



CS 312

PROGRAMMING IN JAVA

Updated: Protocol. No. 23 / 28.06.2022

Lecturer: Assist. Prof. Aleksandar Ivanov, PhD

ANNOTATION

The discipline "Programming in JAVA" is a basic course for majors in the field 4.6. Informatics and Computer Science. The course introduces the basic Java tools for basic and object-oriented programming – variables, conditional operators, loop operators, working with classes, objects, constructors, dynamic and static variables and methods, inheritance, polymorphism; file handling and exception handling; programming with visual components.

MAIN OBJECTIVES

The aim of this course is to introduce students to the basic tools of the Java language for object oriented programming. Specifically, students will be able to:

- to understand the essence of the basic syntactic rules of the Java language;
- apply the basic principles of object-oriented programming through the tools of Java
- acquire practical skills for developing applications with visual components.

PREREQUISITES

For the successful completion of the course, knowledge of Programming and Object-oriented programming is required.

FORM OF EDUCATION

The forms of education are regular, part-time and distance learning.

STATUS AND STRUCTURE

Major	Status	ECTS Credits	Full-time study				Part-time study			
			L	S	P	Total	L	S	P	Total
Software engineering	Mandatory	6	30	30	0	60	15	15	0	30
Computer systems and technologies	Electable	6	30	30	0	60	15	15	0	30

COURSE SYLLABUS

Topic 1. The Java language - general characteristics, advantages of the Java language, areas of application. Structure and execution of a simple Java program. Primitive data types: constants, variables and operations. Assignment operator. Priority of operations. Conversion of values between primitive types.

Topic 2. Conditional operators — if, else, switch. Nested conditions. Boolean expressions. Comparing the values of objects.

Topic 3. Loop operators — for, while. Nested loops.

Topic 4. Basic input and output.

Topic 5. Defining a class, constructor, field, method, object. Access fields and methods of the class. The modifiers public, private, static, final. Scope of variables.

Topic 6. Arrays - one-dimensional and multidimensional. Basic array operations. Arrays of primitive values and arrays of objects.

Topic 7. Expanding knowledge about class and object. Static and dynamic methods. Passing values between formal and factual parameters. Access the fields and the class and object methods.

Topic 8. Class inheritance. Abstract class. Interfaces. Polymorphism. Conversion between classes (upcasting/downcasting). Redefining and defining a method. Internal and anonymous classes - syntax and applications. Implementation and application of basic interfaces: Iterable, Iterator and Comparable.

Topic 9. Classes and interfaces implementing data collections: Collection, List, Set, LinkedList, HashSet. HashMap.

Topic 10. Exceptions. Exception handling. Custom exceptions. Working with files.

Topic 11. Character strings. The String class. Convert from primitive type to String and reverse, automatic conversion and conversion methods.

Topic 12. Building Java GUIs

Topic 13. Visual applications - Layout of visual components and processing of events.

Topic 14. Basic visual components

SEMINAR EXERCISES

Topic 1. Creating an Executable Class. Basic primitive data types, arithmetic, boolean expressions. Priority of operations. Convert values between primitives types.

Topic 2. Creating simple programs with conditions

Topic 3. Solving tasks using loops.

Topic 4. Reading values from user input.

Topic 5. Classes. Constructors. Dynamic methods and fields.

Topic 6. Arrays - definition, methods and properties of arrays. Arrays of primitive values and arrays of objects. Passing arrays as arguments. Applications of arrays, element summation, statistical analyses (feature frequency counting), and sorting.

Topic 7. Creating static methods and fields.

Topic 8. Creating a hierarchy of classes. Abstract classes and methods. Interfaces - syntax and applications. Implementation and use of standard interfaces: *Iterable*, *Iterator* and *Comparable*.

Topic 9. Using collections. Examples revealing the advantages of using the hierarchy of classes and interfaces that implement data collections.

Topic 10. Reading and writing from a file. Object serialization. Applying the interface *Serializable*. Exception handling. Catching and throwing an exception

Topic 11. Working with character strings.

Topic 12. Basic visual components and technology of building a visual interface. Creating applications with a visual interface.

Topic 13. Technology for processing events related to visual components. Using methods when creating a graphical interface. Working with different types of events and implementing interfaces for their processing.

Topic 14. Development of a course assignment on the use of visual components and handling events, exceptions, collections.

COURSEWORK

The completion of the coursework requires each student to receive individual assignment. The topic is specified in agreement with the teacher. The student must code a visual application implementing the relevant functionality from the task.

PLANNED LEARNING ACTIVITIES AND TEACHING METHODS

The planned learning activities in this course aid the learning process. For this purpose:

- At the first lecture, students receive complete information about the content of the program of the discipline, the requirements for completion of the course and the synopsis;
- The lectures follow the course syllabus, grouped by topics and ending with summaries and self-study learning materials;
- Work with the students happens in in small groups (in seminar exercises);
- Intermediate tests are conducted to assess the progress of the students;
- Course assignments with practical examples are introduced at the end of the learning materials which allow for individual work;
- Electronic learning materials support the learning process of students and expand the possibilities of the teacher in the implementation of intermediate control.

EVALUATION METHODS AND CRITERIA

Evaluation criteria are complex and are based on the following factors:

- Intermediate tests: 2 tests of 10 points each. Total 20 points current control
- Coursework: 30 points
- Final exam: 50 points

Note: The evaluation of the coursework is as follows: up to 10 pts for proper function; 10pts for graphical interface; up to 10pts for implemented functionality - a total of 30pts. After summing up the received points, the final grade is formed according to a predefined scale.

RECOMMENDED LITERATURE

1. <https://docs.oracle.com/en/java/>
2. https://www.tutorialspoint.com/java/java_documentation.htm
3. <https://www.w3schools.com/java/>
4. <https://www.javatpoint.com/java-tutorial>
5. <https://www.programiz.com/java-programming>
6. <https://www.geeksforgeeks.org/java/>
7. <https://www.guru99.com/java-tutorial.html>