-world applications, students are afforded invaluable insights into cutting-edge technologies and their myriad applications across diverse domains.

In summation, Dr. Đorđević academic odyssey is a testament to his commitment to excellence in research, education, and innovation. His active participation in international student competitions, coupled with his roles at prestigious institutions, establishes him as a driving force in shaping the future of technology and its profound societal impact.

INTRODUCTION

Exploring fundamental programming concepts using Java language. Topics include algorithms, variables, control statements, and methods. Throughout the practicum a series of progressive assignments help students gain hands-on experience in designing and writing algorithms to solve a computational problem, implementing programs within an integrated development environment, and explaining how programs implement algorithms in terms of instruction processing, program execution, and running processes. Learning Outcomes includes:

LO1 Apply basic programming concepts to write simple programs that use data types, variables, constants, expressions, and statements, with a focus on these constructs in Java.

LO2 Write programs that apply control flow via conditional statements and iterations.

LO3 Write programs that employ one-dimensional arrays of different data types with initialization, accessing, traversing, and searching.

LO4 Develop programs using built-in and user-defined functions with parameter lists.

Lecture Notes on Fundamentals of Programming

by

Milan Đorđević

I would like to dedicate this Lecture Notes to my wife Jasmina, my daughters Nina, Maša and Mila and my mother Slavica!

CHAPTER

1

PROGRAMING TECHNOLOGY – USE JAVA

In this practicum we will use the **high level** programming language called **Java**.

You do not need to download any software as we will use a cloud based solution called **Repl.it** which will allow you to write and compile code online. Before you can use it you should attempt to create an account. Signup and create an account at the following link: **http://www.repl.it** This will only take a few minutes and you can start coding immediately! We will use this development environment during every chapter. Make your account in **repl.it** for **Java** and have a look around the IDE (Integrated Development Environment). Look at the input screen on the Left. Look at the output screen on the Right

- OK you are ready to start programming.
- Let's create and understand the first Hello World Program

PROGRAMMING BASICS &TERMINOLOGY

The Hello World Program

Traditionally, the first program you write when learning a new programming language is called the ***Hello World*** *program*.

All it does is output the words Hello, World! to the screen. In Java, it looks like this:

Java code:

```
public class Main {
  public static void main(String[] args) {
    System.out.println("Hello, world!");
  }
}
```

Output:

```
Hello, world!
```

When this program runs, it displays the following:

Hello, World!

Notice that the output does not include the quotation marks. Java programs are made up of ***class*** and ***method*** definitions, and methods are made up of ***statements***. A **statement** is a line of code that performs a basic action.

In the Hello World program, this line is a **print statement** that displays a message back to the user:

System.out.println("Hello world!");

System.out.println displays results on the screen;

println stands for *print line*.

The print statement ends with a semicolon (;)

Java is case-sensitive, which means that uppercase and lowercase are not the same.

In the Hello World program **System** has to begin with an uppercase letter (**system** or **SYSTEM** won't work). A method is a named sequence of statements. This program defines one method named **main**:

public static void main(String[] args)

Java uses curly braces ({ and }) to group codes together. Every opening brace must have a closing brace.

The line that begins with two slashes (//) is a comment, which is a bit of English text that explains the code.

When Java sees //, it ignores everything from there until the end of the line.

Java code:

```
public class Main {
   public static void main(String[] args) {
      // Print the message "Hello, world!" to the console.
      System.out.println("Hello, world!");
   }
}
```

Output:

```
Hello, world!
```

Comments have no effect on the execution of the program, but they make it easier for other programmers (and your future self) to understand what you meant to do.

Displaying Two Messages

You can put as many statements as you like in the main method. For example, to display more than one line of output:

Java code:

```
public class Main {
   public static void main(String[] args) {
      // Display the first message: "Hello, world!"
      System.out.println("Hello, world!");
      // Display the second message: "How are you, Milan?"
      System.out.println("How are you, Milan?");
   }
}
```

Output:

```
Hello, world!
How are you, Milan?
```

You can put comments at the end of a line as well as on lines all by themselves.

Phrases that appear in quotation marks are called **strings**, because they contain a sequence of characters strung together in memory.

Characters can be letters, numbers, punctuation marks, symbols, spaces, tabs, etc.

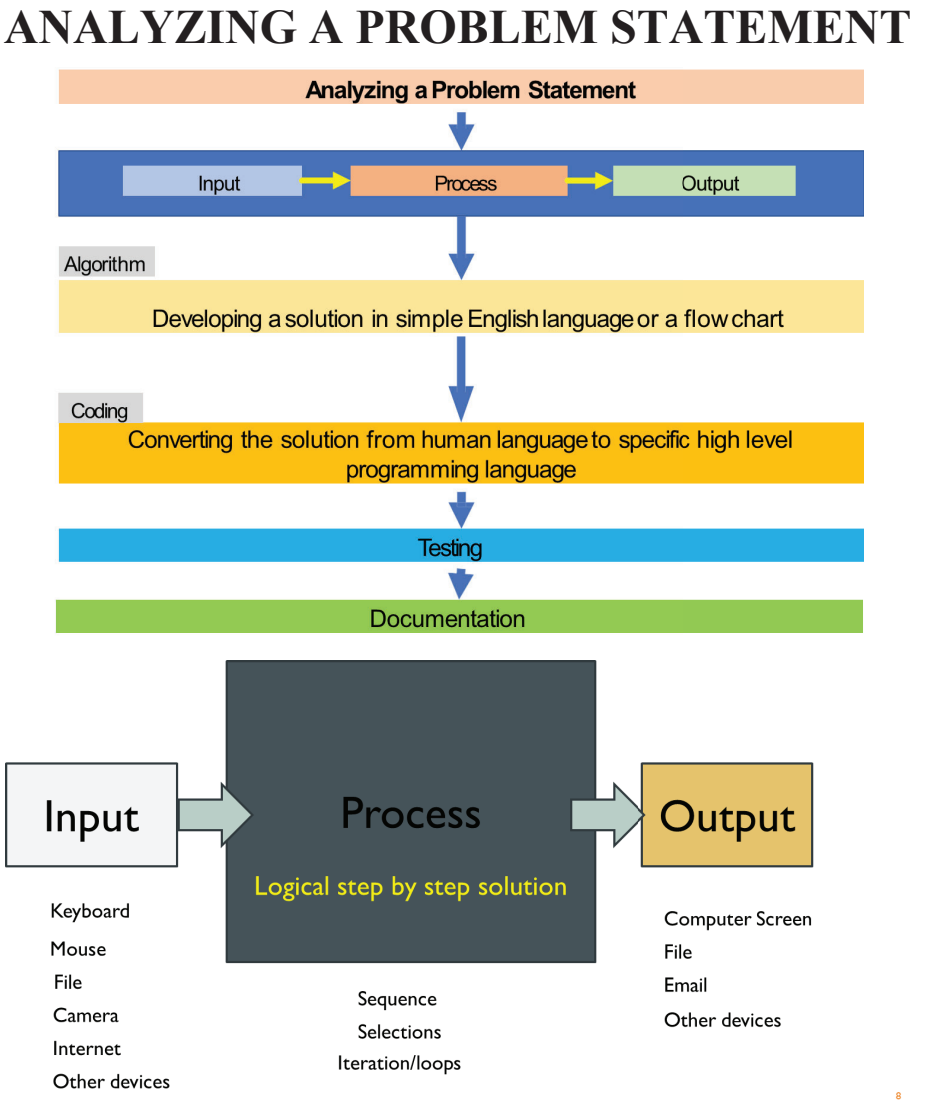
PROCESS OF PROGRAMMING

At the end of this subsection the student should be able to:

- Understand the process of programming
- Recognize the input, process and output

- Analyze a problem statement, develop a simple plan and translate into code

Before we can start to program we first need to understand what is needed. We need to analyze the problem and find a solution.



A simple example. Inputs-process-outputs with no coding.

Example 1:

1. Problem Statement:

   Develop a computer program that calculates and displays the VAT, the total cost including VAT with original price purchased by a customer.

2. Analysis:

   What outputs are produced by the program?
   Original price: 1000
   VAT: 50
   Total: 1050
   What are the inputs required for this solution?
   Total price → Original
   VAT = 5%

3. Algorithm:

   What are the steps required to solve this problem? A step by step description of the solution using plain human language. VAT = original * 5/100 VAT = 1000 * 5/100 = 50

   Total = original + VAT Total = 1000 + 50 = 1050

The step by step solution to solve this problem is:

Give/assign a value to the original price. We will give it the name original
Calculate VAT as: VAT = original * 5/100
Calculate Total as: Total = original + VAT
Print original
Print VAT
Print Total
Translate From Human English Language to Java Programming Language.

4. Coding

A Computer still can not fully understand human language. Coding is simply translating the solution from human language to high-level computer language. How different is a computer language from Human Language?

How different is a computer language from Human Language?

Human Language (English)	Computer Language (Java)
My name is Milan.	`String MyName = "Milan";`
My age is 42.	`int MyAge = 42;`

Good morning.
Good morning.
Good morning.
Good morning
	`for (int i=1; i<5; i++){`
	`System.out.println("Good morning.");`
	`}`

You pass the course if you score 60 or above.	`if (score >= 60) {`
	`System.out.println("Pass"); }`
	`else {`
	`System.out.println("Fail");`
	`}`

Human Language

3. Give/assign a value to the original price: **original**
4. Calculate VAT as: **VAT = original * 5/100**
5. Calculate Total as: **Total = original + VAT**
6. Print original
7. Print VAT
8. Print Total

Java

```
1  class Main {
2    public static void main(String[] args) {
3      float original = 1000;
4      float VAT = original * 5/100;
5      float total = original + VAT;
6      System.out.println("Original: "+original);
7      System.out.println("VAT: "+VAT);
8      System.out.println("Total: "+total);
9    }
10 }
```

DEBUGGING CODE

Debugging is the task of fixing coding errors. Large percentage of coding errors are resulting from: Missing semicolon ;

- Using variables before declaring them
- Using wrong spelling or capitalization when calling variables or a function
- Incorrect number of opened and closed brackets
- Using incorrect data type
- Using special characters or space as part of a variable or function name
- Not proving the correct parameters to a function
- Calling a function from an externals class without the associated object

Common coding errors: To avoid having many errors in your code, please make sure you follow these rules:
Most lines should end by a semicolon ; (The compiler will tell you which line is missing a semicolon)
Make sure you use the same name of the variable with the same case in all your code. In Java, VAT is different from Vat or vat.
Make sure you have no space in the variable name. For example original price is not allowed, but original_price or originalprice are allowed.
Make sure to close all open brackets. The number of opened brackets should match number of closed brackets.
Most keyword in Java (int, float, double, for, if, while, etc.) are lowercase

DOCUMENTATION

Comments are added to the actual code to improve its readability. Single line comments start by // allowing the programmer to write a short explanation of one line of code

//This line calculates the average

A Multi line (paragraph type of) comment. It begins with /* and ends by */. This type of comments is usually added at the beginning of the code or functions.

/

This program calculates the student GPA by adding the points of all course and dividing it by the number of credits.

**/*

JAVA PACKAGES

All modern programming languages come with strong library of codes.

In Java, this library is divided into packages.
Each package is a collection of files/classes the perform many functions.
You need simply to create a variable/object from the require classes and call all its associated code/functions.
This improves the efficiency and reduces the time of coding.
To connect a package to your code, you need to use the keyword import at the beginning of your code.
Example

```
import java.util.*;
import java.util.Scanner;
```

LAB A - CHAPTER 1

Apply basic programming concepts to write simple programs that use data types, variables, constants, expressions, and statements, with a focus on these constructs in Java

- Introduce various programming terminology
- Describe the process of programming

EXERCISE 1-KNOWLEDGE

1. Which statement is used to print output to the console in Java?

a) System.out.println();

b) System.in.read();

c) System.console();

d) Scanner.nextLine();

2. Which of the following is a valid single-line comment in Java?

a) / This is a comment

b) * This is a comment *

c) // This is a comment

d) # This is a comment

3. Which of the following is an advantage of using comments in Java?

a) Comments make the code execute faster

b) Comments make the code more complex

c) Comments make the code easier to understand

d) Comments are required by the Java compiler

4. Which of the following is used to identify the coding errors in a Java program?

a) Debugger

b) Compiler

c) IDE

d) CPU

5. Translating the programming solution to a problem from human language to a high-level computer language is known as:

a) debugging

b) compiling

c) coding

d) executing

EXERCISE 2-KNOWLEDGE

What is a Java package? Give one example of the Java package that will be used to take input from the user.

Answer:

In Java, a package is a way of organizing related classes and interfaces. You can use the Scanner class from the java.util package to read input from the keyboard.

Consider a program that asks the user to enter two integers, then calculates and displays the sum of these two integers.

Your Task:

1. Identify the input components for the given problem-solving scenario.
2. Clearly define the expected output of a problem-solving scenario.
3. Articulate the processing steps necessary to transform input into output.
4. Develop the Java code for the program.
5. Test your program using different sets of data.

Here is a sample run of the program:

Enter first number: **10**

Enter second number: **4**

The sum is: 14

Solution:

1. Input:
   - Input 1: First integer entered by the user
   - Input 2: Second Integer entered by the user
2. Expected Output:
   - The sum of the two integers.
3. Processing Steps (algorithm):
   - Step 1: Get number1
   - Step 2: Get number2
   - Step 3: Calculate the sum using the formula: sum = number1 + number2
   - Step 4: Display the sum

Java code:

```
import java.util.Scanner;

class Main {
   public static void main(String[] args) {
      Scanner scan = new Scanner(System.in);
      // Get number1
      System.out.println("Enter first number: ");
      int number1 = scan.nextInt();
      // Get number2
      System.out.println("Enter second number: ");
      int number2 = scan.nextInt();
      // Calculate the sum
      int sum = number1 + number2;
      // Display the sum
      System.out.println("The sum is: " + sum);
      scan.close(); // Close the Scanner object at the end of its
usage
   }
}
// End class Main
```

Output:

```
Enter first number: 10
Enter second number: 20
The sum is: 30
```

Closing the Scanner object after using it is crucial for efficient resource management, preventing memory leaks, avoiding file locks, ensuring exception safety, and maintaining robust and maintainable code. Properly closing the Scanner releases system resources, prevents resource leaks, and adheres to best practices.

EXERCISE 3-APPLY

Consider a Java program that prompts the user for 2 numbers as integers, then calculates and displays the product and the difference of the 2 numbers.

Your Task:

1. Identify the input components for the given problem-solving scenario.
2. Clearly define the expected output of a problem-solving scenario.
3. Articulate the processing steps necessary to transform input into output.
4. Develop the Java code for the program.
5. Test your program using different sets of data.

Here is a sample run of the program:

Enter number1: 7

Enter number2: 4

Product = 28

Difference = 3

Solution:

1. Input:
   - Input 1: First integer entered by the user
   - Input 2: Second Integer entered by the user.
2. Expected Output:
   - The sum of the two integers.
3. Processing Steps (algorithm):
   - Step 1: Get number1
   - Step 2: Get number2
   - Step 3: Calculate the product using the formula: product = number1*number2

- Step 4: Calculate the difference using the formula: difference = number1-number2
- Step 5: Display the product
- Step 6: Display the difference

4. Java code:

```
import java.util.Scanner;

public class ArithmeticOperations {
    public static void main(String[] args) {
        // Create Scanner object for input
        Scanner scanner = new Scanner(System.in);

        // Get first number from user
        System.out.print("Enter number1: ");
        int number1 = scanner.nextInt();

        // Get second number from user
        System.out.print("Enter number2: ");
        int number2 = scanner.nextInt();

        // Calculate sum, product, and difference
        int sum = number1 + number2;
        int product = number1 * number2;
        int difference = number1 - number2;

        // Output the results
        System.out.println("Sum = " + sum);
        System.out.println("Product = " + product);
        System.out.println("Difference = " + difference);

        // Close the scanner
        scanner.close();
    }
}
```

EXERCISE 4-APPLY

Consider a Java program that asks the user to enter three numbers as decimals, then calculates and displays the average of the three numbers.

Your Task:

1. Identify the input components for the given problem-solving scenario.
2. Clearly define the expected output of a problem-solving scenario.
3. Articulate the processing steps necessary to transform input into output.
4. Develop the Java code for the program.
5. Test your program using different sets of data.

Here is a sample run of the program:

Enter first number: **2.5**

Enter second number 2: **3.5**

Enter third number 3: **6**

Average: 4.0

Solution:

1. Input:
   - Input 1: First decimal number entered by the user
   - Input 2: Second decimal number entered by the user.
   - Input 3: Third decimal number entered by the user.
2. Expected Output:
   - The average of the two integers.
3. Processing Steps (algorithm):
   - Step 1: Get number1
   - Step 2: Get number2
   - Step 3: Get number3
   - Step 4: Calculate the average using the formula: average = (number1+number2+number3)/3
   - Step 5: Display the average
4. Java code:

```
import java.util.Scanner;

public class AverageCalculator {
    public static void main(String[] args) {
        // Create Scanner object for input
        Scanner scanner = new Scanner(System.in);

        // Get the three numbers from user
        System.out.print("Enter first number: ");
        double number1 = scanner.nextDouble();
        System.out.print("Enter second number: ");
```

```
        double number2 = scanner.nextDouble();
        System.out.print("Enter third number: ");
        double number3 = scanner.nextDouble();

        // Calculate the average of the three numbers
        double average = (number1 + number2 + number3) / 3;

        // Display the average
        System.out.println("Average: " + average);

        // Close the scanner
        scanner.close();
    }
}
```

EXERCISE 5-APPLY

Consider a Java program that asks the user for the original price then calculates and displays the vat amount and the total cost including the vat. The vat rate is 5% and vat amount = vat rate *original price.

Your Task:

1. Write the algorithm for the given problem.
2. Develop the Java code for the given problem.

Here is a sample run of the program:

Enter original price: **100**

Vat amount: **5 AED**

Total cost: 105 AED

Solution:

Algorithm:

Step 1: Get original price

Step 2: Calculate vat amount as 0.05* original price

Step 3: Calculate total cost = original price +vat amount

Step4: Display vat amount

Step 5: Display total cost

Java code:

```
import java.util.*;

class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      // Create Scanner object

      System.out.println("Enter original price:");
      double originalPrice = scanner.nextDouble();
      // Get original price

      double vat = 0.05 * originalPrice;
      // Calculate vat amount

      double totalCost = originalPrice + vat;
      // Calculate total cost

      System.out.println("Vat amount: " + vat + " AED");
      // Display vat amount

      System.out.println("Total cost: " + totalCost + " AED");
      // Display total cost

      scanner.close(); // Close the Scanner object at the end of its usage
   }
```