

# Development of a Complex of Practical Works on Programming Technology in the Android Studio Environment

Irina Pugacheva, Oxana Nass

**Abstract** — Currently, mobile devices are widely used in many areas of human life. Thus, mobile applications have found application in such industries as robotics, logistics, navigation, cartography, etc. Therefore, training students in this area is of great importance. In this regard, the relevance of the topic is determined by the need for a theoretical and methodological substantiation of the peculiarities of studying the basics of programming mobile devices, as well as developing a set of practical works on programming technology in the Android Studio environment.

The aim of the study is to develop a theory and methodology, as well as a set of works for practical training of students in the field of developing programs for mobile devices using the Android Studio environment.

It is supposed to use the following research methods: theoretical analysis of scientific and technical literature on the research topic, as well as national and international standards in the field of programming mobile applications; programming in the Android Studio environment; processing, analysis, generalization and approbation of research results.

The research process uses an integrated software development environment and Android Studio 4.0.

Practical work is aimed at mastering by students the skills of using programming technologies, which are significant for increasing the efficiency of mastering the basics of programming mobile applications and their further free use..

**Keywords**— learning, programming, android.

## I. INTRODUCTION

Over the past decade, mobile devices have evolved, opening up new ways of learning called mobile learning. Leveraging self-study using technology that is already available to many students is the goal of mobile learning apps. Students of the 21st century are digital savvy learners, most of whom spend countless hours on their mobile devices and game consoles. They rely heavily on smartphones for both communication and entertainment. This is an important consideration when designing a learning application.

Mobile learning is made effective by such key features of mobile learning as: time and place-independent individual learning; collaboration with peers and teachers in both formal and informal settings; interactivity and ubiquity of mobile devices. GUI (Graphical User Interface) is gaining more importance than ever because mobile learning needs to make the best use of limited screen space. Building mobile apps for learning is a complex area of research.

In this article, we present and discuss the creation of a mobile application of a set of practical works on programming technology in the Studio environment. Learning programming is perceived by many students as a difficult subject to master. This is due to a misconception that students think that writing code requires a deep understanding of abstract concepts as well as complex cognitive efforts. We can now argue that mobile learning can stimulate student motivation and interest in learning by providing them with challenging and contextualized learning. Students can learn at their own pace and anywhere using any device with the Android operating system.

## II. ANDROID STUDIO ENVIRONMENT

Android Studio is a new and fully integrated application development environment (IDE) released by Google for the Android operating system. This product aims to provide developers with new tools for creating applications, as well as provide an alternative to Eclipse, which is currently the most popular development environment. Main features - the possibility of layout in real time is implemented, many options for screen sizes and resolutions are available, and much more.

Android Studio lets you see any visual changes that can be made in real time in the application. You can also see how the application will simultaneously look on different Android devices, with different settings and screen resolutions. The product also features new packaging and coding tools. This will prevent you from getting lost in the project when you are dealing with a lot of code. The program also includes a drag-and-drop function that allows you to move components through the user interface. On top of that, the new development environment has Google Cloud Messaging, which allows data to be sent from the server to Android devices via the cloud. This is a great way to send push notifications to apps. It is also possible to localize applications using the program. This will allow you to program while maintaining control of the application.

What else does Android Studio offer?

- reliable and simple development environment;
- Easily check the performance of the application on different types of devices;
- helpers and templates for common Android programming elements;
- full-featured editor with many additional tools to speed up application development.

Thus, the set of required programs is very small. Taken together, these programs provide a sufficient number of tools for building applications.

Google in February 2020 launched Android Studio 3.6, the latest version of its integrated development environment (IDE), with a particular focus on "solving quality problems, primarily in code editing and debugging use cases." This release is the first release since Project Marble, the fancy title of an initiative announced by Google late last year to improve core Android Studio features. Android 3.6 introduces a small set of new features, polishes existing features, and fixes common bugs and performance improvements.

*Android Studio 3.6 Features:*

*Split view in design editors.* Design editors such as the Layout Editor and Navigation Editor now provide a Split view that allows you to view the Design and Code views of the user interface at the same time. The split view replaces the preview window and can be customized on a per-file basis to preserve contextual information such as zoom factor and design presentation options so you can choose the view that best suits each use case.

*Color Picker Resources Tab:* It is now easier to apply colors that you have defined as color resources. The color picker now populates color resources in your application so that you can select and replace color resource values. The color picker is available in both the design tools and the XML editor. *View binding* makes it easier to write code that interacts with views, ensuring compile-time safety when accessing views in your code. When enabled, view binding generates a binding class for each XML layout file exposed in this module. In most cases view binding replaces `findViewById`, you can reference all views with ID without

risking null pointer exclusion or class casting. These differences mean that incompatibility between layout and code will cause the build to fail at compile time, not run time.

*Android NDK Updates:* Previously supported in Java, these features are now also supported in Kotlin. You can navigate from the JNI declaration to the corresponding C / C ++ implementation function (view this mapping by hovering over the C or C ++ element marker next to the line number in the managed source file). You can automatically create a stub implementation function for the JNI declaration (first define the JNI declaration and then enter “jni: or the name of the method in the C / C ++ file to activate).

*IntelliJ Platform Update:* The 2019.2 IntelliJ Platform release includes many improvements, from a new Service Toolbox to significantly improved startup times.

*Add Classes with Apply Changes.* You can now add a class and then deploy that code change to the running application by clicking Apply Code Changes or Apply Changes and Restart Activity.

*Android Gradle Plugin (AGP) updates:* Support for the Maven Publish Gradle plug-in that allows you to publish build artifacts to the Apache Maven repository. The Android Gradle plugin generates a component for each build variant artifact in a library app or module, which you can use to configure publishing to the Maven repository. In addition, the Android Gradle plugin has significantly improved annotation / KAPT processing performance for large projects - AGP now generates R-class bytecode directly instead of in .java files.

*New packaging tool:* The default packaging tool has been changed to zipflinger for debug builds. You should see an improvement in build speed, but you can also revert to using the old packaging tool by setting `android.useNewApkCreator = false` in your `gradle.properties` file.

*Android Emulator - Google Maps UI:* Android Emulator 29.2.12 provides a new way for app developers to interact with an emulated device location. The Google Maps user interface is built into an advanced menu of controls to make it easier to point to locations as well as create routes from pairs of locations. Individual points can be saved and re-sent to the device as virtual locations, and routes can be created by entering addresses or tapping two points. These routes can be played back in real time as locations along the route are sent to the guest OS.

Android Studio 3.6 also includes the usual performance improvements and bug fixes in addition to new features.

### **III. ANALYSIS OF EXISTING COMPLEXES OF PRACTICAL WORK, ALLOWING YOU TO LEARN HOW TO PROGRAM IN THE ANDROID STUDIO ENVIRONMENT**

School Android (<https://android-school.ru>)

Let's consider a set of practical works from the site "School of Android" (<https://android-school.ru/ustanovka-android-studio-emulator-hello-world>).

This site is a rather limited resource as it has only 5 lessons for learning Android Studio. The author of the site on the main page explains the current situation of this project: due to lack of time, it is not possible to create a high-quality collection of lessons on Android development for beginners.

The lessons contain material on the following topics:

1. Installing Android Studio. Creation of an emulator. Launch of the first project.
2. The structure of the Android Studio project. Gradle build system.
3. Layout of screens. Layout views.
4. RelativeLayout.
5. TextView, Button. Interacting with interface elements from Java code.

In general, the lessons are written in an accessible language, all explanations correspond to the actions and are displayed on the corresponding images. The only drawback is a very large amount of material on one page; it takes some time to comprehend one lesson. For easier understanding, it would be better to break one topic into several lessons.

There are also 3 articles on the site: "Moving from Retrofit 1 to Retrofit 2", "Google OAuth authorization through Google Play Services 9.2 (and higher)" and "Retrofit 2: Uploading files to the server". This material, in turn, allows you to more carefully work out some of the nuances of the capabilities of the Android Studio environment.

STARTANDROID (<https://startandroid.ru>)

Let's take a closer look at the complex of practical works "Start Android - Android tutorial for beginners and advanced" from the site "STARTANDROID".

This site is quite an extensive resource as it has 190 lessons for learning Android Studio. The author of the site in the first introduction lesson gives a description and principles of the work of his resource. The resource itself began to be developed back in 2011 and, accordingly, all the lessons were made for older versions of Android. Some of them have been fixed for new versions if there were any major differences between versions. The author himself warns about this and provides explanations of how to cope with problems during the training on this resource. Since the site contains a forum, you can always ask questions or read more about possible problems in the discussion thread of each lesson, which is an additional advantage of this resource. The last lesson dates back to 2017, which is also a disadvantage, since recently new versions have been released and I would like to see new lessons for their capabilities.

This resource, unlike some authors, does not seek to teach programming in "24 hours" or in "30 lessons".

The author tried to make each lesson as independent and detached as possible, so that you can go in, see the desired topic and not look at a bunch of unnecessary things. I tried to select the examples that most clearly reflect the topic of the lesson.

All lessons are written in a rather professional language, which is some difficulty in understanding for people who have not previously encountered programming in principle. Also, projects are developed in Java, and knowledge of SQL, XML and other related technologies may be required.

The advantages of this site include the fact that it can be read and studied not only in Russian, but also in English, all lessons are translated into this language in full, with all available descriptions of the work and the corresponding images.

As a result of the analysis of the complexes of practical work presented on the sites in the global Internet and allowing you to learn programming in the Android Studio environment, the following conclusions were made:

- a set of practical works "Android School (<https://android-school.ru>)" is written in clear and easy-to-understand language, but has few lessons;
- the complex of practical works "STARTANDROID (<https://startandroid.ru>)" is well and conveniently structured, and also has step-by-step instructions for completing tasks;
- however, the analyzed complexes "School Android (<https://android-school.ru>)", "STARTANDROID (<https://startandroid.ru>)" do not fully meet the requirements of the curriculum and catalog of elective disciplines of the specialty "Information systems".

Almost all existing sites involve training people who already have extensive programming experience. A fairly limited number of resources suggests learning from scratch, and they are all written in a rather complex professional language.

#### **IV. DEVELOPMENT OF THE CONTENT OF PRACTICAL WORKS ON PROGRAMMING TECHNOLOGY IN THE ANDROID STUDIO ENVIRONMENT**

The purpose of introducing a set of practical works in the course of study is to increase the effectiveness of training, interest in the subject, motivation to work in the classroom, as well as updating previously acquired knowledge from various subject areas and the ability to apply them to real life situations.

The use of the acquired skills contributes to the independent thinking of students, develops the ability to find and solve problems, attracting knowledge from different areas for this. Also, by programming the constructed different models, students will learn to predict the results and possible consequences of different solutions.

Developed practical works for teaching programming technology will be based on the use of the Android Studio environment.

Main goals:

- development of students' creative abilities and logical thinking;
- development of skills to build a hypothesis and compare it with the result;
- development of the ability to be creative in solving problems;
- development of the ability to apply knowledge from various fields of knowledge;
- development of the ability to express thoughts in a clear logical sequence, to defend their point of view.

To develop the content of practical work, the materials of the discipline "Development of mobile applications" were used for 2nd year students of the specialty "Information systems" [3]. In the approximate program for this discipline, 20 hours are allocated for practical work - this is 22% of the total time (90 hours) allocated for the discipline.

This table (Table I.) shows the topics of the practical sessions and the number of hours for each topic.

Practical work is aimed at mastering by students the skills of using programming technologies, which are significant not only for the formation of programming literacy, but also for increasing the efficiency of mastering the basics of programming mobile applications and their further free use.

Table I. The topics of the practical sessions and the number of hours for each topic

№	Topic of the practical session	Number of hours
1	Installing Android Studio	1
2	Android project structure	1
3	Graphical user interface	2
4	Working with correction elements in XML	1
5	Working with standard elements and indicators	1
6	Working with external data sources	2
7	Working with the Layout element	1
8	Working with service components	1
9	Using IntentService	2
10	Working with TextView, Button	1
11	Using BroadcastReceiver	1
12	Testing	2
13	Using findViewById	1
14	Using OnClickListener	1
15	Working with phone devices	2

Practical work should be built in accordance with the following principles:

- theoretical material, as a rule, is not singled out as a separate block, but is organically combined with practical activity;
- students' activities within the framework of these works are practice-oriented;
- in the process of educational activities, feedback is realized between the teacher and students; the teacher poses a problem to students, motivates them to solve these problems and further corrects their activities;
- each practical work is in most cases a logical continuation of the previous one;
- material, both theoretical and practical, is presented from simple to complex;
- at the beginning of training, an idea of the ultimate goal is formed - the creation of a project; students understand that they have to move consistently, step by step towards its implementation.

The types of control are assumed to be the following: intermediate control, which is achieved through the tasks that students solve independently, after which they demonstrate their solutions to the teacher; verification work; project activities.

#### ACKNOWLEDGMENT

This publication is the result of the project implementation ERASMUS+ ACeSYRI:  
Advanced Centre for PhD Students and Young Researchers in Informatics  
reg.no. 610166-EPP-1-2019-1-SK-EPPKA2-CBHE-JP



Co-funded by the  
Erasmus+ Programme  
of the European Union



**REFERENCES**

- [1] R. Alhassan, 'Mobile Learning as a Method of Ubiquitous Learning: Students' Attitudes, Readiness, and Possible Barriers to Implementation in Higher Education', *Journal of Education and Learning*, vol. 5, no. 1, 2016.
- [2] A. I. Wasserman, 'Software engineering issues for mobile application development', *ACM Transactions on Information Systems*, pp. 1–4, 2010.
- [3] C. D. Rodríguez and T. M. Cumming, 'Employing mobile technology to improve language skills of young students with language-based disabilities', *Assistive Technology*, vol. 29, no.3, pp. 161–169, Jul. 2017
- [4] Shtyrova I.A., Vishtak N.M., Remarenko S.A. USE OF A MOBILE APPLICATION FOR A UNIVERSITY OF ADDITIONAL EDUCATION // *Modern science-intensive technologies*. - 2019. - No. 2. - pp. 153-157.
- [5] Dercy, Lauren. Development of applications for Android devices. Volume 1. Basic principles / Darcy Lauren, Conder Shane. - SPB: Lori, 2014 . pp. 412.
- [6] Kalita, ON Distance learning Russian verbs of movement in a training course on RFL using mobile applications: Author's abstract. dis. Cand. ped. Sciences: 13.00.02 / O. N. Kalita; Grew up. University of Friendship of Peoples. - Ulyanovsk, 2016 . pp. 23.
- [7] Nigmatullina, E.A. etc. Programming. In 2 volumes.Vol. 1. : a textbook for students. universities / ed. N.I. Pack. - M.: Publishing Center "Academy", 2013. pp.- 272.
- [8] Robert I. V. Theory and methodology of informatization of education (psychological, pedagogical and technological aspects) / I. V. Robert. - M.: IIO RAO, 2010. pp – 356.
- [9] Smart technologies in higher education // Scientific and methodological conference "Smart technologies in education: a portrait of a graduate 2020". [Electronic resource]. URL: <http://library.fa.ru/exhib.asp?id=199#top>
- [10] Dannells, Dana. Controlled natural language generation from a multilingual FrameNet-based grammar / Dana Dannells and Normunds Gruzitis. In: *Controlled Natural Language, Lecture Notes in Computer Science*, Volume 8625, Springer, 2014, pp. 155-166 (Scopus un Web of Science).