Правительство Российской Федерации

Федеральное государственное автономное учреждение высшего профессионального образования

«Национальный исследовательский университет «Высшая школа экономики»

Факультет Бизнес-информатика

Отделение Программная инженерия

Программа дисциплины

"Современное программирование"

(Advanced Programming, the discipline is provided in English)

для направления 231000.68 "Программная инженерия"

подготовки магистра

Автор программы

к.т.н. Е.М.Гринкруг

egrinkrug@hse.ru

Рекомендована секцией УМС
по бизнес-информатике

Председатель _____

«_____ » _____ 2013 г.

Одобрена на заседании кафедры
Управления разработкой программного
обеспечения (УРПО)

Зав. кафедрой С.М. Авдошин

« ____ » _____ 2013 г.

Утверждена Ученым Советом факультета
Бизнес-информатики

Ученый секретарь _____

« ____ » _____ 2013 г.

Москва - 2013

I. Summary

Author of the Program:

Assoc. Prof., PhD. Efim M. Grinkrug (responsible lecturer)

General Information about Training Course:

The course is offered to students of the Master Program **«System and Software Engineering** (SSE)» (code 231000.68) under chief subjects *«Methods and Theory of Software Engineering))* and *(Management of Software Developments* in the School of Software Engineering, Faculty of Business Informatics of the National Research University - Higher School of Economics (HSE).

This optional course is aimed to minimize students diversity in their programming abilities when entering the master program after having been educated in different Universities in Russia and abroad. External magister program students at the beginning of their study have programming skills and experience that are significantly different from those of the students who have graduated from NRU HSE, software engineering department. The goal of the course is to minimize that difference.

It is a one module course, which is delivered in module #1 of the first academic year with one lecture and one seminar class per week with intensive homework expected.

Prerequisites:

The course is based on the knowledge of foundations of discrete mathematics (including logic and set theory), computer science, and computer programming. Students are expected to be familiar with programming principles, applied to the one of the well known programming languages, and be familiar with one software development environment, at least, when using their favorite programming language.

Course Objectives:

The objective of Advanced Programming course delivery is to form professional competencies related to practical software development using the most popular object-oriented programming language and correspondent environment – Java. Currently, Java-programming is the most widely used in software research projects and in software industry world-wide, and ability to develop software in Java is required for successful education at the master program. Since master students who have got their bachelor degree in software engineering from NRU HSE before entering the master program have already proved their skills and experience this course is aimed to provide, the course for them is optional.

Abstract:

The course is based on the course "Object-oriented Analysis and Programming" delivered at the bachelor program of the Software Engineering department (by the same author).

The course is aimed for students who have no experience in modern java programming when entering magister program. The goal of the course is to learn java programming language to an extent that enables students to participate successfully in studying all other courses where practical software development skills are needed.

Training Objectives:

The nature of the course is that it is highly dependent on the students involved: the whole approach to the course is based on students diversity aligning, and the diversity can not be predicted in advance.

After taking this course the student should have achieved the following objectives:

Be able to develop Java programs using modern development tools to the extent that corresponds to the Java Standard Edition related programming basics, with selectable specific technologies.

Be able to participate in software development and programming exercises required at other courses during their master program studies.

Be able to implement practical home works, assignments and course works using Java-programming language, when required.

II. Topic-wise Curricula Plan

The plan of the course is highly dependent on the students' initial knowledge, skills and programming abilities, since the students are entering the course after their graduation from different Universities from different countries. That diversity must be taken into account when developing the schedule of the course.

The content of the course corresponds to the topics, listed below:

1. Introduction to Java programming and Development Environment.
2. Base programming structures in Java.
3. Objects and Classes.
4. Inheritance and Polymorphism.
5. Interfaces and Inner classes.
6. Collections and Generic programming.
7. Parallel programming. Multithreading.
8. Event-driven programming.
9. Graphics programming and GUI development.
10. Debugging, testing and deploying Java-software.
11. Overview of modern Java-programming technologies.

III. Base Textbook(s) and Recommended Readings

- Bruce Eckel. Thinking in Java. Fourth Edition. Prentice Hall, 2006
- Cay S. Horstmann, Gary Cornell. Core Java 2, vol.1 Fundamentals, Eighth Edition, 2007.
- Java Developers Kit Documentation.
  http://www.oracle.com/technetwork/java/javase/documentation/index.html
- Joshua Bloch. Effective Java. Second Edition. 2008.

IV. Software Tools and Environments

For successful learning the discipline student should use the following software (except the University standard software installation):

- Java Developer Kit (JDK) for Java Platform Standard Edition
  http://www.oracle.com/us/technologies/java/standard-edition/overview/index.html

- IntelliJ IDEA IDE (http://www.jetbrains.com/idea/) - the tool is used as the main development instrument; students are granted the free of charge academic license for the whole learning year(s).

V. Forms of Control

Current control: attendance record, seminar-based knowledge control, group project control;
Final control: exam at the end of module 1;

- The final course grade is a sum of the following elements:

1) attendance record (A);
2) practice activities (S);
3) written test (T);
4) exam (E).

The overall and accumulated course grades Go and Ga (10-point scale each) are calculated as follows:

Ga = 0.15A + 0.15S + 0.7T;

Go = 0.6Ga + 0.4E.

The overall and accumulated course grades Go and Ga (10-point scale each) include results achieved by students in their attendance record A, practice activities S, written test T and exam E; it is rounded up to an integer number of points. The rounding procedure accounts for students' practice activities during seminars. Intermediate assessment retakes are not allowed. Conversion of the concluding rounded grade (FE) to five-point scale grade is done in accordance with the following table:

Summary Table : Correspondence of ten-point (10) to five-point (5) system's marks

Ten-point scale [10]	Five-point scale [5]
1 - unsatisfactory **2** - very bad **3** - bad	Unsatisfactory ***(UnS)*** **- 2**
4 - satisfactory **5** - quite satisfactory	Satisfactory **(S) - 3**
6 - good **7** - very good	Good **(G) - 4**
8 - nearly excellent **9** - excellent **10** - brilliantly	Excellent **(E) - 5**

The author of program: _____ E.Grinkrug.