



Syllabus

Java Programming Language

Duration of the course and training schedule

- The course involves 184 training double-classes.
- Duration is 1 year.
- Training schedule – 2 double-classes twice a week and at least 6 hours for completion of homework assignment.

By the end of the course the student will be able to:

- develop software using Java;
- project classes of varying degrees of complexity and create hierarchies of classes in order to solve practical problems;
- use fundamental principles of development of back-end solutions by means of Java;
- understand mechanisms of multi-threading in Java.

Advantages of the Java course

- Educational program is arranged in accordance with the IT market requirements.
- The course aims to provide students with practical skills of development; the program involves the delivery of summative course papers and final project, which will be included into the students' portfolios.
- The course is taught by skilled teachers.
- Duration of the course is 1 year. It allows students to get professional skills in the field of development of software projects.
- Students are taught in groups of small size.
- Classes are conducted in comfortable classrooms using the advanced equipment.
- Trainings on the development of soft skills.
- The course of technical English.

Graduates of the course will be able to work in IT and start-up companies.

Enrollment Requirements

- The age from 15 to 55 years.
- Strong knowledge of the PC.

Plan

1. **Java Programming Language** 80 double-classes
2. **Database Theory** 16 double-classes
3. **Creating Web-pages Using HTML5 and CSS3** 14 double-classes
4. **Developing Client-Side Scripts Using JavaScript** 16 double-classes
5. **Developing Back-End Solutions Using Java**..... 58 double-classes

Java Programming Language

Version 2.0.0

Course duration: 80 double-classes

Course objective

To teach the student to program in Java. Learn to apply the right mechanisms and constructions to solve a particular problem.

Upon the completion of the course, the student will be able to:

- understand fundamentals of creating programs using Java;
- create, compile, and debug projects in IDE Eclipse;
- project and implement various algorithms;
- use mechanisms of conditions and loops;
- apply arrays to store data;
- use sorting algorithms and data search;
- understand OOP principles;
- project classes of varying complexity;
- create class hierarchies to solve real world problems;
- use Generics to build template classes;
- throw and handle exceptions;
- choose and use JCF classes;
- save and read information from files;
- understand Java multithreading mechanisms;
- use lambdas;
- use version control system;
- understand basics of teamwork;
- apply projection patterns;
- use unit testing.

Upon completion of this course, the student submits a practical task and takes a theoretical exam on course materials. For admission to the examination, all home and practical tasks must be submitted. Practical task should cover a maximum of material from different sections of the course.

Topic Plan

Module 1.	Introduction to the Java programming language	4 double-classes
Module 2.	Variables, data types, operators.	4 double-classes
Module 3.	Logical operators, branching operators, bitwise operators	5 double-classes
Module 4.	Loops.	5 double-classes
Module 5.	Strings, one-dimensional and multi-dimensional arrays	5 double-classes
Module 6.	Methods (through the example of static methods).	4 double-classes
Module 7.	Object-oriented programming.	16 double-classes
Module 8.	Exceptions	4 double-classes
Module 9.	JavaCollectionFramework	4 double-classes
Module 10.	Annotations, anonymous classes, Lambda expressions	2 double-classes
Module 11.	Working with files	3 double-classes
Module 12.	Stream API.	2 double-classes
Module 13.	Multithreading.	3 double-classes
Module 14.	Version control systems	3 double-classes
Module 15.	Teamwork, software project management	4 double-classes
Module 16.	Usage of junit.	2 double-classes
Module 17.	Projection patterns.	4 double-classes
Module 18.	The MVC pattern.	2 double-classes
Module 19.	Principles of designing SOLID classes	2 double-classes
Module 20.	Exam	2 double-classes

Module 1. Introduction to the Java programming language

1. Introduction.
2. Algorithm.
3. The concept of flow-chart.
4. Eclipse Development Environment.
5. JShell.

Module 2. Variables, data types, operators

1. Data types.
2. Variable.
3. Constants and literals.
4. Operators.
5. Examples of program development using flow-charts.

Module 3. Logical operators, branching operators, bitwise operators

1. Data type conversion.
2. Logical operators.
3. Table of priority of operators.
4. Logical decision constructions. Branching operators.
5. Bitwise operators.

Module 4. Loops

1. Loops.
2. Working with the Eclipse integrated debugger.

Module 5. Strings, one-dimensional and multi-dimensional arrays

1. Working with strings.
2. Arrays.
3. Addition algorithms.
4. Search algorithms.
5. Sorting algorithms.
6. The concept of the algorithm complexity.
7. Multi-dimensional arrays.

Module 6. Methods (through the example of static methods)

1. Methods.
2. Scope.
3. Recursion.

Module 7. Object-oriented programming

1. Introduction to the object-oriented programming.
2. The concept of class.
3. The concept of object.
4. The concept of a class member, class field, and class method.
5. Access specifiers.
6. Object constructors.
7. The this keyword.
8. Overloading methods and constructors.
9. Static methods of classes.
10. Passing objects to a method.
11. Scope in methods of classes.
12. Inheritance.
13. The concept of interface.
14. The final keyword.
15. Garbage collection.
16. Packages.
17. Generics.

Module 8. Exceptions

1. What is an exception situation?
2. Principles of handling exceptions.
3. The concept of checked and unchecked exceptions.
4. The try keyword.
5. The catch keyword.

6. The throw keyword.
7. The finally keyword.
8. Details of using exception situations.
9. Callstack unwinding.

Module 9. JavaCollectionFramework

1. Wrapper classes.
2. Introduction to JCF.
3. JCF interfaces.
4. Creation of collections using factory methods.
5. JCF classes.

Module 10. Annotations, anonymous classes, Lambda expressions

1. Annotations.
2. Anonymous classes.
3. Lambda expressions.

Module 11. Working with files

1. Introduction to the java.io package.
2. Input/output streams.
3. Object serialization.

Module 12. Working with files

1. Stream API.
2. What is Stream API?
3. Tasks and objectives.
4. Example use.

Module 13. Multithreading

1. Multithreading in Java.
2. The use of ExecutorService.
3. Case study.

Module 14. Version control systems

1. What is version control?
2. Why do we need version control?
3. Review of version control systems.
4. Git.
5. Usage of external services (github).

Module 15. Teamwork, software project management

1. What is software project management?
2. Causes of occurrence of software project management discipline.
3. Gantt charts.
4. Important issues in software project management.
5. Analysis of terms of the topical area.
6. Project characteristics.
7. Costs associated with the project.
8. General review of models and methods of development process.
9. About Scrum in details.

Module 16. Usage of jUnit

1. What is unit testing?
2. Tasks and objectives of unit testing.
3. The need for unit testing.
4. Overview of tools used for unit testing.
5. The jUnit tool.

Module 17. Projection patterns

1. What are projection patterns.
2. Reasons of projection pattern occurrence.
3. The concept of projection pattern.
4. Principles of applying projection patterns.

5. Principles of selecting projection patterns.
6. Principles of separating patterns into categories.
7. Introduction to UML.
8. Usage of UML when analyzing projection patterns.
9. Creational patterns.
10. Design patterns.
11. Behavioral patterns.

Module 18. The MVC pattern

1. What is MVC pattern?
2. Tasks and objectives of the Model-View-Controller pattern.
3. Model.
4. View.
5. Controller.
6. Example use of the MVC pattern.

Module 19. Principles of designing SOLID classes

1. Review of issues occurring at design and development of classes.
2. Design principles of SOLID classes.
3. Example use of SOLID principles.

Module 20. Exam

Database Theory

Version 1.0.0

Course duration: 16 double-classes

Course objective

Introduction to the database management system (DBMS). Teach the student SQL (structured query language); the principles of normalization; working with stored procedures, triggers, views, and user-defined functions. Gain theoretical and practical knowledge about DBMS.

Upon completion of the course, the attendee will:

- have a good understanding of SQL;
- be able to create multi-table queries;
- understand the principles of operation of sub-queries and aggregate functions;
- be able to carry out the database normalization;
- use the stored procedures, triggers, views, and user-defined functions.

Upon completion of this course, the student submits a practical task and takes a theoretical exam on course materials. For admission to the examination, all home and practical tasks must be submitted. Practical task should cover a maximum of material from different sections of the course.

Topic Plan

Module 1. Introduction to the database theory	2 double-classes
Module 2. SELECT, INSERT, UPDATE, and DELETE queries	2 double-classes
Module 3. Multi-table databases.....	2 double-classes
Module 4. Aggregation functions.....	2 double-classes
Module 5. Joinings	2 double-classes
Module 6. Views, stored procedures, triggers.....	4 double-classes
Module 7. Exam	2 double-classes

Module 1. Introduction to the database theory

1. Introduction to the database theory.
2. History of the Oracle DBMS.
3. Architecture of the Oracle DBMS.
4. Versions of the Oracle DBMS.
5. Utilities.
6. Demonstration: Oracle DBMS installation.
7. Architecture of a database being under the control of Oracle.
Comparison with other DBMSs.
8. Demonstration.
9. Practice: creation of a database using the configuration file.
10. Basics of interaction with Oracle.
11. Queries.

Module 2. SELECT, INSERT, UPDATE, and DELETE queries

1. The SELECT operator.
2. IN, BETWEEN, and LIKE keywords.
3. The INSERT operator.
4. The UPDATE operator.
5. The DELETE operator.

Module 3. Multi-table databases

1. Abnormalities in the interaction with a single-table database.
2. Principles of creation of a multi-table database.
3. Multi-table queries.

Module 4. Aggregation functions

1. Aggregation functions.
2. The concept of grouping. The GROUP BY keyword.
3. The HAVING keyword. Comparative analysis of HAVING and WHERE.
4. Subqueries.

Module 5. Joinings

1. Operators applied in subqueries.
2. Joining the results of a query.
3. The JOIN unions.

Module 6. Views, stored procedures, triggers

1. The concept of transaction. The use of transactions.
2. Views.
3. Stored procedures.
4. Triggers.

Module 7. Exam

Creating Web-pages Using HTML5 and CSS3

Version 3.0.0

Course duration: 14 double-classes

Course objective

To teach the student to develop and create layout of static web pages using the HTML5, CSS3 technologies. To present a full picture about the technological chain of creating web sites and to form understanding of trends of web technologies development. To teach the student to pick the most suitable way to develop web pages. To teach them to test and check the code of web pages.

Upon the completion of the course, the student will be able to:

- know and apply HTML basics: tags, attributes, and way of structuring contents of web pages in order to create formatted documents;
- know and apply CSS basics: values, lists, colors, fonts, and other formatting metrics;
- have skills of checking and debugging the code of web documents;
- have skills of formatting contents of web documents for various screens: from standard browsers to mobile devices;
- have skills of fast and quality formatting of complex web documents;
- know HTML5 and CSS3 basics.

Upon the completion of this course, the student submits all practical tasks of the course. The student is graded based on all submitted tasks. The obligatory practical task submitted at the end of the course, which the student works on at home: create a web site and put it on the Internet. Basic requirements: box layout, valid code.

Topic Plan

Module 1.	Introduction to the Web technologies. Structure of HTML. Text formatting with HTML	2 double-classes
Module 2.	Formatting with CSS. Lists. CSS indents and fields	3 double-classes
Module 3.	Graphics in web design. Graphics power. Hyperlinks. Navigation principles of the web site	3 double-classes
Module 4.	Tables.	2 double-classes
Module 5.	Positioning. Executing layout of Web pages by means of blocks	2 double-classes
Module 6.	Forms. Frames	2 double-classes

Module 1. Introduction to the Web technologies. Structure of HTML.

Text formatting with HTML

1. Introduction to the subject.
2. Introduction to markup languages. Extensible Hypertext Markup Language (HTML).
3. Tags as a basic element of the HTML structure. The rules of writing tags and their attributes in the HTML5 standard through the example of the ``, `<i>`, `<u>`, ``, `<sup>`, `<sub>`, and `
` tags. Syntax differences of HTML4, XHTML, and HTML5.
4. Main mistakes when writing tags.
5. Structure of the HTML5 document.
6. Character encodings of a page and `<meta>` tags.
7. Classification of tags: inline and block.
8. Text formatting model: headings and paragraphs. The `p`, `<h1>..<h6>` elements. Text alignment within inline elements: the `align` attribute.
9. Classification of tags: logical and physical tags.
10. Practice: creating a basic web page.

Module 2. Formatting with CSS. Lists. CSS indents and fields

1. Cascading Style Sheets (CSS).
2. Tags containing no formatting: `<div>` is a block tag; `` an inline tag.
3. Analogy of HTML and CSS by the example of inline and block tags.
4. The use of class and id attributes to set styles.
5. The use of the external CSS style files.
6. Practice: text formatting by means of CSS.
7. Creating lists.
8. Creating nested lists.
9. Formatting lists using CSS.
10. Definition lists: `<dl>`, `<dd>`, and `<dt>` elements.
11. Controlling indents and fields.
12. Practice: creating lists.

Module 3. Graphics in Web design. Graphics power. Hyperlinks.

Navigation principles of the web site

1. Graphics file formats in Web.
2. The `` tag and its attributes (src, alt, width, height, border).
3. Background of a page: the background property.
4. General information about hyperlinks.
5. Absolute and relative addressing.
6. Creating a menu by means of the structure of lists (``, ``), its formatting. The display property. Converting a link into a block element.
7. Pseudo classes.
8. The cursor property of CSS.
9. Practice: working on the development of an image gallery.
10. Properties from CSS3.
11. Working with multimedia.

Module 4. Tables

1. Creating a basic table. The `<table>`, `<tr>` and `<td>` tags.
2. Joining cells: `colspan` and `rowspan` attributes.
3. Tags of the logical table structuring: `<thead>`, `<tbody>`, `<tfoot>`.
4. Tags of logical column grouping: `<colgroup>`, `<col>`.
5. Table border control: `frame` and `rules` attributes.
6. Practice: creating complex tables.
7. Basics of creating table layout. Example of creating table layout: its disadvantages.

Module 5. Positioning. Executing layout of Web pages by means of blocks

1. The position property.
2. The visibility and overflow properties.
3. Practice.
4. Basics of the box layout. Rules of creating layout.
5. Considering the simplest structures of pages.
6. Responsive structure. Blocks with negative margin.

Module 6. Forms. Frames

1. Introduction to forms.
2. Controls of forms.
3. Developing forms using HTML.
4. HTML5 form elements.
5. Form validation using HTML5.
6. Formatting form elements using CSS.
7. Frames and their structure (theory).

Development of Client-Side Scripts Using JavaScript

Version 2.0.0

Course duration: 16 double-classes

Course objective

To teach the student to develop client-side scripts using JavaScript. To teach them to pick appropriate mechanisms and structures to solve a particular problem.

Upon the completion of the course, the student will:

- master basic structures of JavaScript, such as variables, conditionals, loops, strings, arrays, functions, etc.;
- get acquainted with OOP and its basic terms;
- be able to handle errors;
- be knowledgeable in the concepts of event, event handler;
- create handler functions of various events;
- understand differences between BOM and DOM;
- be able to interact with objects from BOM and DOM;
- be knowledgeable in subtle details of implementing client-side scripts for various browsers;
- master the principles of creating forms and analyzing user data with regular expressions;
- be able to store user data using cookie mechanism;
- understand subtle details of applying HTML5 in relation to JavaScript;
- be able to serialize and parse data using JSON;
- master principles of creating asynchronous requests using Ajax.

Upon the completion of this course, the student submits all practical tasks of the course. Grades for the subject are issued based on all submitted tasks.

Topic Plan

Module 1. Introduction to JavaScript.....	2 double-classes
Module 2. Object. Arrays. Array object. Strings. String object. Date object. Math object. Introduction to the object-oriented programming (OOP).....	2 double-classes
Module 3. Processing events	2 double-classes
Module 4. Browser Object Model. Document Object Model	2 double-classes
Module 5. Forms	2 double-classes
Module 6. Form validity verification. The use of Cookie	2 double-classes
Module 7. Canvas drawing, support of media	2 double-classes
Module 8. JSON, Ajax	2 double-classes

Module 1. Introduction to JavaScript

1. Client-side scripts.
2. What is JavaScript?
3. Origin of JavaScript.
4. Differences between JavaScript and Java, JScript, ECMAScript.
5. JavaScript versions.
6. The concept of Document Object Model.
7. The concept of Browser Object Model.
8. Implementation into HTML documents. JavaScript code editors.
9. The <noscript> tag.
10. Basics of syntax.
11. Variables. Variable name rules.
12. Data types.
13. Operators.
14. Data input/output. Dialog boxes.
15. Conditions.
16. Loops.
17. What is a function?
18. The arguments object.
19. Variable scope. The 'this' keyword.
20. Recursion.

Module 2. Object. Arrays. Array object. Strings. String object. Date object. Math object. Introduction to the object-oriented programming (OOP)

1. Objects.
2. Arrays.
3. Strings.
4. Delays and intervals. Periodic function call.
5. Date object. Processing date and time.
6. Math object. Properties and methods. Random numbers.
7. What is OOP?
8. Three fundamental principles of OOP.

9. The concept of class and object in terms of JavaScript.
10. Properties.
11. Methods.
12. Property accessors.
13. Constructor.
14. The concept of prototype.
15. Inheritance.

Module 3. Processing events

1. What is an event?
2. What is an event handler?
3. Processing events in scripts.
4. Controlling styles of Web page elements.
5. The event object and its properties.
6. Default event handlers (standard handlers), call restriction of a standard handler.
7. The Image object. Controlling the drawings and rollovers.

Module 4. Browser Object Model. Document Object Model

1. What is Browser Object Model?
2. Objects of Browser Object Model.
3. What is Document Object Model?
4. Differences between DOM and BOM.
5. Representing HTML documents as a tree.
6. DOM objects. Hierarchy of nodes.
7. Properties and methods of DOM. The DOM event model.
8. Changing the DOM tree.
9. Introduction to the Document and Link objects.
10. Managing the highlight and text range: Selection and TextRange objects.
11. Features of DOM in HTML5.

Module 5. Forms

1. Application of forms. Arrangement of form elements in HTML.
2. The Forms collection. Developing and programming elements of form.

Module 6. Form validity verification. The use of Cookie

1. The RegExp object. The rules of recording regular expressions.
2. Methods of the String and RegExp objects for handling regular expressions.
3. Verification of validity of the form data.
4. What is a cookie?
5. Advantages and disadvantages of cookie.
6. Creation, use, and removal of cookie.

Module 7. Canvas drawing, support of media

1. What is canvas?
2. Basic features.
3. Support of media.

Module 8. JSON, Ajax

1. What is JSON?
2. Objectives and tasks of JSON.
3. The JSON syntax.
4. The JSON object.
5. Configuring the user serialization in JSON. The toJSON method.
6. Synchronous and asynchronous requests.
7. What is Ajax?
8. The XMLHttpRequest object.
9. Methods and properties of XMLHttpRequest.
10. The concept of the HTTP header.
11. The use of the GET method. URL encoding.
12. The use of the POST method.

Development of Back-End Solutions Using Java

Version 2.0.0

Course duration: 58 double-classes

Course objective

To teach them to pick appropriate mechanisms and structures to solve a particular real-world problem.

Upon the completion of the course, the student will be able to:

- understand fundamentals of creating back-end solutions using Java;
- create, compile, and debug web applications;
- interact with data sources;
- use network mechanisms;
- create servlets;
- understand and apply the MVC pattern;
- create JSP solutions;
- use Spring framework;
- use Hibernate library.

Upon the completion of this course, the student submits practical task and takes test of knowledge on course materials. In order to be admitted for the test, all home tasks and practical tasks must be submitted. Practical task must cover a maximum of the material from various sections of the course.

Topic Plan

Module 1. Introduction to network technologies	2 double-classes
Module 2. Network interaction	4 double-classes
Module 3. Introduction to developing back-end solutions using Java	6 double-classes
Module 4. Interaction with data sources	6 double-classes
Module 5. JavaServer Pages, Tags in JSP.....	8 double-classes
Module 6. Introduction to Spring	20 double-classes
Module 7. Introduction to Hibernate, Spring Data.....	10 double-classes
Module 8. Exam.....	2 double-classes

Module 1. Introduction to network technologies

1. What is network and server-side programming?
2. The objectives and tasks of the network and server-side programming.
3. What is a network?
4. Types of networks.
5. The OSI model.
6. Basic terms.
7. Client and server interaction pattern.

Module 2. Network interaction

1. The overview of the java.net package.
2. The InetAddress class.
3. The Socket class.
4. The ServerSocket class.
5. The DatagramSocket class.
6. The DatagramPacket class.
7. Practical project. Creating a file server.

Module 3. Introduction to developing back-end solutions using Java

1. Introduction to server-side programming.
2. Brief overview of useful utilities and libraries.
3. The concept of servlet.

Module 4. Interaction with data sources

1. Data sources.
2. JDBC.
3. Working with JDBC.

Module 5. JavaServer Pages

1. What is JSP?
2. Objectives and tasks of JSP.
3. Origin of JSP.

4. The concept of directive.
5. Error handling in JSP.
6. JSP and Model View Controller.
7. Expression Language in JSP.
8. JavaBean.
9. Java Standard Tag Library.
10. Different types of Tags.
11. The use of Conditional Tags.
12. The use of Iteration Tags.
13. The example use of different Tags.
14. What is Custom Tags?
15. What is Tag Files?
16. What is JSP Fragment?
17. The examples use.

Module 6. Introduction to Spring

1. What is Spring?
2. Objectives and tasks of Spring.
3. Origin.
4. The Spring architecture.
5. REST and SOAP
6. Spring MVC.
7. The Spring MVC architecture.
8. The examples use.
9. Spring Boot.
10. Spring Security.
11. Spring Data.
12. Microservice architecture.

Module 7. Introduction to Hibernate, Spring Data

1. What is Hibernate?
2. Aims and tasks of Hibernate.

3. Origin.
4. The Hibernate architecture.
5. The examples use.

Module 8. Exam