Federal State Autonomous Educational Institution of Higher Education "Moscow Institute of Physics and Technology (National Research University)"

APPROVED
Head of the Phystech School of Applied Mathematics and Informatics
A.M. Raygorodskiy

Work program of the course (training module)

course:	Java/Язык программирования JAVA
major:	Applied Mathematics and Informatics
specialization:	Computer Science/Информатика
	Phystech School of Applied Mathematics and Informatics
	Chair of Algorithms and Programming Technologies
term:	3
qualification:	Bachelor

Semester, form of interim assessment: 5 (fall) - Grading test

Academic hours: 60 AH in total, including:
lectures: 30 AH.
seminars: 0 AH.
laboratory practical: 30 AH.

Independent work: 120 AH.

In total: 180 AH, credits in total: 4

Authors of the program:

O.N. Ivchenko, senior professor
I.E. Stolov, assistant

The program was discussed at the Chair of Algorithms and Programming Technologies 21.05.2020

Annotation

The aim of this course is to acquire basic skills in all stages of industrial software product development, from architecture design, coding, and ending with test automation and deployment.The main programming language for the course is Java, the most popular industrial programming language. The basics of programming in Java, tools for developing, building and testing programs in this language will be considered. Also, the course will touch upon the topics of design patterns, test-driven development, continuous delivery / integration.

Practical classes will aim to master the skills of writing programs in Java, linking programs with other systems (for example, databases and web services), testing, deploying, building and publishing programs, including in the form of web services.

1. Study objective

Purpose of the course

Mastering the rules of the Java programming language and the techniques of using the Java language in programming practice.

Tasks of the course

The acquisition by students of skills in the design and implementation of applications in the Java language using the techniques of object-oriented programming, multithreading primitives and web technologies; mastering by students of modern development practices: using IDE, version control systems, unit testing.

2. List of the planned results of the course (training module), correlated with the planned results of the mastering the educational program

Mastering the discipline is aimed at the formation of the following competencies:

Code and the name of the competence	Competency indicators
UC-3 Interact effectively with project team members and fulfill one's role properly	UC-3.1 Establish different types of communication (educational, scientific, business, informal, etc.)
	UC-3.2 Interact with other team members to fulfill the project objectives
Gen.Pro.C-2 Use modern IT and software tools to perform professional tasks in compliance with information security requirements	Gen.Pro.C-2.1 Apply modern computing tools and Internet services in professional settings
	Gen.Pro.C-2.2 Apply numerical mathematical methods and use software applications for scientific problem-solving in professional settings

3. List of the planned results of the course (training module)

As a result of studying the course the student should:

know:

- The principle of executing Java programs using the JVM; how garbage collection works in Java;
- Java data types;
- execution flow control in Java;
- main classes and capabilities of the standard library;
- rules for working with exceptions;
- principles of development of parameterized classes and methods (generics);
- the internal structure of the standard library containers and the time complexity of their operations;
- stream API data processing;
- interaction with relational DBMS using JDBC API;
- principles of development of multithreaded code in Java and tools of the standard library; Java memory model;
- Java Reflection API capabilities;
- applying annotations and processing annotations at the Reflection API level;
- how the DI container works.

be able to:

- Implement a general-purpose library in the Java language using specified interfaces;
- add support for multithreading to the application, analyze the thread safety of the implementation;
- cover code with unit tests using the JUnit framework, analyze code coverage with tests;
- work with a distributed version control system git;
- use code review tools on the Github service;
- implement an application that provides an HTTP API using the Spring framework.

master:

- Skills in working with objects and flows and outlook in choosing an architectural solution to the task.

4. Content of the course (training module), structured by topics (sections), indicating the number of allocated academic hours and types of training sessions

4.1. The sections of the course (training module) and the complexity of the types of training sessions

№	Topic (section) of the course	Types of training sessions, including independent work			
		Lectures	Seminars	Laboratory practical	Independent work
1	Java syntax	6		6	12
2	Arrays, Collections	3		3	12
3	OOP	3		3	12
4	General-purpose instruments	3		3	12
5	Application testing, Java build tools	3		3	18
6	Stream API	3		3	12
7	Multithreading & Concurrency	3		3	18
8	Databases	3		3	12
9	Spring framework	3		3	12
AH in total		30		30	120
Exam preparation		0 AH.			
Total complexity		180 AH., credits in total 4			

4.2. Content of the course (training module), structured by topics (sections)

Semester: 5 (Fall)

1. Java syntax

This module is aimed to introduce the students to the Java syntax. Keywords, identifiers, data types, literals, branches, cycles, primitive types and objects, Input-output are covered.

2. Arrays, Collections

This module gives the overview on arrays, Collections and Strings in Java, their functionality and operations' time complexity.

3. OOP

Object-Oriented Programming principles and best practices are discussed in this module. Also, structure of classes and interfaces in Java are taken into account.

4. General-purpose instruments

This module is dedicated to working with date and time, generics, enums, regular expressions, JavaDoc and other general purpose instruments

5. Application testing, Java build tools

Application building with Maven and testing & continuous integration

6. Stream API

This module is devoted to data streams in Java, their abilities and management

7. Multithreading & Concurrency

Multithreaded java programming, starting from the theory of concurrency and Java memory model. Also, low-level and high-level instruments of synchronization, Runnable and Callable interfaces, Executors and concurrent data structures are discussed.

8. Databases

Communication between Java applications and databases. Sql fundamentals and DB principles

9. Spring framework

An overview on Spring framework, including HTTP API application implementation.

5. Description of the material and technical facilities that are necessary for the implementation of the educational process of the course (training module)

An auditorium with whiteboard and projector or plasma panel.

6. List of the main and additional literature, that is necessary for the course (training module) mastering

Main literature

Additional literature

7. List of web resources that are necessary for the course (training module) mastering

Not used

8. List of information technologies used for implementation of the educational process, including a list of software and information reference systems (if necessary)

1. Oracle Java JDK 11 or later.
2. IntellijIDEA IDE or latest version of Eclipse.

9. Guidelines for students to master the course

Successful mastering the course requires intense unaided work of the student. The course program provides the minimum required time for a student to work on a topic. Independent work includes:
- study of educational material (based on lecture notes, educational and scientific literature), preparation of answers to questions intended for independent study, proof of individual statements, properties;
- preparation for practical training, two individual homework assignments.

Intermediate knowledge control is carried out in the form of written questionnaires on the theory, and during the course of mastering the course, the student must complete three individual homeworks.

SUPPLEMENT

Assessment funds for course (training module)

major:	Applied Mathematics and Informatics
specialization:	Computer Science/Информатика
	Phystech School of Applied Mathematics and Informatics
	Chair of Algorithms and Programming Technologies
term:	3
qualification:	Bachelor

Semester, form of interim assessment: 5 (fall) - Grading test

Authors:

O.N. Ivchenko, senior professor
I.E. Stolov, assistant

1. Competencies formed during the process of studying the course

Code and the name of the competence	Competency indicators
UC-3 Interact effectively with project team members and fulfill one's role properly	UC-3.1 Establish different types of communication (educational, scientific, business, informal, etc.)
	UC-3.2 Interact with other team members to fulfill the project objectives
Gen.Pro.C-2 Use modern IT and software tools to perform professional tasks in compliance with information security requirements	Gen.Pro.C-2.1 Apply modern computing tools and Internet services in professional settings
	Gen.Pro.C-2.2 Apply numerical mathematical methods and use software applications for scientific problem-solving in professional settings

2. Competency assessment indicators

As a result of studying the course the student should:

know:

- The principle of executing Java programs using the JVM; how garbage collection works in Java;
- Java data types;
- execution flow control in Java;
- main classes and capabilities of the standard library;
- rules for working with exceptions;
- principles of development of parameterized classes and methods (generics);
- the internal structure of the standard library containers and the time complexity of their operations;
- stream API data processing;
- interaction with relational DBMS using JDBC API;
- principles of development of multithreaded code in Java and tools of the standard library; Java memory model;
- Java Reflection API capabilities;
- applying annotations and processing annotations at the Reflection API level;
- how the DI container works.

be able to:

- Implement a general-purpose library in the Java language using specified interfaces;
- add support for multithreading to the application, analyze the thread safety of the implementation;
- cover code with unit tests using the JUnit framework, analyze code coverage with tests;
- work with a distributed version control system git;
- use code review tools on the Github service;
- implement an application that provides an HTTP API using the Spring framework.

master:

- Skills in working with objects and flows and outlook in choosing an architectural solution to the task.

3. List of typical control tasks used to evaluate knowledge and skills

Problem A. Always know your enemy.

Your task is to create deadlock and livelock. Also, you need to explain (in JavaDoc comment or in separate file) the correctness of your solution.

Grading system.

2 points for deadlock implementation.

3 points for livelock implementation.

1 point for tests / demonstrations that show the correctness of your code.

Notes

A deadlock is a state in which each member of a group of actions is waiting for some other member to release a lock.

Livelock occurs when two or more processes continually repeat the same interaction in response to changes in the other processes without doing any useful work. These processes are not in the waiting state, and they are running concurrently.

You may use this project for help. There you will find test cases for livelock and deadlock. However, keep in mind that these tests do not guarantee that your code works / doesn't work, since they don't cover all possible cases.

Problem B. Almost real database

Your task is to improve the toy database from the 10th seminar. As you remember, this implementation allows multiple readers to read at the same time but there is a problem that writers need to wait for all the readers to finish reading. So, there is a possible situation when readers keep coming and each time a new reader appears before the previous one finishes reading. This results in starvation for a writer. Your task is to fix that problem.

Easy version: we suppose that there are few writers. Formally, the average time period between two write operations is much greater than the average write duration. So, in the easy version you don't need to think about the readers' starvation. Your goal is to come up with an implementation that

Allows readers to read simultaneously.

Guarantees that a writer won't wait more than constant * max_read_time before it starts writing in most cases.

Hard version: we don't suppose anything about readers and writers. In the hard version you need to build a solution that gives the following guarantees:

Readers can read simultaneously.

Readers and writers perform their operations in 'FIFO' order.

For example, if threads appear in following order R1, W1, R2, R3, R4, W2 (R -- reader, W-- writer) then, R1 reads first, after that W1 writes, then R2, R3 and R4 read simultaneously and after that W2 writes. This derandomization guarantees the lack of starvation.

Grading system.

5 points for easy version with tests.

10 points for hard version with tests.

4. Evaluation criteria

1. What is the difference between overloading and overloading?
2. What is the difference between an abstract class and an interface? Show how these differences have changed in Java 8+.
3. In what ways can you create a Stream in Java.
4. What is the difference between JRE, JVM and JDK?
5. What does the final keyword say?
6. What values are the default variables initialized to?
7. What is the abstract modifier used for?
8. What are marker interfaces? How are they different from ordinary ones?
9. What is the use of static initialization blocks in Java?
10. Can a static method be overridden or overloaded?

excellent

10 comprehensive, systematized, deep knowledge of the curriculum of the discipline and the ability to confidently apply them in practice when solving specific problems, free and correct justification of decisions made;

9 systematic, deep knowledge of the curriculum of the discipline and the ability to confidently apply them in practice when solving specific problems, the correct justification of decisions made;

8 deep knowledge of the curriculum of the discipline and the ability to apply them in practice when solving specific problems, the correct justification of decisions made;

good

7 firmly knows the material, correctly and essentially sets out it, knows how to apply the knowledge gained in practice, but admits some inaccuracies in the answer or in solving problems;

6 knows the material, correctly presents it, knows how to apply the acquired knowledge in practice, but admits some inaccuracies in the answer or in solving problems;

5 knows the basic material, correctly presents it, knows how to apply the knowledge gained in practice, but admits inaccuracy in the answer or in solving problems;

satisfactorily

4 fragmented, fragmented nature of knowledge, insufficiently correct wording of basic concepts, violation of logical sequence in the presentation of program material, but at the same time he owns the main sections of the curriculum necessary for further training and can apply the acquired knowledge in the standard situation;

3 the nature of knowledge is sufficient for further training and can apply the acquired knowledge on the model in a standard situation;

unsatisfactory

2 does not know most of the main content of the curriculum of the discipline, makes gross errors in the wording of the basic concepts of the discipline and does not know how to correctly use the knowledge gained in solving typical practical problems.

1 does not know the wording of the basic concepts of the discipline and does not know how to use the knowledge gained in solving typical practical problems.

5. Methodological materials defining the procedures for the assessment of knowledge, skills, abilities and/or experience

Differentiated classification is carried out taking into account current academic performance and the results of the completion of term paper. If necessary, in the process of interviewing a student, a selective survey is conducted on the knowledge of control questions, and typical tasks are offered.