

**Федеральное государственное автономное образовательное учреждение  
высшего образования  
"Национальный исследовательский университет  
"Высшая школа экономики"**

Факультет компьютерных наук  
Департамент программной инженерии

**Рабочая программа дисциплины  
Основы разработки мобильных приложений  
(на английском языке)  
Mobile Applications Development: Basics**

для образовательной программы «Системная и программная инженерия»  
направления подготовки 09.04.04 «Программная инженерия»  
уровень - магистр

Разработчик программы  
Макаров С.Л., к.т.н., доцент, [smakarov@hse.ru](mailto:smakarov@hse.ru)

Одобрена на заседании департамента программной инженерии «\_\_»\_\_\_\_\_ 2016 г.  
Руководитель департамента Авдошин С.М. \_\_\_\_\_

Утверждена Академическим советом образовательной программы  
«\_\_»\_\_\_\_\_ 2016 г., № протокола \_\_\_\_\_

Академический руководитель образовательной программы  
Александров Д.В. \_\_\_\_\_

Москва, 2016

*Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы*



## 1 Course Description

### a. Title of the Course

The course title is Mobile Applications Development: Basics.

### b. Pre-requisites

Studying of the "Mobile Applications Development: Basics" discipline is based on the following courses:

- Programming,
- Introduction to Software Engineering,
- Functional and Logic Programming,
- Data Bases,
- Program Verification,
- Quality Provision and Testing,
- Operating systems,
- Software Constructing,
- Algorithms and Data Structures,
- Foreign language (English).

### c. Course Type

The course is compulsory.

### d. Abstract

The course introduces methods, platforms and tools for professional mobile applications development. It is based on the training materials (tutorials) from mobile apps developers for the following platforms: IBM Bluemix, Apple iOS, Google Android, Microsoft Windows Phone, Microsoft Azure, Amazon Web Services, Adobe PhoneGap, etc. The course is aimed at studying the major current approaches to development of mobile applications. Topics include basics of mobile application development, debugging, logging and testing applications, using emulators and more. The course is divided into 3 parts considering 3 major ways of mobile applications development. There are 15 laboratory exercises with the same assignments for each student: 5 for each of the 3 approaches of mobile applications development.

## 2 Learning Objectives

The discipline goal is to give students basic skills of the 3 basic ways of mobile application development using different platforms, techniques and software which eventually help students get the skills necessary for successful mobile application development.

Topics to be learnt and tasks to be solved to achieve the goal are:

- getting to know different types of mobile operating systems;
- differentiating between different types of environments used to develop mobile applications;
- meeting the special features of mobile application development;
- being aware of the 3 basic ways of mobile application development;
- studying at least 3 mobile application environments (IDEs) considering the 3 approaches;
- studying extra techniques and ways of mobile application programming;



- getting skills in developing and testing of mobile OS application by solving different tasks during the course studies.

### 3 Learning Outcomes

As a result of studying the course a student is supposed:

*to know and be able to use:*

- basic concepts, approaches and techniques of mobile application development;
- different software for doing that (at least 3 environments (IDEs) according to the 3 ways of mobile application development);
- different mobile platforms architecture;
- special features of mobile application development.

*to be able to:*

- develop and test a mobile OS application;
- search, install and use the latest versions of at least 3 environments (IDEs) for mobile application development;
- make changes of mobile application programming code, user interface and understand ideas and code of example applications.

As a result of studying the course student develops the competencies shown in table 1.

Table 1 - Student competence description

Competence	Key	Descriptors	Forms and techniques of the study which lead to developing the competence
Being able to analyze, verify, evaluate fullness of the information found and retrieved from different sources during professional activities and add and synthesize lacking information if needed.	SC*-M6 (6)	Student shows ability to analyze, verify and evaluate the information given considering developing a mobile application and find the lacking information pieces if needed using different resources.	Attending lectures, working on homework tasks, attending laboratory works, communicating with colleagues and searching and studying information needed using internet resources during the self-study sessions.
Being able to create principally new ideas and products, being creative, initiative.	SPC*-M8 (PC-8)	Student shows ability to create an idea and put it into a new mobile application which is then developed using skills taken during the laboratory assignments.	Attending lectures, preparing and solving different homework tasks, attending laboratory works, searching and studying information needed using internet resources.
Being able to work in a multidisciplinary team including international environment.	SPC-M9 (PC-9)	Student shows ability to communicate in an international environment and to explain his ideas to other team members, to share work needed to be done between colleagues and to be responsible for his/her particular part of the work.	Attending lectures, preparing and solving different homework tasks, attending laboratory works and working as a group, working on the project in a team, searching and studying information needed using internet resources.
Ability to plan, manage and control meeting requirements.	PC*-13	Student is able to develop a requirement set for the mobile application being developed and fulfill them.	Attending lectures, preparing and solving different homework tasks, attending laboratory works, searching and studying of information



Competence	Key	Descriptors	Forms and techniques of the study which lead to developing the competence
			needed using internet resources.
Ability to apply up-to-date software development technologies using automated systems of planning and management, ability to fulfill quality control of the software being developed.	PC-18	Student is able to control and follow the software development plan and can evaluate a quality of the developed software at any time so as to make changes if necessary.	Attending lectures, preparing and solving different homework tasks, attending laboratory works, searching and studying of information needed using internet resources.

\* SC - system competence, SPC - socially personal competence, PC - professional (instrumental) competence.

## 4 Course Plan

Table 2 - Course topics

№	Topic	Total hours	Classroom activities		Self-study
			Lectures	Practice	
1.	Introduction. Mobile applications development features. 3 approaches to building mobile applications: native, web, hybrid.	15	2	3	10
2.	Natives: Android platform. Introduction, history, architecture, Java interfaces review, Android application structure, components.	15	2	3	10
3.	Natives: Android platform. IDEs and software. Android Virtual Device, Android SDK Manager. Downloading, installing and using Eclipse IDE and Android Studio IDE. Eclipse and Android Studio: project structure, creating the first application.	15	2	3	10
4.	Natives: iOS platform. Introduction, history, architecture, Objective-C, Swift, simulator, iOS SDK. Downloading, installing and using XCode and macOS. Installing the software onto PC platform. Virtual machines. XCode: project structure, creating the first application.	15	2	3	10
5.	Natives: Windows platform. IDEs and software. Emulator, Windows SDK. Downloading, installing and using Microsoft Visual Studio. Project structure, creating the first application.	15	2	4	9
6.	Web (HTML5) IDEs: review. Working with Adobe PhoneGap: creating the first cross-platform application in 5 steps.	15	2	4	9
7.	Hybrids: tools and frameworks review. Working with Xamarin. Project structure, creating the first basic application using Xamarin.	25	2	3	10



8.	Emulator services: review, enhanced speed and different approaches of speeding up AVD. The Genymotion project.	15	2	3	10
9.	Intel XDK IDE: review, features, abilities, installing and using. Project structure, creating the first basic application with Intel XDK.	15	2	4	10
10.	Marmalade SDK: review, features, abilities, installing and using. Project structure, creating the first basic application.	15	2	3	9
11.	Cloud services review: Google Cloud, IBM Bluemix, Microsoft Azure, Amazon Web Services and their features for developing mobile applications.	15	2	4	20
12.	Marketing and publication of mobile application in the store. Mobile store review: Google Play, Apple App Store, Windows Store, Amazon Store. Preparing and publishing an application in Google Play.	15	2	3	9
Total:		190	24	40	126

## 5 Reading List

### 5.1 Required

1. Android Application Development in 24 Hours, Sams Teach Yourself (4th Edition) [Text] / Carmen Delessio, Lauren Darcey, Shane Conder. - SAMS, 2015. - ISBN-13: 978-0672337390. ISBN-10: 0672337398.
2. Neil Smyth. iOS 9 App Development Essentials: Learn to Develop iOS 9 Apps Using Xcode 7 and Swift 2 [Text] / CreateSpace Independent Publishing Platform, 2015.
3. Microservices from Theory to Practice. Creating Applications in IBM Bluemix Using the Microservices Approach. – First Edition. [Text] / IBM: Redbooks, 2015.
4. A Beginner's Guide to Mobile Marketing (Digital and Social Media Marketing and Advertising Collection) - Kindle Edition [Text] / Karen Mishra, Molly Garris. - Business Expert Press, LLC, 2015. - ISBN-13: 978-1-60649-841-5 (e-book).- Collection ISSN: 2333-8822 (print).
5. 3 Approaches to Building Mobile Apps: Which Is Best? [Electronic Resource] / Paul Rubens, December 26, 2013. – URL: <http://www.enterpriseappstoday.com/management-software/3-approaches-to-building-mobile-apps-which-is-best.html>
6. HTML5, native and hybrid – choosing the best approach [Electronic Resource] / Dave Kearney, May 18, 2015. - URL: <http://blog.fluidui.com/html5-native-and-hybrid-choosing-the-best-approach/>
7. Microsoft Visual Studio [Electronic Resource] / Microsoft, 2016. - URL: <https://www.visualstudio.com/en-us/dn469161>
8. Google Android Developer Site [Electronic Resource] / Google Inc, 2016. - URL: <https://developer.android.com/samples/index.html>.
9. Getting Stated: Adobe PhoneGap [Electronic resource] / Adobe Systems Inc., 2016. – URL: <http://docs.phonegap.com/>



10. Xamarin [Electronic resource] / Xamarin Inc., 2016. – URL: <https://www.xamarin.com/>
11. Intel XDK [Electronic Resource] / The site authors, 2016. - URL: <https://software.intel.com/en-us/intel-xdk>
12. Marmalade SDK [Electronic Resource] / Marmalade Technologies Ltd, 2016. - URL: <https://www.madewithmarmalade.com/platform/technicaldetails>

## 5.2 Optional

13. Android Application Development All-in-One For Dummies, 2nd Edition [Text] / Barry Burd. - John Wiley & Sons Inc., 2015. ISBN-13: 978-1118973806. ISBN-10: 1118973801.
14. Android Programming: The Big Nerd Ranch Guide (2nd Edition) [Text] / Bill Phillips, Chris Stewart, Brian Hardy, Kristin Marsicano. - Big Nerd Ranch, LLC, 2015. - ISBN-13: 978-0134171456. ISBN-10: 0134171454.
15. Head First Android Development (1st Edition) [Text] / Dawn Griffiths, David Griffiths. - O'Reilly Media, Inc., 2015. - ISBN-13: 978-1449362188. ISBN-10: 1449362184.
16. Android Application Development Cookbook - Second Edition [Text] / Rick Boyer, Kyle Mew. - Packt Publishing, 2016. - ISBN-13: 978-1-78588-619-5.
17. App Quality: Secrets for Agile App Teams [Text] / Jason Joseph Arbon. - Jason Arbon, 2014.- ISBN: 9781499751277.
18. iOS Forensics Cookbook [Text] / Bhanu Birani, Mayank Birani. - Packt Publishing, 2016. - ISBN: 978-1-78398-846-4.
19. Resources of Apple Developer Website [Electronic Resource] / Apple Inc, 2016. - URL: <https://developer.apple.com/resources/>
20. IBM Bluemix Website [Electronic Resource] / IBM, 2016. - URL: <https://console.ng.bluemix.net/>
21. Microsoft Azure Website [Electronic Resource] / Microsoft, 2016. - URL: <https://azure.microsoft.com/en-us/>
22. Amazon Web Services Website [Electronic Resource] / Amazon Web Services, Inc. or its affiliates, 2016. - URL: [https://aws.amazon.com/?nc1=h\\_ls](https://aws.amazon.com/?nc1=h_ls)
23. Google Cloud Platform Website [Electronic Resource] / Google Inc, 2016. - URL: <https://cloud.google.com/>
24. PhoneGap [Electronic resource] / Adobe Systems Inc., 2016. - URL: <http://phonegap.com/>
25. Eclipse IDE Site [Electronic Resource] / Developers of the site. 2016 - URL: <http://www.eclipse.org/>
26. Android Studio IDE Site [Electronic Resource] / Google Inc., 2016. - URL: <https://developer.android.com/develop/index.html>
27. Java SE Site [Electronic Resource] / Oracle Corporation, 2016. - URL: <http://www.oracle.com/technetwork/java/javase/overview/index.html>
28. Google Developer Console Website [Electronic Resource] / Google Inc., 2016. - URL: <https://console.developers.google.com>.
29. Official Google developers blog [Electronic Resource] / various authors and Google Inc., 2016. - URL: <http://googleblog.blogspot.com/search/label/Android>.
30. Official Android Blog [Electronic Resource] / various authors and Google Inc., 2016. - URL: <https://android.googleblog.com/>
31. Genymotion Emulator Site [Electronic Resource] / The site authors, 2016. - URL: <https://www.genymotion.com/>
32. XCode 8 Beta [Electronic Resource] / Apple Inc., 2016. - URL: <https://developer.apple.com/xcode/>
33. Swift 3 [Electronic Resource] / Apple Inc., 2016. - URL: <https://swift.org/>



34. Stackoverflow [Electronic resource] / Question-Answer blog platform with different users as authors. Site design and logo by Stack Exchange Inc., 2016. - URL: <http://stackoverflow.com/>
35. AppCode IDE [Electronic resource] / JetBrains, 2000 - 2016. - URL: <http://www.jetbrains.com/objc/>
36. Hackintosh [Electronic resource] / Hackintosh.com, 2016. - URL: <http://www.hackintosh.com/>
37. Apple Life forum [Electronic resource] / forum platform with different users as authors (language - russian)., 2016. – URL: <https://applelife.ru/>
38. Thinking in Java (4th ed.) [Text] // Bruce Eckel. - Prentice Hall, 2006.

### 5.3 Dictionaries, wiki

39. Objective-C [Electronic resource] / Wikipedia – free encyclopedia. Different users as authors, last modified on September 14, 2016. – URL: <http://en.wikipedia.org/wiki/Objective-C>
40. Swift [Electronic resource] / Wikipedia – free encyclopedia. Different users as authors, last modified on September 17, 2016. – URL: [https://en.wikipedia.org/wiki/Swift\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language))

### 5.4 Software needed for students

Students need the following software installed to successfully study the course:

- Browsers: the latest versions of the Chrome, Opera, Firefox.
- MS Word 97 or higher / Apache OpenOffice 3 (OpenOffice.org Writer) or higher.
- JDK 1.8 (Java SE 8.x).
- Android SDK.
- Android Studio 2.1.
- Eclipse Neon or Juno.
- Adobe PhoneGap.
- Xamarin.

## 6 Grading System

Current control is evaluated as following:

$$O_{current} = 0.6 \cdot O_{HA} + 0.4 \cdot O_T$$

The homework assignments (HA) mark is

$$O_{HA} = 10 \cdot (1/15) \cdot \sum (O_{curi}), i=\{1, \dots, 15\}$$

where

$$O_{curi} = \{0, 1\}$$

The written test (T) mark is

$$O_T = 2 \cdot O_{answer}$$



where

$$O_{answer} = \{0,1\}$$

So for example if a student has passed all the 15 laboratory (home) assignments and answered correctly all the 5 test questions the  $O_{current} = 10$  (using the 10 grade scale).

The final control is evaluated like

$$O_{final} = 2 \cdot O_{answer}$$

where

$$O_{answer} = \{0,1\}$$

If a student answered 4 questions of 5 total he gets 8 mark (using the 10 grade scale).

And the most final "result" mark of a student knowledge which goes to the master diploma is evaluated according to the following formula using the 10 grade scale:

$$O_{result} = 0,6 \cdot O_{current} + 0,4 \cdot O_{final}$$

## 7 Guidelines for Knowledge Assessment

Table 7.1 - Forms of student knowledge control

Type of control	Form of control	The whole year / modules				Parameters**
		1	2	3	4	
Current	Laboratory assignments/exercises	*	*			There are 15 (laboratory) assignments concerning different topics of the course. Passing an assignments means to show it working on mobile phone/emulator and to be able to answer any questions regarding the code of the assignment script and theory. Each student gets the same assignments. Usually it takes 10 minutes to evaluate solving of each assignment. For more information check section 7.2 of this document.
Current	Test	*				Written test concerning different lecture topics about mobile application development.
Final	Examination		*			Oral examination consisting of 5 questions

\*\* Parameters specify a control format: written, oral, computer testing, timing, and what exactly a student should do to pass, etc.

### 7.1 Knowledge and skill estimating criteria

Current control consists of 15 laboratory assignments and 1 written test. Each assignment is evaluated as 1 if it has been successfully passed and as 0 if it was not (1 grade scale). The test is evaluated using the 5 grade scale. Each correct answer means getting 1 grade.

There is no intermediate control.

Final control means passing the examination and is evaluated using the 5 grade scale. Each correct answer means getting 1 grade.

The 5 grade scale used for written test and final control consists of 5 grades: 0 (failed: nothing was done, no questions answered), 1 (1 question was answered), 2 (2 questions answered), 3 (3 questions), 4 (4 questions) and 5 (5 questions, excellent answers).

As long as the National Research University Higher School of Economics has the 10 grade scale system all the results above are then transformed into the 10 grade scale as written in the 6 section of the document.

## 7.2 Current control sample tasks

The current control consists of 15 laboratory assignments.

Laboratory assignments basically match the lecture topics and consist of 5 actual assignments for 3 different environments or frameworks according to the 3 different approaches of developing a mobile application: native, web (HTML5) and hybrid. So, for example, the following 5 assignments should be done in Android Studio, Adobe PhoneGap and Xamarin (the choice of IDEs and environments is up to student):

1. Develop a "Hello, World!" application. The application consists of 1 activity which contains 1 TextView containing the "Hello, World!" text, but in order to develop the application student needs to find, download and install all the necessary development tools.

2. Create a new project and develop an application which sets the TextView element as an output of the EditText input text element after pressing a Button element of the interface. Also the activity should contain TextView of the author of the application and the name of the student group. Start the project using the emulator and ensure everything works.

3. Create a multiactivity application. First activity contains TextView with a number or a name of the activity, EditText element for entering some text and a Button with "Next" or "Go to the 2nd activity/window/screen" text on it or just "2" as the Button text. Also there should be the TextView with the name and the group of the author. After pressing the Button element the 1st activity changes to the second. The second activity contains TextView with a number of a name of the activity, TextView with the text like "You've entered the following:" and with another TextView under this one which text is set to the first activity EditText input text value. Also the second activity should have a Button element to return to the first activity with the "1" or "Back" or "Go back to the text editing screen/window/activity" text on it. After pressing this button a user is sent to the first activity. Start the project using the emulator and ensure everything works.

4. Create an application which contains optional and context menus. Selecting different items of the menus should change something on the screen - change size of interface elements, their color, make some of them invisible and visible again, clean an EditText element, etc.

5. Create your own user list. For example, get access to the phone contact list using Permission changes in AndroidManifest.xml file and then copy the contact list into your own list and show it to the user. Or create a list like the twitter formatted message containing a picture (avatar) and some text on the right of the picture. Elements can be set up as a static array. Elements of the list are clickable: after a click (tap) on the list item user can see a Toast message which contains the item name (text).

## 7.3 Sample questions for the final control

1. Special features of mobile application development.
2. 3 ways of mobile application development.
3. What are the tools for mobile applications development?



4. How to put your application onto a market (Google Play, App Store)?
5. How to create a simple application using XCode?
6. What is a hybrid application?
7. What cloud systems do you know for developing mobile applications?
8. What kind of applications can you develop using Adobe PhoneGap?
9. What are the special features of Intel XDK?
10. What is a permission for mobile application?
11. What is activity?
12. Describe Android OS architecture.
13. Who or what is Dalvik?
14. How to install macOS on PC platform?
15. Describe different types of layouts.
16. What is the manifest file of a mobile application for?
17. Describe Android application Eclipse project structure.
18. What are the names of the simplest interface elements used to develop an application according to the laboratory assignment #3?
19. How to configure an emulator in Android Studio?
20. What is the name of the latest version of Android?
21. How to create the apk file for your mobile application?
22. What is dp? Sp?

## 8 Methods of Instruction

Classes of the course are conducted as the following forms:

- lectures/master classes/coding sessions of the computer presentation format;
- discussing various questions rising during the lectures, discussing them via email;
- practice (laboratory assignments in classrooms);
- self-studies with the help of the lecture presentations, software needed, internet, literature sources.

### 8.1 Recommendations for the lecturer

Lectures should be given in a presentation form which means a report followed by info- or just graphics shown with the help of presentation software and a computer connected to a projector and internet. Experience confirms that the presentation lectures are the most effective way to teach students during the course. The lecturer should consider giving lectures on how to make the laboratory assignments with several examples - how it should look and work and how it should not.

Each laboratory assignment is made by students on the grounds of the task itself and the according lectures which are given to students as a presentation software file if necessary. Each assignment passing is preparation to pass the exam. If student have questions considering the assignments the questions should be asked right away during the lecture. If they are not asked, the lecturer still should ask students if they have any questions regarding the lecture or assignments given.

When the lecturer tests students who are trying to pass any type of the controls mentioned above it is necessary for the lecturer to pay attention at the way each student tells about his work and answers the questions being asked about the code, lectures, etc. Questions should not be repeated from student to student. If a student can hardly answer some questions or the assignment result does not work or raises errors it is crucial to explain and help student about the code errors and how to avoid them, and give correct answer to the question asked with maximum amount of information and explanations (of course excluding the exam case), otherwise the student will keep coming with the same mistakes again and again.



## 8.2 Recommendations for the students

Students have to attend lectures because all the problems while working with the software, programming languages, ways and techniques of building and analyzing mobile applications, all laboratory assignments as long as examination questions are described and given in the lectures and coding sessions.

Implementation of each assignment should be different for each student. Having the same assignments doesn't mean the same interface including the application UI element position, styles and more because people are different.

Do not try to pass an assignment u have nothing to do about - it is obvious for the lecturer, and you know lecturers do not like copycats much.

Do not learn by heart all the answers to all the question to prepare for the exam because 1) all you need to do to prepare for the exam is to attend lectures and listen carefully; 2) there is no guarantee you will be given the questions you have learnt by heart (there are always some extra questions); 3) there are additional exam questions which are very possible including the assignment questions. That is why good work during all the course and lectures especially is a high exam mark guarantee.

## 9 Special Equipment and Software Support

Computer classes with MS Windows XP/7/8/10 and section 5.4 software - for the practice.

Classroom with good quality wi-fi or VGA/HDMI projector (not less than 1024x768 px) and Microsoft PowerPoint 2002 or higher installed on a laptop/notebook along with section 5.4 software - for giving lectures.