

**MINISTRY OF DEVELOPMENT OF INFORMATION TECHNOLOGIES
AND COMMUNICATIONS OF THE REPUBLIC OF UZBEKISTAN**

**NUKUS BRANCH OF THE TASHKENT UNIVERSITY OF
INFORMATION TECHNOLOGIES
NAMED AFTER MUHAMMAD AL-XORAZMIY**

«COMPUTER ENGINEERING» faculty

«Computer engineering» department

Approved for defence

Head of department

R.Oteniyazov

2017 «__» _____

QUALIFICATION WORK

Subject: «Creating timetable Android mobile application based on client-server architecture»

Graduate:	Saginbaev Allayar Maxsetbaevich
Supervisor:	I.Sharapova
Reviewer:	Sh.Eshmuratov
Consultant:	A.Kadirov

Nukus 2017

TABLE OF CONTENTS

INTRODUCTION	3
CHAPTER 1. USING JAVA AND ANDROID SDK	5
1.1. History of Android programming	5
1.2. Android application structure	15
1.3. Integrated Development environments for Android.....	16
1.4. Client-server technology	26
CHAPTER 2. USING ANDROID STUDIO AND PHP FOR CREATING APPLICATIONS	33
2.1. Creating layouts and fragments	33
2.2. Working with http and AsyncTask	40
2.3. Server-side data processing with PHP	45
2.4. The problem of automated timetable generation	49
CHAPTER 3. CREATING MOBILE APPLICATION OF UNIVERSITY TIMETABLE	54
3.1. Installing and configuring Android SDK and Android Studio	54
3.2. Processing requests to server	61
3.3. Server-side processing	65
3.4. Building APK.....	67
CONCLUSION	70
REFERENCES:	71

INTRODUCTION

We live in the era of the development of mobile information technology. Every fifth of us goes online only from a mobile device, on average 5-7 applications we use constantly. Everything started with a simple phone book in the mobile phone, and in the last three years mobile developers have started to offer truly smart mobile programs that solve household and business problems for us.

A qualitative jerk in the creation of mobile applications was the launch of the production of smartphones with large touch screens. Their productive capacity allowed developers to create mobile applications of a new level. It is at this moment that native development is developing - the development of mobile programs in the language of a certain platform. New mobile technologies have allowed the development of applications that economically use the resources of the smartphone, while much more productive and more functional than its predecessors.

Mobile devices are equipped with increasingly powerful processors, which allows developers to use modern information technology to develop mobile applications. In other words, high technology has finally reached mobile development through the creation of powerful smartphones.

Now applications can analyze analytically information from several sources and help the user make decisions, control processes and solve other important tasks with the least time and analytical costs for himself. That in turn helps to optimize business processes, increase productivity and efficiency of decisions.

Experts argue that the further path of mobile development will be built specifically towards creating applications that are trained to process certain information and provide solutions to users based on their personal needs.

Applications will become much more powerful in terms of performance and less energy-intensive, thanks to the improvement of mobile devices themselves and the development of new methods.

The purpose of this final qualifying work is the development of a mobile application for the automated receipt of a training timetable. Today almost every

university student has his own smartphone. And the scope of delivery of the training schedule has so far remained one of the most lagging areas in the activity of the university. Very few universities provide timely and interactive delivery of the study schedule to students.

Therefore, **the relevance and actuality** of this is to use the most modern information technology to automate the activities of the dean's office and optimize the learning process.

The first chapter discusses the theoretical part of using Java languages to develop mobile applications. Also, the client-server technology is considered, as well as questions of sending http requests from mobile devices.

In the second chapter, the process of developing a mobile application Timetable Our University for Android, which allows online to receive a schedule of classes for students and teachers. In particular, the process of design development, query execution, and server processing of requests is considered. At the end of the chapter, the process of assembling the finished APK file is discussed.

In the conclusion section on final qualifying work we give a summary of our work, the reached results are described.

CHAPTER 1. USING JAVA AND ANDROID SDK

1.1. History of Android programming

The origins of the Android operating system are rooted in 2002. At this time, the creators of Google are interested in a set of software developments E. Rubin. Initially, the project to create a new OS for mobile devices under a great secret was engaged in the company Android Inc., which was later purchased by Google.

In November 2007, the long-awaited announcement of Android took place. But it turned out that Google went further - they created an alliance that brings together the leading manufacturers of mobile phones and software developers. This association called Open Handset Alliance includes companies that have access to the source code. That is why the first phones on the basis of this operating system of such leaders of the industry as HTC, Motorola, Samsung, Acer are now on sale. Google has wisely thought out a campaign to promote its new development.

The emergence of an operating system like Android was a welcome development in the IT environment. Therefore, during the past year, advanced users were eagerly awaiting the release of the first mobile devices based on Android.

This operating system for smartphones was developed on the Linux kernel and equipped with a modified Java-environment. Similar ideas were worked at the same time by competing companies, but Google rightfully attracted the public's attention to its product. The fact is that Linux has traditionally been intended only for work on desktop computers and laptops, that's why among IT-developers for a long time there was an idea about the possibility of creating its mobile version.

If the Android operating system meets the expectations of users and caused a stir, Google will be able to claim leadership in the field of mobile OS. Prior to this, smartphone manufacturers offered their own software platform development for their devices. Also on the mobile OS market without much success trying to break through the good old Windows. It would seem that such diversity should please consumers, but not everything is so simple. Due to differences in operating systems, it is difficult to launch some mobile applications, which, for example, can

work on Nokia and can not find contact with LG. If Android becomes the undisputed leader among mobile OS, in the coming years developers will write hundreds of new applications for this platform, and the user will not have such problems.

The Android platform is easily adaptable for use on VGA, 2D graphics libraries based on the OpenGL ES 1.0 [<http://developer.android.com>] specification, as well as traditional tools for smartphones. Android uses a database for structured data. Android supports a large number of technologies that provide communication, including: GSM, Bluetooth, EDGE, 3G and WiFi [<http://developer.android.com>].

To exchange messages, both SMS and MMS services are available, including streaming messages. On Android, a browser based on the WebKit application framework is available, and there is also the ability to run programs in Java. Android supports the following formats of audio / video data and images: MPEG-4, H.264, MP3, and AAC, AMR, JPG, PNG, GIF[5].

From non-standard equipment there is support for a video camera, camera, touchscreen, GPS, compass, accelerometer, 3D graphics accelerator[3]. On October 21, 2008, the OHA Alliance published the source code for the Android platform. The release included the entire Android stack: both the operating system, middleware, and the main end applications written in Java.

The first smartphone on the Android platform was the device T-Mobile G1 (HTC G1), the presentation of which took place on September 23, 2008. Soon followed by numerous announcements from other manufacturers of smartphones about the intention to release devices from Android.

Since the Android platform was very profitable for the creation of budget smartphones, Chinese companies decided to take advantage of this opportunity immediately after the publication of the source code for Android. As a result, as early as in 2008, numerous products were offered for sale, largely targeting the local market. In the future, this trend only continued.

Until the commencement of the activity in the Android showcase showed up and created, Linux-undertakings of the main world organizations makers of cell phones. In this way, some of them were quit, including the reason the best prospects of Android gadgets. In any case, this does not imply that improvement did not speak to any intrigue: a few organizations have depended on their Linux stage.

One such organization - American Motorola. It is the Motorola on two or three Japanese telephones was as of not long ago gave the lion's offer of offers of Linux-based gadgets. It is interested that the American organization did not position them like cell phones. The reason was the closeness of the OS, as it fiercely sounds for devotees of Linux.

To obstruct the capacity to introduce outsider programming, it was chosen to enhance unwavering quality of a creating stage, a notoriety which could "recolor" an acting mischievously program. Really, along these lines, the Motorola gadget takes a middle of the road position amongst telephones and cell phones.

Tragically, this approach brought the normal natural products for a drawn out stretch of time portable Linux and could contend with the Symbian OS and Windows Mobile. Against this foundation, the vision of Android designers are deserving of acclaim. The creation by the Motorola cell phones on the Linux stage turned even toward the start of this century.

They had a place with the Accompli arrangement A6188. At that point took after the model of the Accompli 008. In 2003 started the advancement of Linux-based Motorola gadgets to another level. They utilized a variant of the OS MontaVista CEE (Consumer Electronics Edition) Linux organization MontaVista Software.

Curiously, this working framework was expected for cell phones as well as for different family unit machines, and advanced gadgets. On the mass improvement of this bearing was reported in February 2003. Also, in December of that year demonstrate discharged Motorola A760. The machine showed up in the Asian market and was fascinating less a program stuffing, what number of

specialized elements. For instance, capacities, for example, Bluetooth, infrared, media player, touch screen (secured with dynamic top). The last situation added to the further fame of Linux-based Motorola gadgets in Asia, particularly in China, since it gave the chance to penmanship characters.

For quite a long while the organization routinely discharged Linux gadget with high usefulness and consistent touch screen. All of them were accessible just on the neighborhood Chinese market or notwithstanding falling on the counter. Furthermore, from 2005 started to show up the Linux demonstrate without the touch screen. By 2007, the organization has arranged the in a general sense new gadgets in view of the stage, MOTOMAGX.

Hence, due to the emergency in the organization, it was chosen to reduce additionally deal with MOTOMAGX items and to wager on the Symbian OS UIQ stage. One of the most recent Linux items, the organization was the Motorola VE66 slider. However UIQ gadgets did not spare the organization. In states of developing emergency, it was chosen to relinquish this stage, concentrating all endeavors on the generation... Android gadgets! Delegates of the new wave ought to be such gadgets as the Motorola DEXT.

Be that as it may, Motorola was by all account not the only maker of Linux-based gadgets. Such gadgets can be found in the lineup of different organizations, for example, Finnish Nokia. Not at all like Motorola items, they were distinctive in belief system of the gadget. Nokia 770, N800, N810 was totally without the GSM module. To make calls just utilizing VoIP utilizing WLAN association. Worked all the gadget running Internet Tablet OS, which spoke to not that other, as the Linux-based stage Maemo. It depends on Debian GNU/Linux with the 2.6 kernel.x. In this way, the name turned into the authority Maemo stage - witness the as of late reported N900, running Maemo 5.

Therefore, the intrigue was not just the equipment qualities of the gadgets on this stage, yet as a matter of first importance, the maker's way to deal with the working framework. The Maemo stage depended on free programming, however, all things considered, it was a weak arrangement, as a feature of the OS parts and

the quantity of uses was shut, not accessible in the source code. The rule was vital most importantly to make on a given stage local applications outsider designers - in some ways it was the harbinger of Android.

In spite of its scale, the pioneer, Google with its Android extend did not. Indeed, even before the principal reports about the new stage it was conceivable to take note of a few patterns in the improvement of versatile Linux, which are in this manner reflected in the advancement of Android-gadgets. It is essentially the union of endeavors of engineers. Numerous producers know about the destructiveness of splashing strengths to make the diverse variants of Linux.

Slowly they started to shape organizations together for the advancement of normal stages. It's entertaining, yet this pattern was again prompted discontinuity of endeavors. Prior to the appearance of the Alliance Open Handset Alliance driven by Google, there were a few such gatherings.

Besides, the enrollment did not keep a few organizations to take part in parallel private advancements. In November 2005 the association was established LiPS (Linux Phone Standards) Forum. Today it has 20 organizations. The most well known are the ACCESS (assimilated at the time, PalmSource), ARM, France Telecom, MontaVista.

The Alliance was shaped to build up a full standard working framework. Their endeavors LiPS composed with another gathering of engineers, OSDL (Open Source Development Labs). In January 2007, a Fund was set up LiMo (Linux Mobile) Foundation. The members likewise collaborated to make a bound together versatile Linux stage. The gathering included Motorola, Vodafone Group, NTT DoCoMo, Samsung Electronics, NEC, McAfee, LG Electronics and so on. Another pattern is the commercialization of portable Linux. The inclusion of vast partnerships has again scrutinized the standards of Open Source.

Along these lines, Alliance LiMo Foundation has chosen to give API to your framework on the states of permitting FPL, not free GPL. The source code was accessible for individuals from the LiMo. In the meantime, even before the Android advertise existed truly free, open to all activities. Among them I might

want to specify the organization Trolltech and FIC (OpenMoko). Items Norwegian Trolltech has since quite a while ago settled itself available. Numerous Linux gadgets running her posterity Qtopia Phone Edition.

The framework highlights a rich arrangement of uses and the accessibility of advancement apparatuses. For running most recent forms in August 2006 was submitted to the Linux-portable Greenphone. Gadget qualities were great: CPU Intel (Marvell) XScale 312 MHz, 64 MB RAM and 128 MB ROM, touch screen, Bluetooth, Wi-Fi. The product stage was Qtopia Phone Edition 4.1.4. The unit was provided and SDK for programming improvement.

A test group of gadgets was discharged in December 2006, yet was restricted. Overrated (\$700) reflected in the deals. Be that as it may, the organization did not consider Greenphone a business item, seeing it just as an equipment stage for engineers. It is not astounding that when Trolltech better equipment (a Linux cell phone OpenMoko Neo1973), the fundamental consideration was paid to this new venture. Accordingly, in October 2007, it was declared the end of advancement of the Greenphone.

Indeed, and later she Trolltech was obtained by the seller. A great deal more inspired by the destiny of OpenMoko (Open Moblie Kommunications). The venture organization FIC was planned to make monstrous Linux-based cell phone. The result of the Taiwanese organization FIC (First International Computer) Neo1973 had magnificent qualities. Counting a 2.8-inch VGA touch-screen, Samsung processor the s3c2442 400 MHz, SMedia 3362 representation quickening agent, 256MB of memory, Bluetooth, Wi-Fi, GPS-beneficiary.

It has been accounted for to take a shot at possess OpenMoko programming stage (Embedded Linux with portion 2.6.17). Joining with Trolltech has prompted the support of Qtopia Phone Edition. The expression "free" cell phone OpenMoko could be requested. Particularly when you consider the suggested cross-stage shell OpenMoko, the capacity to keep running on any appropriate gadget. We can state that the OpenMoko extend most completely was the precursor of Android, in which we see it now.

Android Features

To run applications, the Dalvik virtual machine is optimized for mobile phones. Files of type Java Class are converted by the usual Java compiler into classes of other format (.dex) for launching in Dalvik. For licensing reasons, Dalvik uses Apache Harmony instead of OpenJDK.

Table 1. Name of releases of all versions of Android

Version	Name	Year	Proportion
2.3.3 - 2.3.7	Gingerbread	2010	0,9%
4.0.3 - 4.0.4	Ice Cream Sandwich	2011	0,9%
4.1.x	Jelly Bean	2012	3,5%
4.2.x	Jelly Bean	2012	5,1%
4.3	Jelly Bean	2013	1,5%
4.4	KitKat	2013	20,0%
5.0	Lollipop	2014	9,0%
5.1	Lollipop	2015	23,0%
6.0	Marshmallow	2015	31,2%
7.0	Nougat	2016	4,5%
7.1	Nougat	2016	0,4%

Android OS Architecture

If you represent the component model of Android in the form of a hierarchy, then at the very bottom, the core of the operating system will be located in the very basis. It provides the functioning of the system and is responsible for security, memory management, power system and processes, and also provides a network stack and a driver model. The kernel also acts as an abstraction layer between the hardware and software.

The "higher" kernel, as the middleware, is a set of libraries (Libraries), designed to solve common problems requiring high efficiency. That is, this level is

responsible for providing implemented algorithms for higher layers, support for file formats, encoding and decoding information (for example, multimedia codecs can be cited), drawing graphics and much more. Libraries are implemented in C / C ++ and compiled for specific hardware devices, together with which they are supplied by the manufacturer in a pre-installed form.

At the same level is Android Runtime - the runtime of application programs. Its key components are a set of standard libraries and a Dalvik virtual machine. Each application in the Android OS runs in its own instance of the Dalvik virtual machine. Thus, all running processes are isolated from the operating system and from each other. The architecture of Android Runtime is such that the work of programs is carried out strictly within the environment of the virtual machine. Thanks to this, the kernel of the operating system is protected against possible harm from other components of the operating system. Therefore, the code with errors or malware can not spoil the Android OS and the device on its base. This protective function, along with the execution of program code, is one of the key for Android Runtime.

The level above is the Application Framework, sometimes called the application framework layer. It is through the application frameworks that developers get access to the API provided by the system components lying below the level. In addition, thanks to the architecture of the framework, any application is provided with the already implemented capabilities of other applications that are allowed to access.

The basic set of services and systems that underlie each application and are parts of the framework include:

1. A rich and extensible set of Views that can be used to create visual application components, such as lists, text fields, tables, buttons, or even an embedded web browser.
2. Content Providers, managing data that some applications open to others, so that they can use them for their work.

3. Resource Manager, which provides access to resources that do not carry code, for example, to string data, graphics, files and others.

4. Notification Manager, through which all applications can display their own notifications for the user in the status bar.

5. The Activity Manager, which manages the life cycles of applications, saves the history of work with actions, and provides a navigation system for them.

6. Location Manager, allowing applications to periodically receive updated information about the current geographic location of the device.

On the off chance that you speak to the segment model of Android as a chain of importance, then at the extremely base, the center of the working framework will be situated in the very premise. It gives the working of the framework and is in charge of security, memory administration, control framework and forms, and furthermore gives a system stack and a driver display. The bit likewise goes about as a deliberation layer between the equipment and programming.

The "higher" piece, as the middleware, is an arrangement of (Libraries), intended to take care of regular issues requiring high proficiency. That is, this level is in charge of giving executed calculations to higher layers, bolster for record organizations, encoding and interpreting data (for instance, sight and sound codecs can be referred to), drawing illustrations and considerably more. Libraries are actualized in C/C ++ and assembled for particular equipment gadgets, together with which they are provided by the maker in a pre-introduced shape.

At a similar level is Android Runtime - the runtime of utilization projects. Its key parts are an arrangement of standard libraries and a Dalvik virtual machine. Every application in the Android OS keeps running in its own case of the Dalvik virtual machine. Along these lines, every single running procedure are segregated from the working framework and from each other. The design of Android Runtime is to such an extent that the work of projects is completed entirely inside the earth of the virtual machine. On account of this, the portion of the working framework is shielded against conceivable damage from different parts of the working framework. Along these lines, the code with blunders or malware can not ruin the

Android OS and the gadget on its base. This defensive capacity, alongside the execution of program code, is one of the key for Android Runtime.

The level above is the Application Framework, at times called the application system layer. It is through the application structures that designers access the API given by the framework segments lying underneath the level. Also, because of the engineering of the structure, any application is given the officially actualized capacities of different applications that are permitted to get to.

The fundamental arrangement of administrations and frameworks that underlie every application and are parts of the system include:

1. A rich and extensible arrangement of Views that can be utilized to make visual application segments, for example, records, content fields, tables, catches, or even an inserted web program.

2. Content Providers, overseeing information that a few applications open to others, with the goal that they can utilize them for their work.

3. Asset Manager, which gives access to assets that don't convey code, for instance, to string information, design, records and others.

4. Warning Manager, through which all applications can show their own notices for the client in the status bar.

5. The Activity Manager, which deals with the life cycles of utilizations, spares the historical backdrop of work with activities, and gives a route framework to them.

6. Area Manager, enabling applications to occasionally get refreshed data about the current geographic area of the gadget.

At the highest point of the Android programming stack is the Application layer. This incorporates an arrangement of essential applications that Preinstalled on Android OS. For instance, it incorporates a program, an email customer, a program for sending SMS, maps, a schedule, a contact chief and numerous others.

The rundown of coordinated applications may shift contingent upon the gadget show and the adaptation of Android. Furthermore this essential set to the

application layer are all applications for the Android stage, including those introduced by the client.

Ordinarily, applications for Android are composed in the Java dialect, however it is conceivable to create programs in C/C++ (utilizing the Native Development Kit). Exotics can be known as the utilization of Basic (with the assistance of Simple) and different dialects. You can likewise make your own particular projects with the assistance of use creators, for example, App Inventor.

1.2. Android application structure

Android applications are composed in Java programming dialect. In any case it is imperative to recall that without utilizing the standard Java Virtual Machine (JVM) android applications can't be executed. Rather, Google has made a custom VM called Dalvik. Dalvik is in charge of Android is a product stack for portable contraption that contains a working framework, likewise a middleware and key application. The Android SDK give the apparatuses and APIs which are important to create application on the stage of Android innovation utilizing the java programming dialect. Android in light of Linux rendition 2.6.

The Android SDK gives a broad arrangement of use programming interfaces (APIs) that is both contemporary and safe. Android handset center framework administrations are open and approachable to all applications. At the point when given the suitable authorizations, Android application can share information among each other and get to shared assets on the framework safely

By giving an open advancement stage, Android propose build up the capacity to construct to a great degree rich and inventive application. Engineers are allowed to get advantage of the gadget equipment, get to area data, execute foundation administrations, add warnings to the status bar, and significantly more. Hidden all applications is resource of administrations and framework, including:

- A rich and extensible arrangement of Views that can be utilized to build up an application, containing records, matrices, content boxes, catches and even an embeddable web program.

- Content Provider that enables application to get to information from different applications, or to share their own information. x A Resource Manager, giving access to non-code assets which incorporates limited strings, design and format documents.
- A Notification Manager that empower all application to show custom alarms in the status bar.
- An Activity Manager that deals with the life cycle of utilization and gives a typical route backstack.

Android incorporates a cluster of center libraries that encourage a large portion of the usefulness accessible in center libraries of the Java programming dialect. Every Android application executes in its own particular procedure, when its own occasion of the Dalvik Virtual machine. Dalvik has been composed so that a gadget can execute different VMs in a proficient way.

The Dalvik VM execute record in the Dalvik Executable (.dex) design which is upgraded for slightest memory impression. The VM is resister-based, and run classes gathered by a java dialect compiler that have been change into the .dex arrange by the included "dx" apparatus. The Dalvik VM discharge on the Linux Kernel for major fundamental usefulness like threading and low-level memory administration.

1.3. Integrated Development environments for Android

May 15 at the conference Google I / O 2013 announced a new development environment for Android Studio. The basis for the new IDE was IntelliJ platform from JetBrains. This platform had many fans, so this news greatly rejoiced. At the moment this is only a beta version for those who like to experiment. You can download the new development environment at developer.android.com.

Main features - real-time imposition is realized, many options for sizes and screen resolutions are available. There is a help topic. Built-in tools to improve the quality of applications and monetization. There are tools for tracking the effectiveness of advertisements. The means of interaction with beta testers has been added. And much more.

The environment contains all the necessary elements for developing applications for the Android platform, testing programs for compatibility with different versions of the platform and designing applications for devices with different screen resolutions (tablets, smartphones, laptops, clocks, car infotainment systems, smart TVs [1]). The structure includes a source editor, a visual interface design environment, tools for debugging and code analysis, emulators, as well as a unified subsystem for assembling, testing and deploying applications, based on Gradle assembly tools and supporting the use of continuous integration tools.

Features:

- Instant start function, which allows to optimize the development cycle (editing / assembling / launching) by reducing the time required for the assembly. The system analyzes the changes made to the code and, if possible, integrates the changed components into the running process of the application, which allows you to observe the result immediately after making changes in the running application without the need to reinstall the APK.

Even if the nature of the changes does not allow you to do without restarting the application, the launch speed will be much higher due to the calculation of the optimal assembly plan and the caching of the unchanged resources of the virtual machine;

- A new emulator that runs three times faster than the version used in previous releases. In combination with improvements in the ADB tool, application and data delivery to the emulator can be performed up to 10 times faster than when deploying the application on a real device. The composition of the emulator is close to typical devices and contains built-in Google Play services, which allows testing additional APIs in it.

The emulator also has new tools for managing calls, battery charge, network connection and GPS. The emulator user interface has been modernized, which now supports window scaling, allows to simulate multi-touch screens and allows to move packets in drag & drop mode (for installation of the program you can simply drag the desired APK-packet with the mouse).

- Integration with the Cloud Test Lab service, allowing to evaluate the application performance on a wide range of different device models, using the service provided by the service to real devices, which allows to identify problems that are not manifested during simulation;

- Added support for the API Indexing API, through which you can declare a new program in the Google application search service. The development environment includes tools for automatically generating a suitable URL-indexing program and testing the indexing process.

- Experimental implementation of the GPU debugging system, which makes it easier to identify problems with rendering in games and applications that use OpenGL ES. The debugger intercepts and analyzes the workflows of the GPU, allowing you to monitor each frame and the state of the GL.

Java Package Architecture

The Java architecture for Android applications resembles the Model-View-Controller pattern. In Android, the fragments and activity represent the Controller classes. On the other hand, they are part of the user interface, so they are also part of the View.

Therefore, it is difficult to assign the fragments (or activity) uniquely to the Controller or the View. It's better to put them in your own fragments package. Activity in this case can be left in the top-level package. If you have more than two or three activities, you can take them out as a separate package.

On the other hand, the architecture can look like a regular MVC, with a package of models containing POJOs, generated using the JSON parser from the API responses, and a views package containing authored View, alerts, View classes associated with action bar, widgets, E. Adapters are the link between data and views. Given that they usually use View, exported via the getView () method, you can enable adapters as a child adapters package in views.

Some controller classes are used throughout the application and work directly with the Android operating system. They can be placed in the managers package. Different classes for processing data, such as "DateUtils", can be stored

in the utils package. The classes responsible for interacting with the backend are in the network package.

1.2.1. Overview of the programming environment

Before you start developing applications for Android, consider the existing tools suitable for this purpose. You can identify the necessary tools, without which the development of mobile applications for Android is simply impossible. On the other hand, there are a large number of auxiliary systems that to some extent simplify the development process.

Compulsory tools include the Android SDK - a set of programming tools that contains the tools needed to create, compile and build a mobile application.

Let us briefly consider the most important tools included in the Android SDK:

- SDK Manager - a tool that allows you to download the components of the Android SDK. Displays Android SDK packages and their status: Installed, Not Installed, Updates available

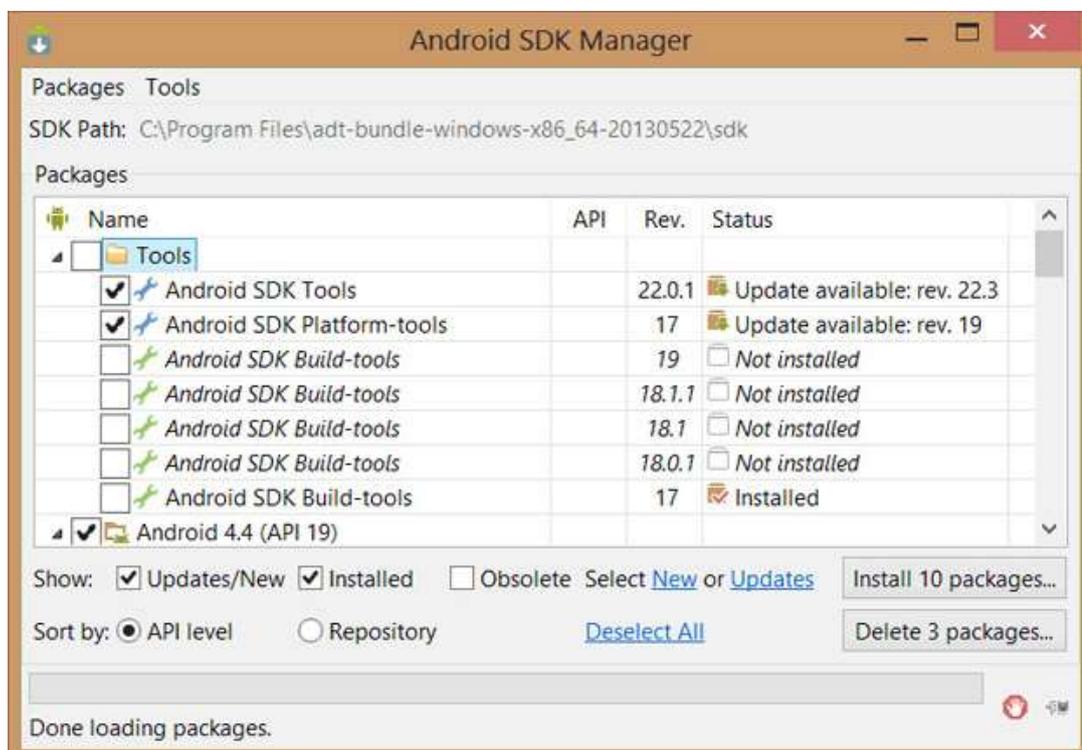


Fig. 1.1 Android SDK Manager

• Debug Monitor is a standalone tool that provides a graphical interface to several tools for analyzing and debugging Android applications:

○ DDMS (Dalvik Debug Monitor Server) provides port forwarding services, screen capture devices, information on the flow and dynamic memory of the device, output information about Android actions in real time (logcat) and much more.

○ Hierarchy Viewer allows you to debug and optimize the user interface of the Android application.

○ Tracer for OpenGL ES - a tool for analyzing OpenGL | ES code used in a mobile application, allows you to capture OpenGL | ES commands and display them on separate frames, which helps to understand how graphic commands are executed.

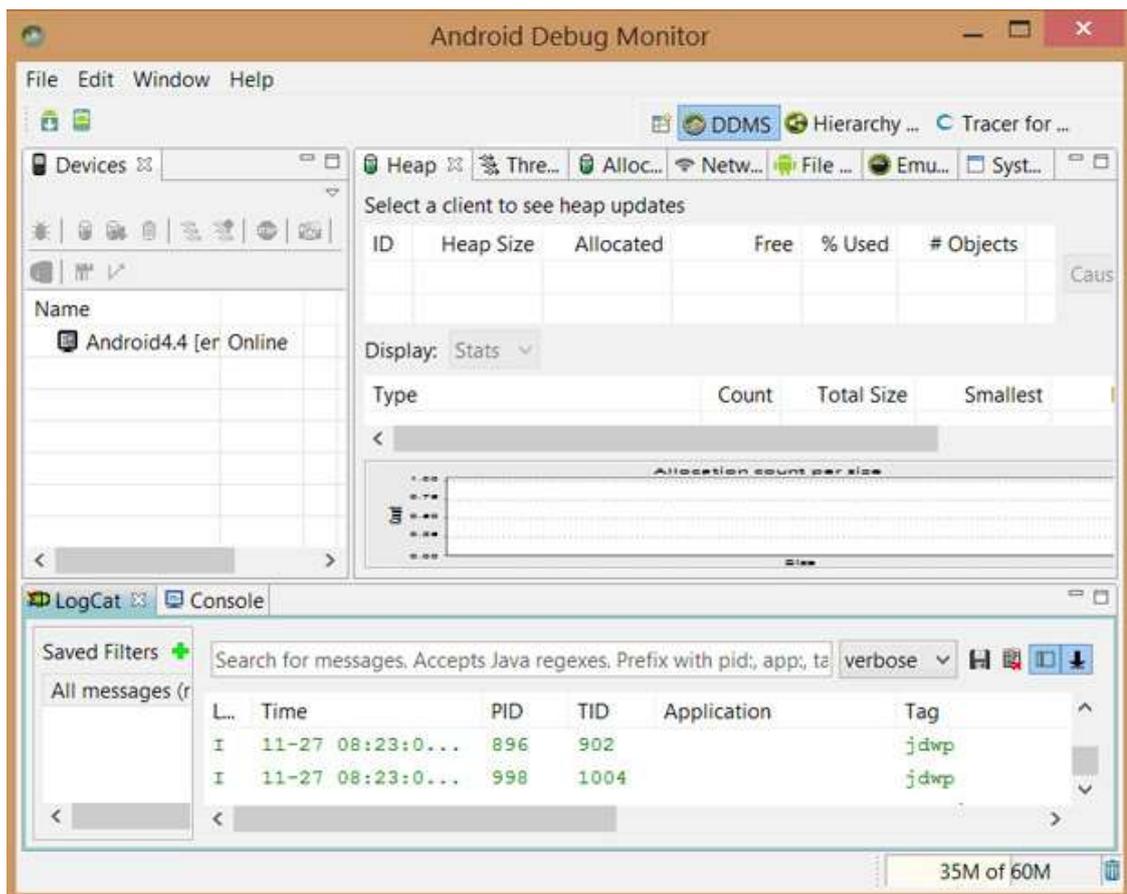


Fig. 1.2 The Monitor Tool window

- Android Emulator (emulator) - a virtual mobile device that is created and runs on the developer's computer, used to develop and test mobile applications without involving real devices.

- AVD Manager - provides a graphical interface for creating and managing virtual Android devices (AVDs), provided by Android Emulator.

(In LR # 1, the creation and use of a virtual device is discussed in detail).

- Android Debug Bridge (adb) - a flexible tool that allows you to manage the state of the emulator or real Android device connected to the computer. It can also be used to install an Android application (.apk file) on a real device.

We reviewed the main tools included in the Android SDK, of course, not all and not enough detail. For a more serious study of the tools, it makes sense to contact the developer's site (<http://developer.android.com/tools/help/index.html>). For the development of mobile applications for Android confident possession of tools from the SDK is enough. If you have any questions, additional instructions for creating projects, compiling, starting from the command line are contained in the manual from Google.

In modern conditions, software development is in most cases carried out using IDE .IDE have undoubted advantages: the process of compiling, assembling and launching an application is usually automated, and for a beginner, it is not necessary to create your first application for labor. But in order to develop seriously, it is necessary to spend the energy and time to study the possibilities of the environment itself. Consider the IDE, suitable for development under Android1.

First, let's talk about two development environments that Google recommends: Android IDE (ADT) and Android Studio.

Android IDE - the development environment for Android, based on Eclipse. Provides integrated tools for developing, building and debugging mobile applications. In this course, the Android IDE is selected as the primary development environment. The possibilities of this environment are considered in more detail in the first laboratory work. Also, there are recommendations for

installing and configuring the environment, creating and running the first application on both the emulator and the real device.

Android Studio is an Android development environment based on IntelliJ IDEA. Like the Android IDE, it provides integrated tools for development and debugging. In addition to all the features expected from IntelliJ, AndroidStudio includes:

- Support for building an application based on Gradle;
- Android-specific refactoring and quick fixing of defects;
- lint tools for finding performance problems, with usability, with version compatibility and others;
- ProGuard features (utility for reducing, optimizing and obfuscating the code) and application signatures;
- Template-based wizards for creating common Android designs and components;
- WYSIWYG editor, working on many screen sizes and resolutions, a preview window showing the running application on several devices and in real time;
- Built-in support for the Google cloud platform.

Now let's look at other tools that can be used to develop mobile applications for Android. Let's start with the tools from Intel - Intel XDK and Intel Beacon Mountain.

Intel XDK makes it easy to develop cross-platform mobile applications; Includes tools for creating, debugging and building software, as well as emulator devices; Supports the development for Android, Apple iOS, Microsoft Windows 8, Tizen[<http://developer.android.com>]; Development languages: HTML5 and JavaScript.

The last topic of this course is fully devoted to the study of a new generation of tools for developing mobile HTML5-applications and Intel XDK, it is expected to develop a mobile application using these tools.

Intel Beacon Mountain is a development environment that allows you to create applications for devices running the Android OS. Provides the tools necessary for designing, developing, debugging and optimizing applications under the Android. It frees the developer from the need to maintain the development system up to date, monitors the updates and adds them to the development environment as they become available. Supports development for target platforms based on Intel Atom and ARM processors.

Beacon Mountain is built on the basis of Android IDE (Eclipse, Android ADT, Android SDK), for more serious development and optimization, the following Intel tools have been added:

- Hardware Accelerated Execution Manager (Intel * HAXM) is a hardware-supported virtualization processor that uses Intel * virtualization technology (Intel * VT) to speed up the emulator in the development environment.

- Graphics Performance Analyzers (Intel * GPA) System Analyzer supports mobile devices with Intel Atom processor running Android OS. Allows developers to optimize system utilization when using OpenGL procedures, providing the ability to receive a variety of system metrics in real time, displaying the workload of CPU, GPU and OpenGL ES API. The developer can run several graphic experiments to identify bottlenecks in the processing of graphics.

- Integrated Performance Primitives (Intel * IPP) Preview - a library of optimized data and image processing that supports mobile devices with the Intel platform running Android. Preview version is part of the full version of Intel IPP, which also supports the Android OS.

- Threading Building Blocks (Intel * TBB) is a widely used, recognized C ++ template library for creating scalable applications and increasing productivity. Supports mobile devices with an Intel platform running Android. Proven algorithms allow developers to effectively parallelize C ++ mobile applications, which improves performance while reducing power costs.

You can not ignore the tools Marmalade SDK.

Marmalade SDK is a cross-platform SDK from Idea works 3D Limited. It is a set of libraries, samples, tools and documentation required for the development, testing and deployment of applications for mobile devices. Used mainly for game development. Many recognized games such as Cut the Rope and Plants vs. Zombies, were developed using this software tool. Unfortunately, the Marmalade SDK is a proprietary software (the cheapest license is \$ 15 per month) and can not be recommended in this tutorial, but the reader can independently try the free 30-day version available at <https://www.madewithmarmalade.com/Free-trial>.

It is impossible not to say about domestic developments. For example, 1C is keeping pace with the times, the version of the 1C 8.3 platform allows you to develop mobile applications. The software product "1C: Enterprise 8. Expansion for handheld computers" provides the ability to work with data from 1C: Enterprise 8 information bases on mobile devices (PDAs, communicators, data collection terminals), as well as on personal computers (including laptops), Who do not have direct access to the information bases of 1C: Enterprise 8.

Of course, this software product has a very narrow scope, but in some cases it can be the most successful solution.

1.2.2. Emulators

Emulation. Standard Android emulator

Emulation in computer technology is a complex of software, hardware, or a combination of them designed to copy (or emulate) the functions of one computing system (guest) to another computer system (host) than the first, so that the emulated Behavior as closely as possible corresponded to the behavior of the original system (guest). The goal is to reproduce the behavior as accurately as possible, unlike various forms of computer modeling, in which the behavior of some abstract model is simulated.

Emulator is a virtual mobile device that runs on a computer. With the help of an emulator, you can develop and test applications without using real devices. Figure 1.7 shows an example of a running standard emulator. Detailed work with emulators is considered in the laboratory.

The merits of using emulators include simplicity of their use and zero cost. The developer does not need to buy a huge number of devices with different characteristics to test the application on different smartphones. It is enough to create several emulators with the required characteristics and run an application on them.

Unfortunately, emulators also have a number of drawbacks:

- Require a lot of system resources.
- Due to differences in the architecture of the processors, the computer and the smartphone are slowly starting up. Modern personal computers are built on x86 and x64 architectures, and most smartphone processors on Android are ARM. The process of emulating one architecture on another is extremely complex and is rather slow.

- In some cases, a standard emulator is not enough. It's about the capabilities of smartphones that conventional computers do not have (for example, the presence of a gps sensor or an accelerometer). In such cases, full debugging can only be done using a real device.

Alternate emulators

The standard emulator, supplied with the Android SDK, does not suit many. There are projects that support the development and development of alternative emulators. As an example, you can bring Genymotion - a fast Android emulator (according to its developers). It contains pre-configured Android images (x86 with hardware acceleration OpenGL). Genymotion is available for Linux, Windows and Mac OS X and requires VirtualBox for its operation. In other words, Genymotion is a virtual machine with an installed Android OS, which the user runs just like other virtual machines. The problem of high consumption of system resources, of course, does not disappear, but the launch speed significantly increases.

Now actively developing.

Debugging features on real devices

The developed application can be launched on a real device, for example, on a smartphone. To do this, it is necessary to do preliminary work.

To run applications developed in the Android IDE, you must:

- Configure the device (enable USB debugging mode).
- Configure the computer (for Windows, you must install the correct driver manually, you need administrator rights).
- Configure the environment and run the project on the device.

1.4. Client-server technology

Often, programmers who start designing and implementing information systems run into a number of problems and complexities that arise not from their inexperience or lack of knowledge, but have their own causes and limitations inherent in software products that have a client-server architecture. Trying to solve these problems, they often have to "reinvent the wheel" rather than using high-tech and standardized means of creating distributed industrial information systems. Such tools include transaction monitors. Three models

"Client-server" is a model of interaction of computers in the network. It rarely happens that they are absolutely equal. Typically, one computer on the network has information and computing resources, such as processors, file system, mail service, printing service, database. Other computers use them. The computer that manages this or that resource is usually called the server of this resource, and the computer that wants to use it - the client. A particular server is characterized by the kind of resource that it owns. So, if the resource is a database, it is a database server whose purpose is to serve customer requests related to processing databases; If the resource is a file system, then talk about the file server or file server, etc.

The same principle extends to the interaction of processes. If one of them performs some functions, providing others with a corresponding set of services, such a process is treated as a server. The processes that use these services are usually called customers.

Today, the technology "client-server" is becoming more widespread, but in itself it does not offer universal recipes. It only gives a general idea of how a modern distributed information system should be organized. At the same time, the

implementation of this technology in specific software products and even in software types differ significantly.

One of the main principles of client-server technology is to separate the functions of a standard application into three groups of different nature. The first group is the input and display function. The second group combines purely applied functions, characteristic for the given subject area. Finally, the third group includes the fundamental functions of storing and managing data (databases, file systems, etc.)

In accordance with this, any application allocates the following logical components:

- a presentation component that implements the functions of the first group;
- application component (business application), supporting the functions of the second group;
- the resource access component (resource access) or the resource manager (Resource manager), which supports the functions of the third group.

Differences in the implementation of applications in the technology of "client-server" are determined by three factors. First, what mechanisms are used to implement the functions of all three groups. Thirdly, how the logical components are distributed among the computers on the network. Three approaches are distinguished, each of which is implemented in the corresponding model:

- Remote Data Access (RDA) model;
- Database server model (Database Server - DBS);
- The application server model (Application Server - AS).

In the RDA model, the presentation component and application component code are combined and executed on the client computer, the latter supports both data input and display functions and purely application functions. Access to information resources is provided, as a rule, by operators of a special language (SQL language, for example, when it comes to databases) or calls to functions of a special library (if there is an appropriate API). Requests for information resources are sent over the network to a remote computer (for example, a database server).

The latter processes and executes queries and returns the data blocks to the client. Speaking of the "client-server" architecture, in most cases this model is meant.

The DSB model is built on the assumption that the process running on the client computer is limited to the presentation functions, while the actual application functions are implemented in stored procedures, also called compiled resident procedures or database procedures. They are stored directly in the database and run on the database server computer (where the component that controls the access to the data, that is, the database engine) also functions. The concept of the information resource is narrowed down to databases, because the mechanism of stored procedures is a distinctive characteristic of the DBS model - there is so far only in the DBMS, and even not in all.

In practice, mixed models are often used when database integrity support and some simple application functions are supported by stored procedures (DBS model), and more complex functions are implemented directly in the application program that runs on the client computer (RDA model). However, such solutions, including elements of two models at once, can not fundamentally change our ideas about their relationship.

In the AS model, the process running on the client computer responds, as usual, to the input and display of data (that is, it implements the functions of the first group). Application functions are performed by a group of processes (application servers) running on a remote computer (or several computers). Access to the information resources necessary for solving application problems is provided in exactly the same way as in the RDA-model. Various types of resources are available from application components-databases, indexed files, queues, etc. Application servers are usually run on the same computer as the resource manager, but they can run on other computers.

What is the fundamental difference between these models? RDA- and DBS-models rely on a two-tier separation scheme. In the RDA-model, application functions are assigned to the client program, in DBS-model the responsibility for their implementation is assumed by the core of the DBMS. In the first case, the

application component merges with the presentation component, in the second case it is integrated into the access component to the information resources. In contrast, the AS-model implements a classic three-part function separation scheme, where the application component is identified as the most important element of the application, universal mechanisms of the multitasking operating system are used to define it, and interfaces with two other components are standardized. Actually, this advantage of the AS-model results from its advantages, which are of great importance for purely practical activity

RDA model

The main advantage of the RDA-model lies in the practical plane. Today, there are many tools that provide fast creation of desktop applications that work with SQL-oriented databases. Most of them support graphical user interface in MS Windows, ODBC standard, contain tools for automatic code generation. In other words, the main advantage of the RDA-model lies in the unification and wide choice of application development tools. The vast majority of these development tools in the fourth generation languages (including the means of programming automation) just create codes in which application functions and presentation functions are mixed.

The RDA-model has a number of limitations. First, the interaction of the client and the server through SQL-queries essentially loads the network. Since the application is unallocated and all its logic is localized on the client computer, the application needs to transmit a large amount of data over the network, possibly redundant. As soon as the number of customers increases, the network turns into a "bottle neck", hindering the speed of the entire information system.

Secondly, satisfactory administration of applications in the RDA-model is almost impossible. Obviously, if the functions different in nature (representation functions and purely application functions) are mixed in the same program written in 4GL, then if you need to change application functions, you have to rewrite the whole program. In the collective work on the project, as a rule, each developer is assigned the implementation of individual application functions, which makes it

impossible to control their mutual consistency. Each of the developers has to program an interface with the user, which calls into question the uniform style of the interface and its integrity.

DBS model

Despite the wide spread, the RDA-model gives way to a more technological DBS-model. The latter is implemented in some relational databases (Ingres, Sybase, Oracle). It is based on the mechanism of stored procedures - a tool for programming the database engine. Procedures are stored in the database dictionary, divided between several clients and executed on the same computer where the database engine is running.

The language in which stored procedures are developed is a procedural extension of the SQL query language and is unique to each specific DBMS. Attempts to standardize the SQL language regarding stored procedures have not yet led to tangible success. In addition, in many implementations the procedures are interpretable, which makes their execution slower than the execution of programs written in third generation languages. The stored procedure mechanism is one of the components of the active database server [<http://itc.ua/articles>].

In the DBS model, the application is distributed. The presentation component is executed on the client computer, while the application component (which implements the business functions) is designed as a set of stored procedures and functions on the database server computer. Advantages of the DBS model in front of the RDA-model are obvious: it is the possibility of central administration of business functions, and reduction of network traffic, and the possibility of separating the procedure between several applications, and saving computer resources by using a once-created plan for the procedure. However, there are disadvantages.

First, the tools used to write stored procedures, strictly speaking, are not programming languages in the full sense of the word. Rather, it's a variety of procedural SQL extensions that do not stand comparison with visual tools and functionality with third-generation languages such as C or Pascal. They are built

into specific DBMS, and, naturally, the scope of their use is limited. Therefore, the system in which the application component is implemented using stored procedures is not mobile relative to the DBMS.

In addition, most DBMSs do not have the ability to debug and test stored procedures, which turns them into a very dangerous mechanism. Indeed, any complex unfulfilled combination of triggers and triggering procedures can, according to the apt statement of one of the developers, "completely spread the entire database."

Secondly, the DBS-model does not provide the required efficiency of the use of computing resources. Objective constraints in the core of the DBMS do not yet allow organizing an effective load balance in its framework, migration of procedures to other database server computers and other useful functions. Attempts by DBMS developers to provide in their systems these capabilities (distributed stored procedures, requests with priorities, etc.) do not yet achieve the desired effect.

Third, decentralization of applications (one of the key factors of modern information technology) requires a significant variety of options for client-server interaction. When implementing an application system, you may need to have such interaction mechanisms as stored queues, asynchronous calls, etc., which are not supported in the DBS model.

Today, it can hardly be said that stored procedures in their current state represent an adequate mechanism for describing business functions and implementing an application component. In order to turn them into a really powerful tool, DBMS developers should reproduce in them the following features:

- Expanding the visual means of language procedures;
- tools for debugging and testing stored procedures;
- Prevention of procedural conflicts with other programs;
- Support for priority processing of requests.

Meanwhile, these capabilities have already been implemented in the AS-model, which reflects the strengths of the client-server technology to the greatest extent.

AS-model

The main element of the three-tiered scheme adopted in the AS-model is the application server. Within its framework, several application functions are implemented, each of which is framed as a service and provides some services to all programs that wish and can use them. Application servers can be multiple, and each of them provides a certain set of services. Any program that uses them is treated as an application client (Application Client - AC).

Details of implementation of application functions in the application server are completely hidden from the application client. AC addresses the request to a particular service, but not to AS, that is, application servers are depersonalized and serve only as a kind of "framework" for processing services, which allows you to effectively manage the load balance. Requests from the AS are queued to the AS process, which extracts and passes them for processing to the service in accordance with the priorities.

The AU is treated more widely than the component of the representation. It can support the interface with the end user (then it is a component of the view), it can provide data from some devices (for example, sensors), and, finally, itself can be AS. The latter allows implementing an application system containing AS of several levels. The architecture of such a system can look like a nucleus surrounded by concentric rings. The kernel consists of application servers that implement basic application functions. Rings symbolize sets of AS being clients in relation to lower-level servers. The number of server levels in the AS model, generally speaking, is unlimited.

It is not difficult to see that the AS-model has a universal character. A clear delineation of logical components and a rational choice of software for their implementation provide the model with a level of flexibility and openness that is

not yet achievable in RDA and DBS models. It is the AS model that is used as the foundation for a relatively new type of software for our users - transaction monitors.

Transaction Monitors

Transaction Processing Monitor (TPM), or, more simply, transaction monitors - software systems (which are often classed as middleware) that provide efficient management of information and computing resources in a distributed system. They are a flexible, open environment for the development and management of mobile applications, focused on the operational processing of distributed transactions. Among the most important features of TPM are scalability, support for the functional completeness and integrity of applications, maximum performance in data processing at low cost, maintaining data integrity in a heterogeneous environment.

CHAPTER 2. USING ANDROID STUDIO AND PHP FOR CREATING APPLICATIONS.

2.1. Creating layouts and fragments

In 2014, Google showed a new design and various new examples of navigation. But at first they used improvised means, which were at hand - fragments.

A year later, the company developed a new NavigationView component based on the FrameLayout ancestor, which became part of the Android Design Support Library.

And the approach was unexpected, but logical. Since the sliding curtain contains a navigation menu, the class was also designed as a menu. Ie, you create standard menu items in res / menu resources and thus get ready navigation.

The necessary recommendations for creating a navigation sliding curtain can be found on the page Navigation drawer - Patterns - Google design guidelines.

Let's move on to the details.

Now the recommendations do not specify the exact size of the curtain, although previously it was necessary to specify the width of the blind itself. Apparently, the new class itself provides the necessary width. The last two attributes are of interest.

The `NavigationView` tag contains a link to its own markup in the `app` attribute: `headerLayout`, which points to the `nav_header_main.xml` file (top of the curtain), and to the menu in the `app: menu` attribute that refers to the `menu / activity_main_drawer.xml` menu resource.

The curtain itself consists of two main parts - at the top there is a picture and text, and in the bottom there is a menu with icons. The menu is in turn divided into two groups. At the top of the icons you can select and the selected item will remain highlighted. The items at the bottom of the menu are not highlighted. If you open the `activity_main.xml` file in Design mode, you can see how the application with an open curtain will look.

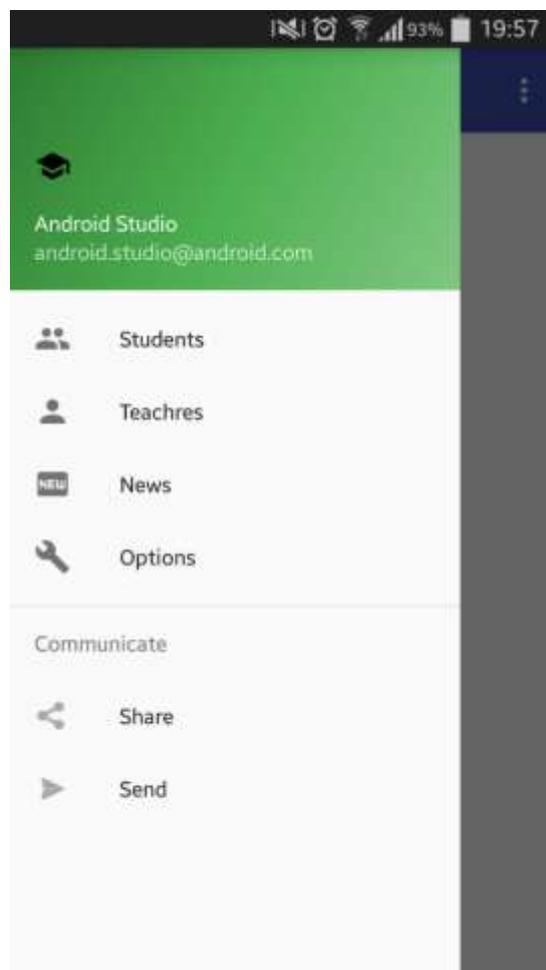


Fig 2.10 The NavigationView Activity

As you can see, there are small discrepancies, but on the whole it is the same. Let's look at its contents.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/drawer_layout"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:fitsSystemWindows="true"
  tools:openDrawer="start">

<include
  layout="@layout/app_bar_main"
  android:layout_width="match_parent"
  android:layout_height="match_parent" />

<android.support.design.widget.NavigationView
  android:id="@+id/nav_view"
  android:layout_width="wrap_content"
  android:layout_height="match_parent"
  android:layout_gravity="start"
  android:fitsSystemWindows="true"
  app:headerLayout="@layout/nav_header_main"
  app:menu="@menu/activity_main_drawer" />

</android.support.v4.widget.DrawerLayout>
```

Now let's look at the resource of the navigation menu

res/menu/activity_main_drawer.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<group android:checkableBehavior="single">
<item
    android:id="@+id/nav_students"
    android:icon="@drawable/ic_people_black_24dp"
    android:title="Students" />
<item
    android:id="@+id/nav_teachers"
    android:icon="@drawable/ic_person_black_24dp"
    android:title="Teachres" />
<item
    android:id="@+id/nav_news"
    android:icon="@drawable/ic_fiber_new_black_24dp"
    android:title="News" />
<item
    android:id="@+id/nav_options"
    android:icon="@drawable/ic_build_black_24dp"
    android:title="Options" />
</group>
<item android:title="Communicate">
<menu>
<item
    android:id="@+id/nav_share"
    android:icon="@drawable/ic_menu_share"
    android:title="Share" />
<item
    android:id="@+id/nav_send"
```

```
        android:icon="@drawable/ic_menu_send"
        android:title="Send" />
</menu>
</item>
</menu>
```

The principle of creating menu items remained standard. Each menu item is an item tag with an icon and text. Grouping uses the group element. The behavior of the menu items in the group is controlled by the android attribute: `checkableBehavior`. The example uses the value `single` - when you click on the menu item, it remains selected (radioButton switch principle). There are three options available.

- `single` - you can select one group item (switch)
- `all` - you can select all the elements of the group (checkbox)
- `none` - items are not selected

There are two main approaches in the use of fragments.

The first method is based on replacing the parent container. Standard layout is created and in the place where the fragments will be used, a container is placed, for example, `FrameLayout`. In the code, the container is replaced with a fragment. If you use a similar script, the fragment tag is not used in the markup, since it can not be changed dynamically. Also, you have to update the `ActionBar`, if it depends on the fragment.

The second option - used separate markup for phones and tablets, which can be placed in different resource folders. For example, if a two-panel layout with two fragments on one activity is used in the tablet, we use the same activity for the phone, but we connect another markup that contains one fragment. When we need to switch to the second fragment, we run the second activity.

The second approach is the most flexible and generally preferred way of using fragments. Activity checks in what mode (its sizes) it is launched and uses different markup from the resources. Graphically it looks like this:

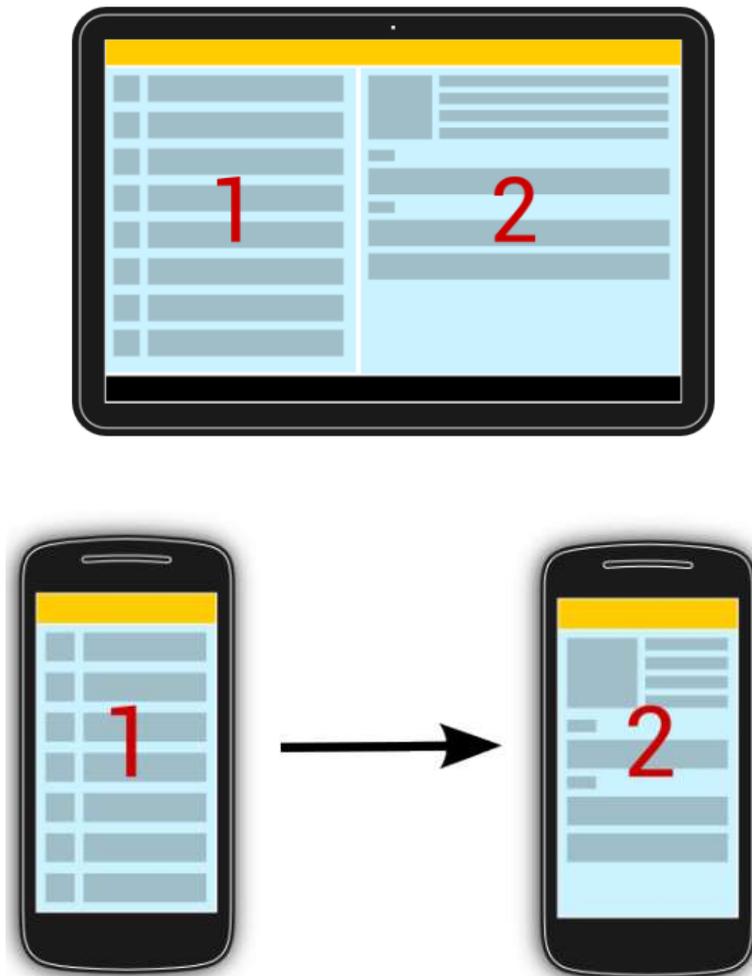


Fig. 2.11 Characteristic Fragment

Main classes

The fragments themselves are inherited from `android.app.Fragment`. There are subclasses of fragments: `ListFragment`, `DialogFragment`, `PreferenceFragment`, `WebViewFragment`, etc. It is not excluded that the number of classes will increase, for example, another class of `MapFragment` appeared.

To interact between fragments, the `android.app.FragmentManager` class is used - a special fragment manager.

As in any office, a special agent does not do the work with his own hands, but uses assistants. For example, for transactions (add, delete, replace), the helper class `android.app.FragmentTransaction` is used.

For comparison, I will list the class names from the compatibility library:

- `android.support.v4.app.FragmentActivity`
- `android.support.v4.app.Fragment`

- android.support.v4.app.FragmentManager
- android.support.v4.app.FragmentTransaction

As you can see, the difference in the one class that I brought first. It is used instead of the standard Activity, so that the system understands that it is necessary to work with fragments. At the moment, the studio creates a project based on ActionBarActivity, which is a subclass of FragmentActivity.

In one application, new fragments and fragments from the compatibility library can not be used.

The general algorithm for working with fragments is as follows:

Each fragment must have its own class. The class is inherited from the class Fragment or similar classes, as mentioned above. It's like creating a new activity or a new component.

Also, as in activity, you create various methods such as onCreate (), etc. If the fragment has markup, then the onCreateView () method is used-consider it an analog of the setContentView () method, in which you connected the activity markup. The onCreateView () method returns a View object, which is the root element of the fragment markup.

The markup for a fragment can be created programmatically or declaratively via XML.

Creating markup for a fragment is no different from creating a markup for activity. Here is a code snippet from the onCreateView () method:

```
public class FirstFragment extends Fragment implements OnClickListener {  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        View view = inflater.inflate(R.layout.first_fragment,  
            container, false);  
        Button nextButton = (Button) view.findViewById(R.id.button_first);  
        nextButton.setOnClickListener(this);  
    }  
}
```

```
    return view;
}
// ...
}
```

Looking at this code, you must understand that the fragment uses the markup from the file `res / layout / first_fragment.xml`, which contains the button with the id `android: id = "@ + id / button_first"`. There is also a similarity with the connection of components in the activity.

Note that before the `findViewById ()` method, `view` is used, since this method refers to the component, not to the activity, as we usually did in programs, when we simply omitted the name of the activity. Those. In our case, we are looking for a link to a button not among activity markers, but inside the markup of the fragment itself.

It should be remembered that in the `inflate ()` method the last parameter must be `false` in most cases.

2.2. Working with http and AsyncTask

Using AsyncTask

Asynchronous tasks are threads running in parallel with the fundamental thread in which the UI of the application is running, and not synchronized with it. This implies the thread of the UI, running such an offbeat undertaking, does not know when the consequence of its operation will come back to it. The benefit of this engineering is the consistent accessibility of the UI for communication. It doesn't solidify, it doesn't quit, sitting tight for the finish of the work of any long haul assignment, yet keeps on working.

The drawback of this approach is the trouble in restoring the consequence of the work of the nonconcurrent errand to the UI. Since the interface does not know when the nonconcurrent thread is done, the thread must itself yield the consequence of its work to the interface, when this outcome is prepared for it.

Moreover, he ought to have the capacity to occasionally answer to the interface on the advance of his work (for instance, to move the advance bar).

All this is finished by the suitable classes (Thread, ThreadPoolExecutor, Runnable) and programming methods, yet it looks moderately troublesome, on the grounds that to stay away from thread clashes it's quite recently unrealistic to control the interface from them. We need to depend on extra structures. Here for this situation additionally helps AsyncTask.

The AsyncTask class is a wrapper over thread classes and makes it less demanding to work with them. The class contains a few abrogated techniques, of which the supersede strategy is `doInBackground ()`, the substance of which, truth be told, will be executed in a different thread.

Also, the `onPreExecute ()`, `onProgressUpdate ()` and `onPostExecute ()` strategies, which are called before the thread is begun, on an order from the thread, and after the thread is ended, are frequently enough utilized, of which, indeed, the interface is controlled (we recall that Directly from the thread control the interface can not).

The AsyncTask strategy enables you to perform offbeat operation in the UI. It performs blocking operations on the specialist thread and afterward distributes the outcomes in the UI thread without having to freely process the threads and/or handlers.

To utilize this technique, you have to make a subclass of AsyncTask and execute the `doInBackground ()` callback strategy that works out of sight thread pool. To refresh the UI, you should execute the `onPostExecute ()` technique, which conveys the outcome from `doInBackground ()` and works in the UI thread, so you can securely refresh the UI. The undertaking is performed by calling the `execute ()` technique from the UI thread.

For instance, you can execute the past illustration utilizing the AsyncTask technique as takes after:

```
public void onClick(View v){
```

```

newDownloadImageTask().execute("http://example.com/image.png");
}

privateclassDownloadImageTaskextendsAsyncTask<String,Void,Bitmap>{
/** The system calls this to perform work in a worker thread and
 * delivers it the parameters given to AsyncTask.execute() */
protectedBitmap doInBackground(String... urls){
return loadImageFromNetwork(urls[0]);
}

/** The system calls this to perform work in the UI thread and delivers
 * the result from doInBackground() */
protected void onPostExecute(Bitmap result){
mImageView.setImageBitmap(result);
}
}
}

```

Now the user interface is protected and the code is simpler, since the work is divided into the part that must be executed in the workflow and the part that must be executed in the UI thread.

Overview of the AsyncTask class:

- You can specify the type of parameters, the progress values and the final value of the job using universal components
- The `doInBackground ()` method is performed automatically in the workflow
- The methods `onPreExecute ()`, `onPostExecute ()` and `onProgressUpdate ()` are started in the user interface thread
- The value returned by the `doInBackground ()` method is sent to the `onPostExecute ()` method

- You can call `publishProgress ()` at any time in `doInBackground ()` to run `onProgressUpdate ()` in the user interface thread
- The job can be canceled at any time from any thread

Another problem that you may encounter when using a workflow is an unpredictable restart of an action due to a configuration change in runtime (for example, when the user changes the screen orientation), which can destroy the worker thread. To see how you can save a job during one of these restarts and how to properly cancel a job when the action is destroyed, examine the source code for the sample Shelves application.

Working with HTTP requests

Today, almost all applications use HTTP / HTTPS requests as a kind of transport for their data. Even if you do not directly use these protocols, many of the SDKs that you most likely have already included in your applications (for example, metrics, fall statistics, advertising) use HTTP / HTTPS to work with the network. To date, there are very few libraries that the developer could freely use in his project. And I'll try to tell you about the main ones in this post.

Android developers have many reasons to opt for third-party libraries, instead of already built-in APIs, such as `HttpURLConnection` or `Apache Client`. For example:

1. Ability to cancel a network call
2. Concurrent query execution
3. Connection pool and reuse of existing socket connections
4. Local server response caching
5. A simple asynchronous interface to avoid locking the main or UI threads
6. Convenient wrapper over REST API
7. Policy of repetition and delays
8. Effective loading and transformation of images
9. Serialization in the form of JSON
10. Support for SPDY, `http / 2`

At the absolute starting point, there were just 2 HTTP customers in Android: `URLConnection` and Apache HTTP Client. According to the official post on the Google blog, `URLConnection` had a few bugs in prior variants of Android:

Before Froyo, `URLConnection` had some not extremely decent bugs. Specifically, calling `close()` in the `Readable InputStream` could degenerate the association pool.

At a similar time, Google did not have any desire to create and relocate to Apache HTTP Client, in light of the fact that the huge size of their API averts us from enhancing this library without losing in reverse similarity. The Android summon does not work effectively on the Apache HTTP Client.

Before Froyo, most designers liked to utilize distinctive customers in view of the OS form:

Apache HTTP customer has few bugs on Eclair and Froyo, so it is the best decision for these adaptations. And for Gingerbread and more youthful is more qualified `URLConnection`. The effortlessness of the API and light weight are useful for Android. Straightforward pressure and reserving of reactions help to build the speed and keep the battery. New applications ought to utilize `URLConnection`.

The `OkHttp` library was made to work specifically with the top level of Java attachments, without utilizing any extra conditions. It comes as a JAR record, so designers can utilize it on any gadget with the JVM (where we incorporate, obviously, and Android). To streamline the move to their library, Square executed `OkHttp` utilizing the `URLConnection` and Apache customer interfaces.

`OkHttp` got a great deal of circulation and group bolster, and at last, Google chosen to utilize rendition 1.5 in Android 4.4 (KitKat). In July 2015, Google formally perceived the `AndroidHttpClient`, in view of Apache, out of date, alongside the arrival of Android 5.1 (Lollipop).

Later on, we will utilize the `HttpURLConnection` library to perform nonconcurrent demands with `AsyncTask`, as this will give us more noteworthy

flexibility of decision in making questions, and will save us the need to introduce non-standard libraries in the Android SDK.

2.3. Server-side data processing with PHP

Inside PHP script has several ways to access data sent to Client by protocol HTTP. Up to version PHP 4.1.0 access to such data was carried out by the names of the transferred variables (recall that the data is transmitted in the form of pairs "The variable name, the symbol "=", Value of the variable. ") Thus, if, for example, first_name = Ninawas given, then the script appeared inside the script Variable \$first_name with the value Nina. If it was required to distinguish between the methods used to transfer the data, then associative arrays \$HTTP_POST_VARS and \$HTTP_GET_VARS, the keys of which were the names of the transferred variables, and the values are the values of these variables, respectively. Thus, if the first_name=Nina pair is passed by the method GET, then \$HTTP_GET_VARS["first_name"] = "Nina".

Use in the program the names of the transferred variables is directly unsafe. Therefore, it was decided PHP 4.1.0 enable to access variables sent using HTTP requests, special array - \$_REQUEST. This array contains data passed by methods POST and GET, and also with the help of HTTP cookies. This is a superglobal associative array, i.e. Its values can be obtained anywhere in the program, using as the key the name of the corresponding variable (form element).

Example 4.2. Suppose we created a registration form for participants in the correspondence school of programming, as in the example above. Then in the file1.php, which processes this form, you can write the following:

```
<?php
$Str = "Hello,
". $_REQUEST["first_name"]."
". $_REQUEST["last_name"]."!<br> ";
$Str. = "You have chosen to study the course on
". $_REQUEST["kurs"];
```

```
echo $Str;
```

```
?>
```

Example 4.2. File 1.php processing form.html

Then, if in the form we entered the name "Vasya", the surname "Petrov" and chose among all courses the course onPHP, on the browser screen we get the following message:

```
Hello, Vasya Petrov!
```

```
You have chosen to study the course on PHP
```

After the introduction of the array\$_REQUEST arrays \$HTTP_POST_VARS and \$HTTP_GET_VARS for homogeneity were renamed to \$_POST and \$_GET, respectively, but they themselves did not disappear from everyday use for reasons of compatibility with previous versions PHP. Unlike their predecessors, arrays \$_POST and \$_GET became super global, i.e. Accessible directly and within functions and methods.

Here is an example of using these arrays. Suppose we need to process a form that contains input elements with the names first_name, last_name, kurs (for example, the form form.html above). The data were transmitted by POST, and the data transmitted by other methods, we do not want to process. You can do this as follows:

```
<?Php
```

```
$Str = "Hello,
```

```
". $_POST["first_name"]."
```

```
". $_POST["last_name"]."!<br> ";
```

```
$Str. = "You have chosen to study the course on".
```

```
$_POST["kurs"];
```

```
echo $str;
```

```
?>
```

Then on the screen of the browser, if we entered the name "Vasya", the name "Petrov" and chose among all courses a course on PHP, we will see the message, as in the previous example:

Hello, Vasya Petrov!

You have chosen to study the course on PHP

In order to preserve the possibility of processing scripts of earlier versions than PHP 4.1.0, was introduced directive `Register_globals`, allowing or prohibiting Access to variables directly by their names. If in Configuration file PHP parameter `Register_globals= On`, then to the variables passed Server methods GET and POST, it is possible to address simply by their names (ie it is possible to write `$first_name`). If `Register_globals=Off`, then you need to write `$_REQUEST["first_name"]` or `$_POST["first_name"]`, `$_GET["first_name"]`, `$HTTP_POST_VARS["first_name"]`, `$HTTP_GET_VARS["first_name"]`. From the security point of view, it is better to disable this directive (i.e.`Register_globals=Off`). When the directive is activated `Register_globals` the above arrays will also contain the data transferred to client.

Sometimes there is a need to learn meaning of any environment variable, such as the method used to transmit the request or IP-Address of the computer that sent Request. You can get this information using the function `Getenv ()`. It returns value The environment variable whose name is passed to it as a parameter.

```
<?
Echo getenv ('REQUEST_METHOD');
// return the used method
Echo getenv ('REMOTE_ADDR');
// print the user's IP address,
// Send the request
?>
```

Example 4.3. Using the `getenv ()` function

As we said, if the method is used GET, then the data is transferred by adding Query strings in the form of pairs "variable_name =Value" to URL of the resource. "Everything that is written in URL after the question mark, can be obtained by using the command

```
Getenv ('QUERY_STRING');
```

Thanks to this, you can use the method GET to transfer data in some other form. For example, specify only the values of several parameters via the plus sign, and in the script, parse the query string into parts or you can transfer the value of just one parameter. In this case, in the array \$_GET will appear an empty element with a key equal to this value (all Query string), and the "+" symbol, found in the query string will be replaced by an underscore "_".

By the method of POST data is transmitted only using forms, and User(Client) does not see which data is sent to server. To see them, the hacker should change our form. Then the server will send the results of processing the wrong form to the wrong place. To avoid this, you can check Address of the page from which the data was sent. This can be done again using the function Getenv ():

```
Getenv ('HTTP_REFERER');
```

Now is the time to solve the problem formulated at the beginning of the lecture.

Example of processing a request using PHP

Recall what the task was, and clarify its wording. It is necessary to write a registration form for participants in the correspondence school of programming and, after registration, send a message to the participant. We called this message a universal letter, but it will differ slightly from the letter that we compiled in the previous lecture. Here we also will not send anything by e-mail, so as not to be like spammers, but simply generate this message and display it on the browser screen. The initial version of the registration form we already mentioned above. We will change it in such a way that each registrant can choose as many courses to visit, and we will not confirm receipt of the registration form.

```
<H2> Student registration form </ h2>  
<Form action = "1.php" method = POST>  
Name <br><input type = text name = "first_name"  
Value = "Enter your name"><br>  
Surname <br><input type = text name = "last_name"><br>  
E-mail <br><input type = text name = "email"><br>
```

```

<P> Choose the course you would like to attend: <br>
<Input type = checkbox name = 'kurs []' value = 'PHP'> PHP <br>
<Input type = checkbox name = 'kurs []' value = 'Lisp'> Lisp <br>
<Input type = checkbox name = 'kurs []' value = 'Perl'> Perl <br>
<Input type = checkbox name = 'kurs []' value = 'Unix'> Unix <br>
<P> What do you want us to know about you?<BR>
<Textarea name = "comment" cols = 32 rows = 5></ textarea>
<Input type = submit value = "Send">
<Input type = reset value = "Cancel">
</ Form>

```

Listing 4.4. Form_final.html

Here everything is quite simple and understandable. The only thing that can be noted is the way the values of the checkbox are passed. When we write in the name of the kurs element, this means that the first checked check box element will be written to the first element of the kurs array, the second checked box to the second element of the array, and so on. You can, of course, just give different names to check box elements, but this will complicate the processing of data if there are many courses.

The script, which all this will parse and process, is called l.php (the form refers to this file, which is written in its attribute *Action*). The default is to transfer the method *GET*, but we indicated *POST*. According to the information received from the registered person, the script generates a corresponding message. If a person has chosen some courses, then he is given a message about the time of their holding and about the lecturers who read them. If the person did not choose anything, then a message is displayed about the next meeting of the correspondence school of programmers (ZSPP).

2.4. The problem of automated timetable generation

Scheduling is one of the most common tasks in planning and optimizing the educational process in educational institutions. On how well the schedule is

composed, the effectiveness of the work of teachers, the assimilation of educational material by students, the rational use of material resources depends.

Automation of scheduling is a classic task in the management systems of an educational institution, but at the moment there is no single, generally accepted method for solving it.

All approaches to the question of scheduling are based on heuristic methods that come to the person with work experience. It is problematic to formalize these methods, since they are connected with the decision making by the operator making up the schedule, which is guided by experience and intuition. Often, the employee who makes up the schedule can not answer the question why he chose a certain variant of the placement of classes, and not some other among the admissible ones.

But, despite the complexity of the formalization of algorithms, it is possible to distinguish the features of such heuristic approaches, based on the requirements for scheduling. Undoubtedly, for each educational institution these requirements are different, because they are historically due to the peculiarity of the organization of the educational process itself. However, even with all the details, we can single out the general requirements for the schedule:

1. The minimum number of lessons for student a day;
2. The maximum number of hours of training load per week for each student;
3. The maximum number of sessions a student a day;
4. Minimizing the windows of students;
5. The account of time distances between cases at change of the case to the trained;
6. Accounting of the wishes of teachers;
7. The cycle of lessons in discipline should not end with a lecture if there are seminars (practical) classes;
8. The cycle of lessons in discipline should not begin with a seminar (practical) lesson, if there are lecture classes;

9. For each lecture session, all stream groups should be approached, having received the same number of hours of seminar (practical) lessons;

10. Do not conduct more than two lectures on the same discipline per day and no more than one / two seminars on the same discipline per day;

11. The minimum number of lessons a teacher a day;

12. The maximum number of sessions a teacher a day;

13. Minimization of windows at the faculty (hereinafter PPP);

14. Minimization of the processing of teachers according to the staff schedule;

15. Minimize the number of simultaneously held the same disciplines in an educational institution at the same time. This directly affects the amount of allocated PPP to ensure the training load;

16. Maximum use of the auditor fund. This includes requirements for the most dense placement of students according to the capacity of the latter, as well as minimizing idle time;

17. The account of time distances between cases at change of the case by the teacher.

The task of scheduling directly depends on the initial conditions. You can group scheduling tasks for these conditions into specific groups:

1. Scheduling with a priori known information about the distribution of groups between the PPP;

2. Scheduling without taking into account the PPP, using only the load of departments;

3. Scheduling without taking into account the workload of the departments.

Let us consider in more detail the features of each of the above-described task groups.

In problems with known information about the distribution of groups between PPPs, the problem arises of taking into account the wishes of the teacher, monitoring trips with changing the case, overlapping in the teacher's schedule (the

presence of several classes at the same time). The employee who composes the timetable is required to compile two reference schedules at once: group schedule and PPP schedule. Particularly difficult task becomes, if the teachers distributed the load among themselves to the accuracy of the group, and the schedule operator can not change this distribution.

Therefore, one of the approaches that allow to reduce the rigid scope of the restrictions in which the operator turns out is the use of the distribution of teachers without groups: the teachers indicate only the faculty, the course and the number of groups that they will lead during the semester. A specific group is assigned by the operator that composes the schedule. This allows you to get one additional degree of freedom, which will reduce the number of deadlock situations.

In tasks in which only the workload of the departments is used, the operator no longer has to take into account the wishes of the PPP or the intersection of their occupations, and the operator is not required to conduct at once two schedules: groups and teachers. However, indirectly, however, the operator has to take into account that the department should minimize the required PPP to ensure the training load, therefore it becomes very important to fulfill the requirement No. 15.

Accounting for this requirement leads to the appearance of such a value as the capacity of the department, which shows how many classes the department can conduct The same discipline at the same time. After scheduling the departments have to arrange the teachers themselves.

In tasks, scheduling without taking into account departments of degrees of freedom is much more than in others, so a very logical approach is the movement from the group or the flow. However, in this case, when composing the timetable, it is not possible to take into account the requirements related to teachers (No. 6, 11-17).

Each requirement for the scheduled schedule, from the above listed (1-17), imposes its own peculiarities on scheduling. For example, if the requirements 3, 4, 10, 11, 12 and 15 are simultaneously present in the task, for optimal placement of

the tasks for group 1 tasks within the teacher and the stream, one must try to place the load that the instructor reads for all stream groups.

For the optimal placement of classes for group 2 tasks, due to the lack of information about the instructor within the stream, we must try to place lectures first, and then, in the following pairs, seminar classes on the subjects whose lectures were in the previous pairs, which allows us to compose the workload for The same discipline. This allows us to indirectly take into account the requirements 11, 12, 15, associated with the teacher.

CHAPTER 3. CREATING MOBILE APPLICATION OF UNIVERSITY TIMETABLE

3.1. Installing and configuring Android SDK and Android Studio

First of all, you need to make sure that you have a JDK (Java Development Kit) installed. It's a must-have component for Java development, and since Android development is done on Java, it's also for Android development too.

You can download JDK at www.oracle.com. You need to select the newest JDK, not JRE

Now let's proceed with the installation of Android Studio.

First, you need to download Android Studio at <https://developer.android.com/>.

One installer will have everything you need - the IDE itself, Android Emulator, Android SDK. What is not included in the package, the installer proves independently.

Let's move directly to the installation. There's nothing unusual in it - the usual installer dialogue. In the process, you will only need to answer one important question - the location of the installation kit and the SDK, and then this is optional. If everything is clear with the studio itself, then the SDK should be attentive. To install the SDK, you need at least 3.2 GB of disk space. This is the minimum, in fact, you need more space, because after some time you will need to download the updated SDK. So if you are not sure if there will be enough room in the future - it's better to change the location to a more spacious disk.

After that, it will be necessary to press the "next" button several times, and this is the end of the installation of Android Studio.

Set up Android Studio

When you first start Android Studio will ask you the standard question about importing the configuration:

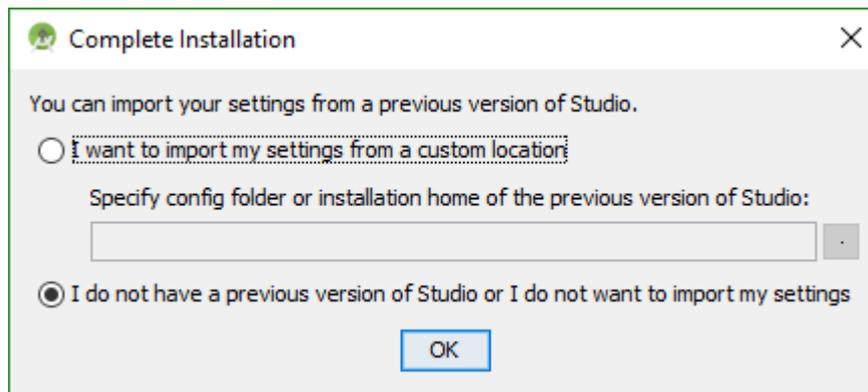


Fig. 3.1 The Standard question about importing the configuration

By default, the same checkbox will be selected, just click on the "OK" button.

After that, Android Studio will start download Android SDK. This may take some time. If the download fails, the IDE will prompt you to try again - be sure to click Retry.

When the download is complete, click "Finish".

In principle, the installation is finished, but we recommend making some more settings for more efficient operation.

First, we will change the topic to dark ("Darcula"). To do this, in the settings (File-> Settings), in the section Appearance & Behavior, select the topic Dracula. Secondly, tick "show line numbers" and "show method separators". The first will display the line numbers to the left of the text, the second will draw the delimiters between the methods in the code.

These two options are incredibly important, and especially for beginners, they make it easier to navigate in large amounts of code.

Third, we will configure the autocompletion. To do this, select "None" in "Case sensitive completion". The default option implies the activation of autocompletion only if the first letter is typed in the correct register. The None option will call for autocompletion, regardless of which register you started typing.

Creation of the first project

After setting up and installing Android Studio, it's time to create our first project.

In the main window of Android Studio, click on "Start a new Android Studio project":

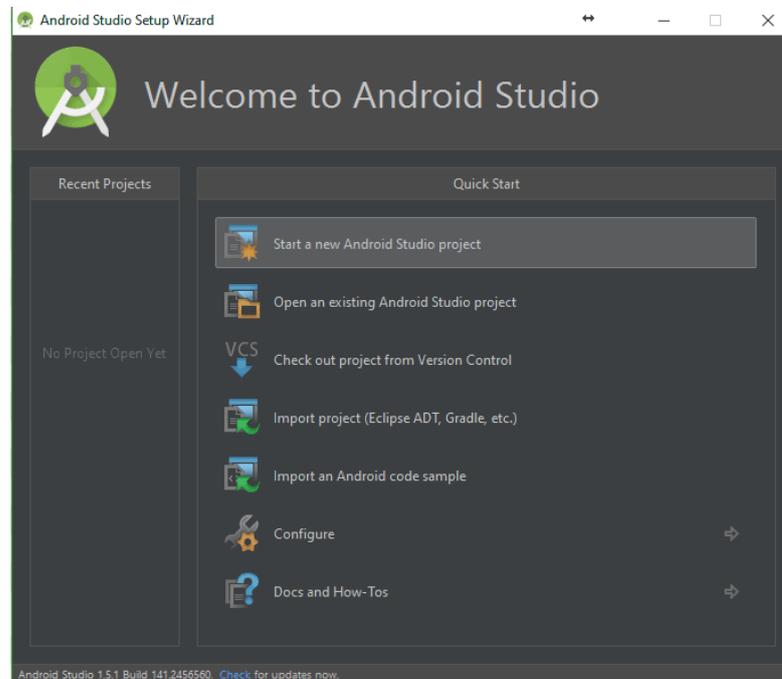


Fig 3.2 The Main window of Android Studio

A new window will appear in which we need to perform several actions.

In the first, you must specify the name of the application, the company domain (the name of the package will be created from these two parameters), and the project location on the disk:

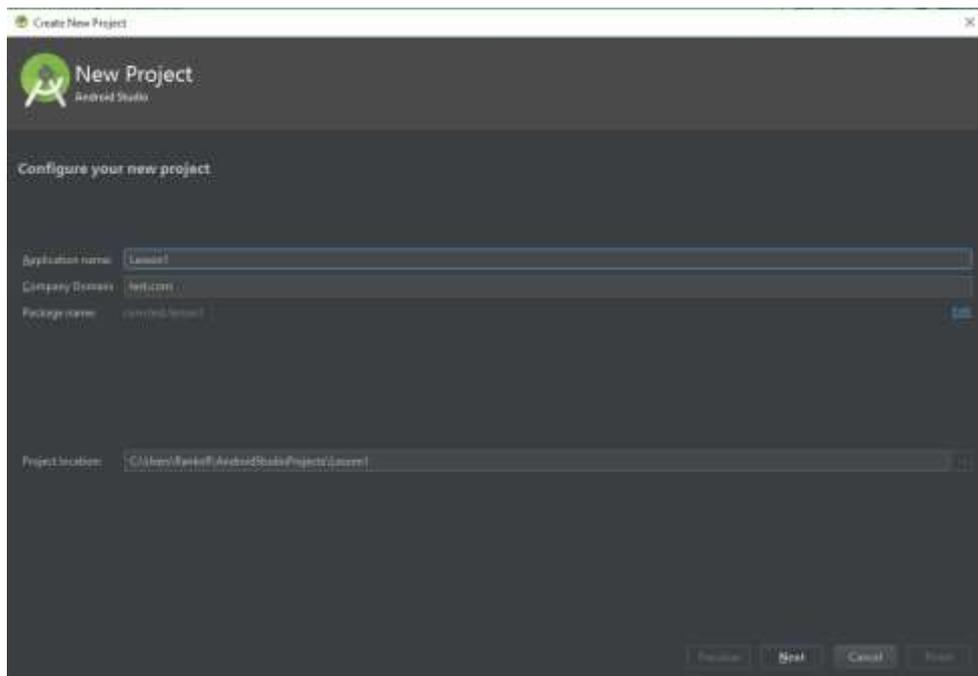


Fig 3.3 The Name of the package will be created from these two parameters

In Android, as in Java, the main application identifier is the package name. If you've worked with Java before, you know what it is. For those who do not know, I recommend Google, or, for example, here is this article.

Next, Android Studio will ask us which and what versions of the SDK we want to use. So far, we have enough "Phone and Tablet" SDK, the API version is 16 instead of the recommended 15th, since the API 15 is already irrelevant and not at all common:

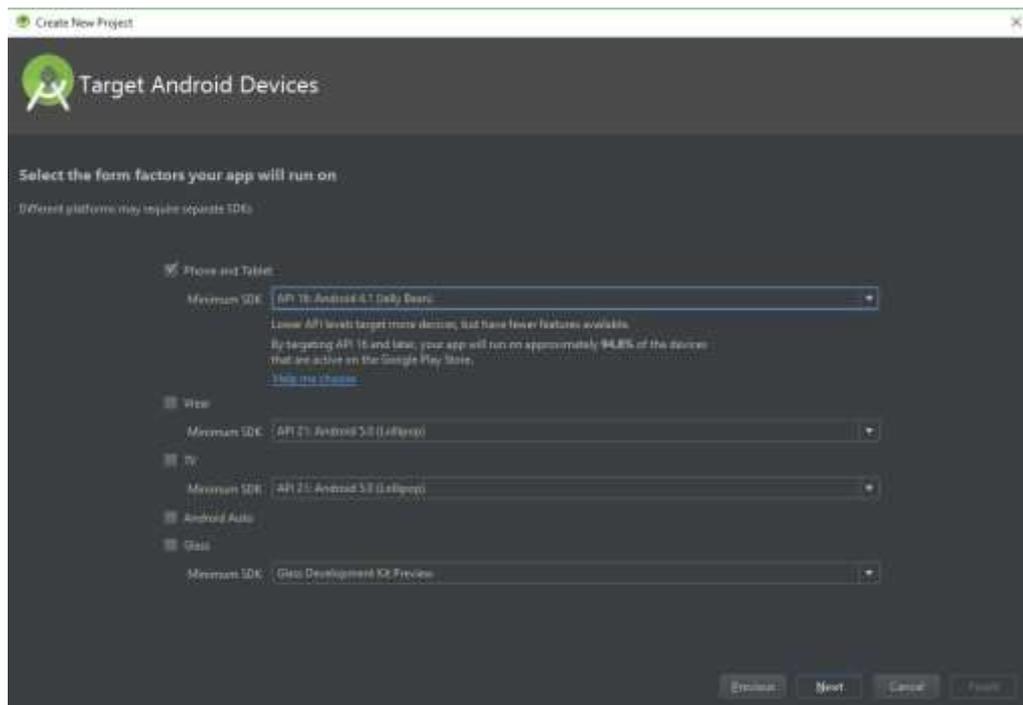


Fig. 3.4 Selected versions of the SDK we want to use.

On the next screen, we will be asked which components of the application we want to create. Select Empty Activity:

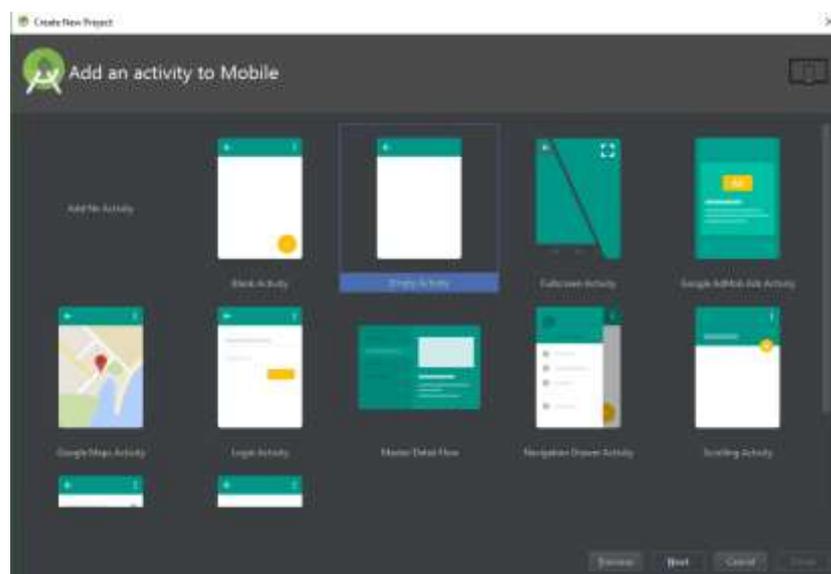


Fig. 3.5 Select Activity

In the next step, simply click "Finish", without changing anything.

Next, we will have to wait a while (from a minute to five minutes, depending on the power of your computer), while Android Studio creates a project.

At the end of this process, you will finally see your first project:

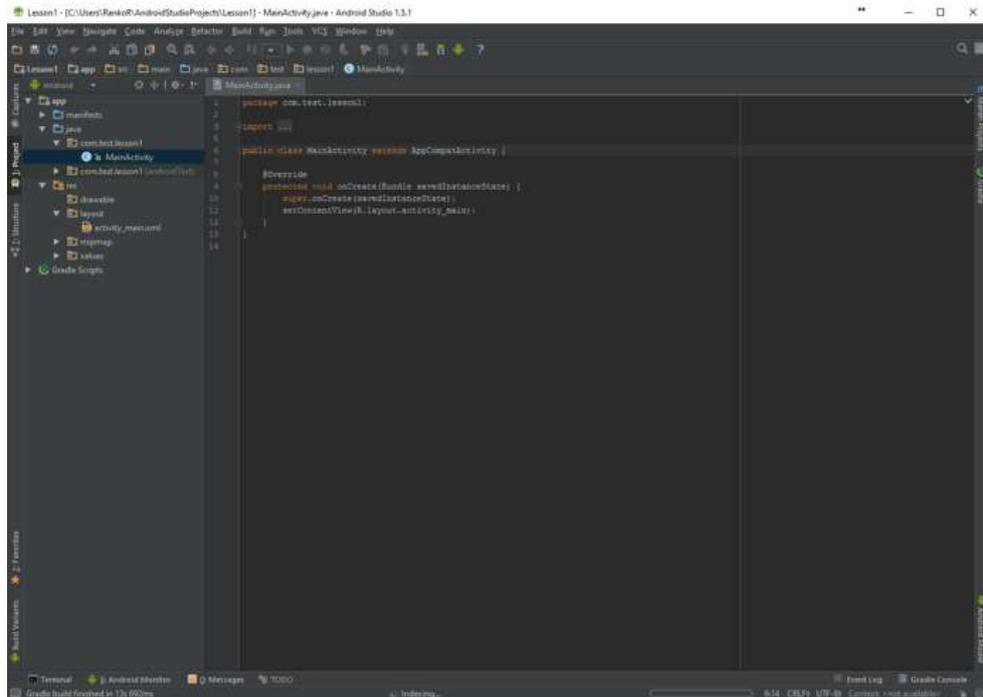


Fig. 3.6 Code editor window

In order to run the created project, we need an Android emulator.

Creating an Android emulator

To create the Android emulator, we need Android AVD Manager (AVD = Android Virtual Device). To start AVD, you must click the appropriate button in the toolbar.

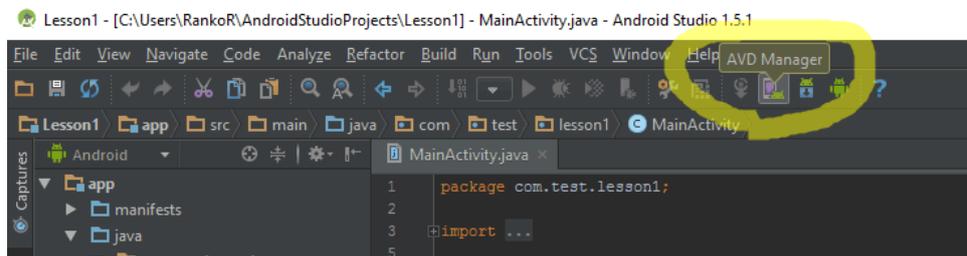


Fig. 3.7 Creating an Android emulator

In the appeared window we press the Create Virtual Device button, where we set the desired virtual device.

And then just click "Next" several times and finally "Finish"

Running Hello World

It's time to launch our first project, created in Android Studio! Click on this button (or Shift-F10):

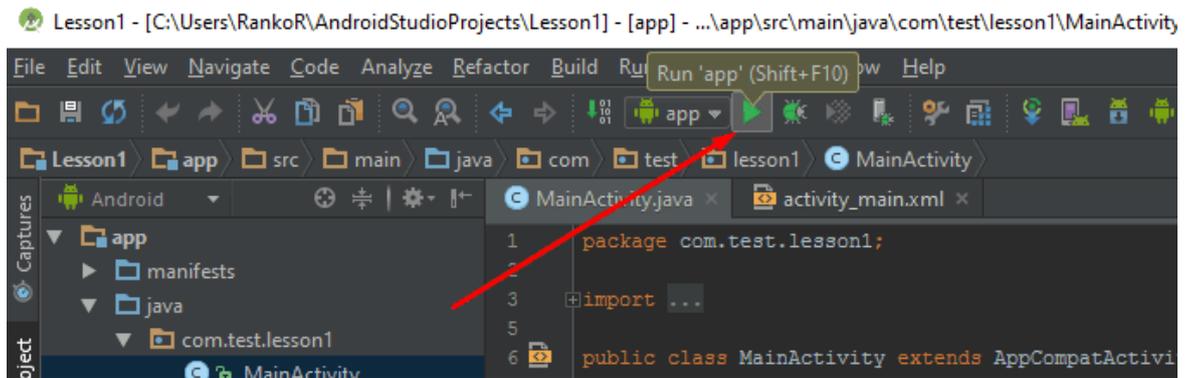


Fig. 3.8 Running

After that you will see a dialog in which you will be asked to select the device on which the IDE should launch the collected application:

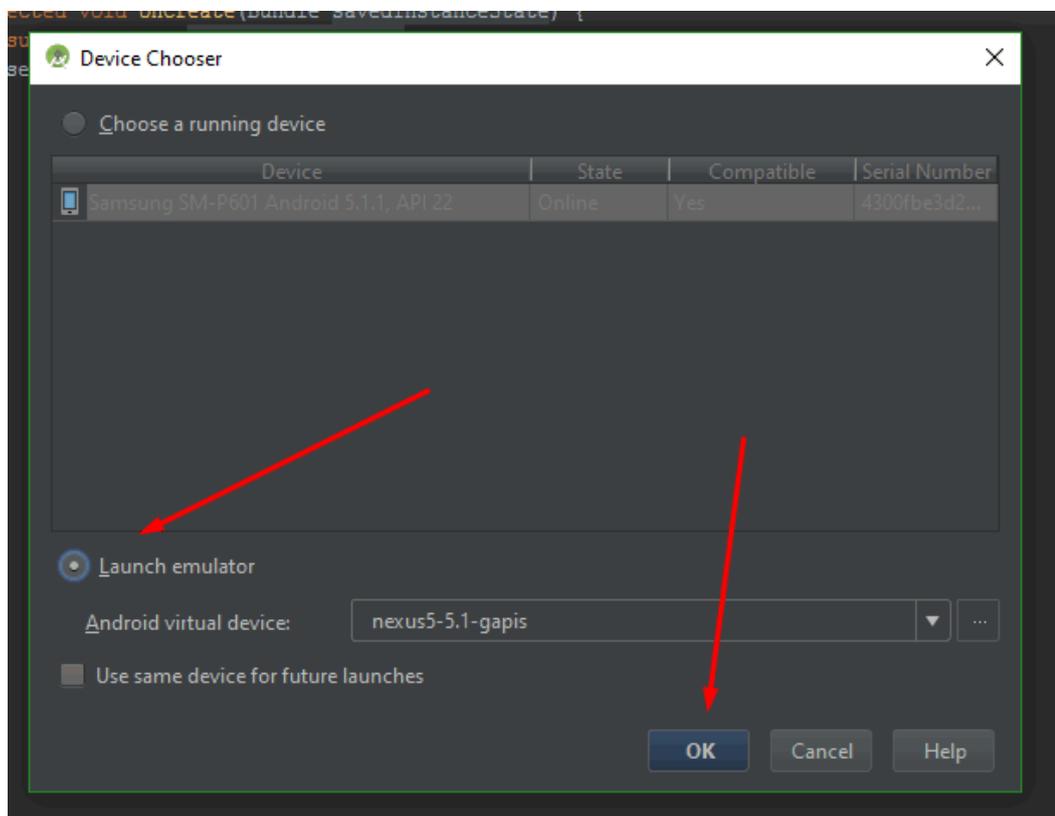


Fig. 3.9 Select the device in the IDE

Put a check mark selected by the arrow and click "OK". The project will start assembling, starting the emulator, installing the application on the emulator, and running the application.

This will take some time (the more powerful your computer is, the less time it will take), since the emulator is a slow enough thing, despite the tremendous improvements in the last couple of years.

And now, after 1-10 minutes (after the emulator starts, of course, the projects will be assembled and launched faster), you will finally see your Hello World on the emulator screen!

3.2. Processing requests to server

The AsyncTask class offers a simple and convenient mechanism for moving laborious operations in the background thread. With it, you can handle events with a graphical stream, which allows you to update the user interface elements to display the results of the task execution or to display the results when the task is completed.

It should be remembered that AsyncTask is not a universal solution for all cases of life. It should be used for not too long operations - loading small images, file operations, operations with basic data, etc.

Directly with the class AsyncTask can not work, you need to inherit from it (extends). Your implementation should provide classes for objects that will be passed as parameters to the execute () method, for variables, as well as for variables where the result will be stored. The format of such a record is as follows:

```
AsyncTask <[Type of input_parameter], [Transition_progress_type], [Result type]>
```

AsyncTask has several basic methods that you need to master first. The `doInBackground ()` method is mandatory, the rest are often used based on the logic of your application:

- `DoInBackground ()` is the main method that is executed in the new thread. Can not access the UI. It is in this method that there must be code for heavy tasks. Accepts a set of parameters of those types that are defined in the implementation of your class.

This method runs in the background thread, so there should be no interaction with the user interface elements in it. Place time-consuming code here using the `publishProgress ()` method, which will allow the `onProgressUpdate ()` handler to pass the changes to the user interface. When the background task is completed, this method returns the end result for the `onPostExecute ()` handler, which will report it to the user interface thread.

- `OnPreExecute ()` - executed before `doInBackground ()`. Has access to UI
- `OnPostExecute ()` - executed after `doInBackground ()` (may not be called if AsyncTask was canceled). Has access to the UI. Use it to update the user interface once your background task is complete. This handler is synchronized with the GUI thread on the call, so inside it you can safely change the elements of the user interface.

- `OnProgressUpdate ()`. Has access to the UI. Override this handler to publish intermediate updates to the user interface. When called, it synchronizes with the GUI thread, so you can safely change the user interface elements in it.

- `PublishProgress ()` - can be called in `doInBackground ()` to display intermediate results in `onProgressUpdate ()`

- `Cancel ()` - cancel task

- `OnCancelled ()` - Has access to the UI. The task was canceled. There are two overloaded versions.

In our project, we will call `URLConnection` to send request to our server to get actual timetable:

```
class bgTask extends AsyncTask<String,String,String>{
```

```

    @Override
    protected void onPreExecute(){
super.onPreExecute();
    }

    @Override
    protected String doInBackground(String... strings){
try{
        URL myurl =new
URL("http://timetable.tatunf.uz/api/groupname/"+strings[0].toString());
        HttpURLConnection con =(HttpURLConnection)myurl.openConnection();
        InputStream response = con.getInputStream();
        Scanner s =new Scanner(response).useDelimiter("\\A");
        String result = s.hasNext()? s.next(): "";
return result;
}catch(MalformedURLException e){
        e.printStackTrace();
}catch(IOException e){
        e.printStackTrace();
    }
return "";
}

    @Override
    protected void onPostExecute(String s){
super.onPostExecute(s);
        txt1.setText(s);
    }
}

```

The `doInBackground ()` method we configure `URLConnection` and send request. We parse data from spinners and send them as GET headers of our HTTP query.

In the `onPostExecute ()` method, we output a response from server into our `TextView` after the task is completed.

Note that if we click on the button while `AsyncTask` is running, a new task will be created and launched on top of the old one. It turns out that the two tasks will work simultaneously with the activity screen. It is necessary to avoid such situations. Later I will show you how.

The class `java.net.HttpURLConnection` is a subclass of `java.net.URLConnection` and allows you to implement the work of sending and receiving data from the network via the HTTP protocol.

The information can be of any type and length. This class should be used to send and receive streaming data whose dimensions cannot be determined in advance. Using this class, you do not need to think about sockets and implement your own methods of communication between the client and the server.

The algorithm of use is as follows:

- Get the `URLConnection` object by calling `URL.openConnection ()` and bring the result to `URLConnection`
- Prepare the required request. The main thing in the request is the network address itself. Also in the query, you can specify various metadata: credentials, content type, session cookies, and so on.
- Optionally load request body. In this case, the `setDoOutput (true)` method is used. The transfer of data written to the stream is returned via the `getOutputStream()` method
- Read the answer. The response header typically includes metadata such as the type and length of the content, the date of the change, the session cookie. You can read the data from the stream through the `getInputStream ()` method. If the answer has no body, then the method returns an empty stream.

- Break the connection. After reading the response from the HttpURLConnection server, close it by calling the disconnect () method. Thus, you release the resources occupied by the connection.

By default, HttpURLConnection uses the GET method. To use POST, call setDoOutput (true) and send the data via openOutputStream (). Other HTTP methods (OPTIONS, HEAD, PUT, DELETE and TRACE) are set via the setRequestMethod (String) method.

3.3. Server-side processing

For server-side processing, we use PHP script of timetable bot for Telegram (http://t.me/tatunf_dars_bot) of Nukus branch of the TUIT. The script is based on MVC model and uses Codeigniter methods to access database tables with current timetable. The database structure of timetable is given below:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	group_id	int(11)			No	None		
3	teacher	varchar(255)	utf8_general_ci		No	None		
4	coursename	varchar(255)	utf8_general_ci		No	None		
5	rawValue	varchar(255)	utf8_general_ci		No	None		
6	room	varchar(255)	utf8_general_ci		No	None		
7	day	int(11)			No	None		
8	num	int(11)			No	None		

Fig. 3.10 The Database structure of timetable

In this table, we store information about lessons. In the group_id field we store information about group, in the teacher field there is an information about lesson teacher, coursename contains information about lesson. All these data also duplicated in rawValue field for faster access. Room field contains room number, the day field is integer field containing values from 1 to 6 representing the day of

the week. Finally, num field contains information about lesson number (represented by a number from 1 to 4).

In the listing below there's a source code of PHP controller to access the timetable:

```
<?php defined('BASEPATH') OR exit('No direct script access allowed');
```

```
class Api extends CI_Controller{  
    function __construct() {  
        parent::__construct();  
    }  
}
```

```
function groupId($id = false)  
{  
    $this->load->model('timetable_model');  
    if ($id !== false) {  
        $response = $this->timetable_model->get_timetable($id,'week',true);  
        echo $response;  
    }  
}
```

```
function groupName($groupName = "")  
{  
    $groupName = strtolower($groupName);  
    $groupName = str_replace('uzb','o'zb',$groupName);  
    $this->load->model('timetable_model');  
    $q = $this->db->where('group_name_api',$groupName)->get('tbl_groups');  
    if ($q->num_rows()>0) {  
  
        $info = $q->row_array();  
        $response = $this->timetable_model->get_timetable($info['id'],'week',true);  
    }  
}
```

```

echo $response;
}
}

```

```

function teacher($fio=""){
    echo "Result:";
}
}

```

3.4. Building APK

Finally, we should build our APK file

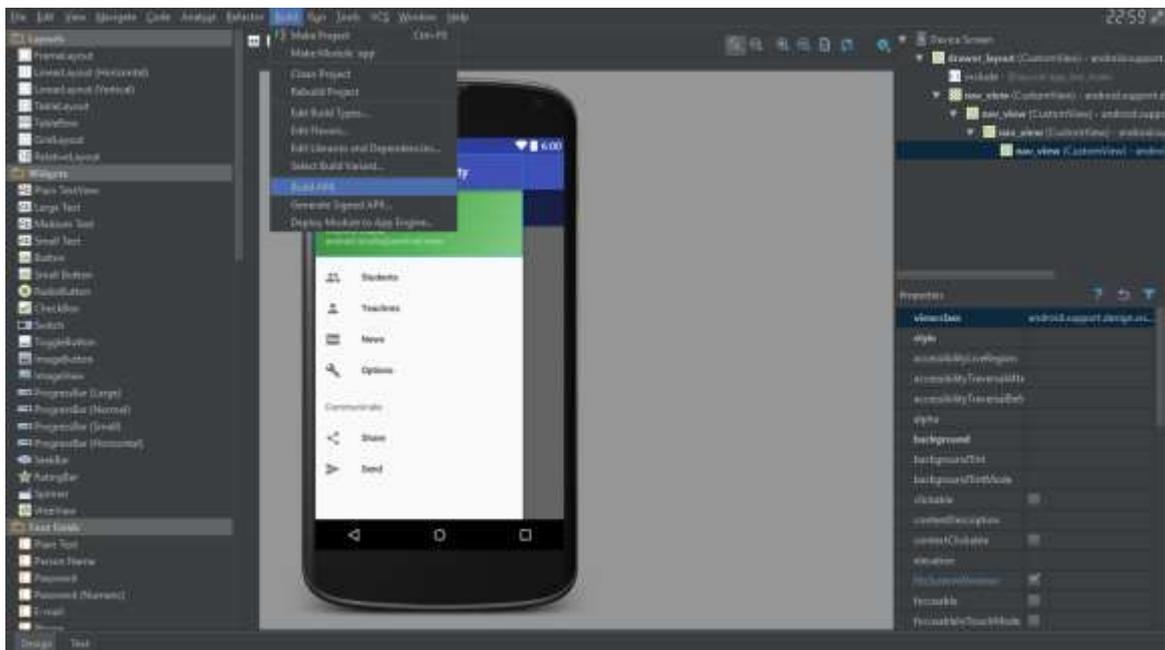


Fig. 3.11. Build apk menu item

To create an APK for your app, follow these steps:

1. Select the build variant you want to build from the **Build Variants** window.
2. Click **Build > Build APK** in the menu bar.

To instead build the APK and immediately run it on a device, click **Run** in the toolbar.

All built APKs are saved in *project-name/module-name/build/outputs/apk/*. You can also locate the generated APKs by clicking the link in the pop-up dialog that appears once the build is complete.

After we run our application on real Android device, we get following window:

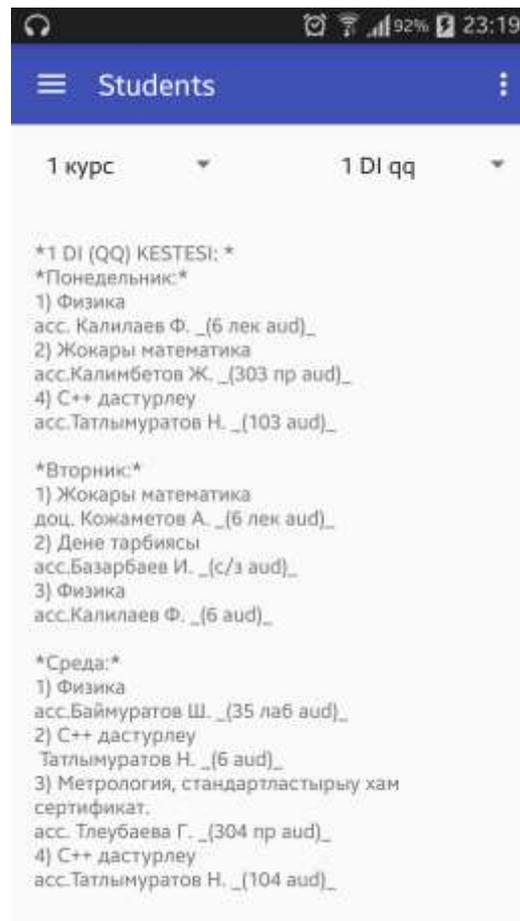


Fig. 3.12 Students Fragment

On this window, we can choose the year and the group and instantly get actual timetable for a chosen group. Surely, application requires internet connection to get updated information about lessons.

You can also observe the timetable of teachers by selecting their last name in appropriate window which appears when you choose Teachers fragment in Navigation drawer. The timetable is represented by days, starting from Monday

until Saturday, and each day contains information about lessons from 1 to 4. For example, here's the time table of Tatlimuratov:

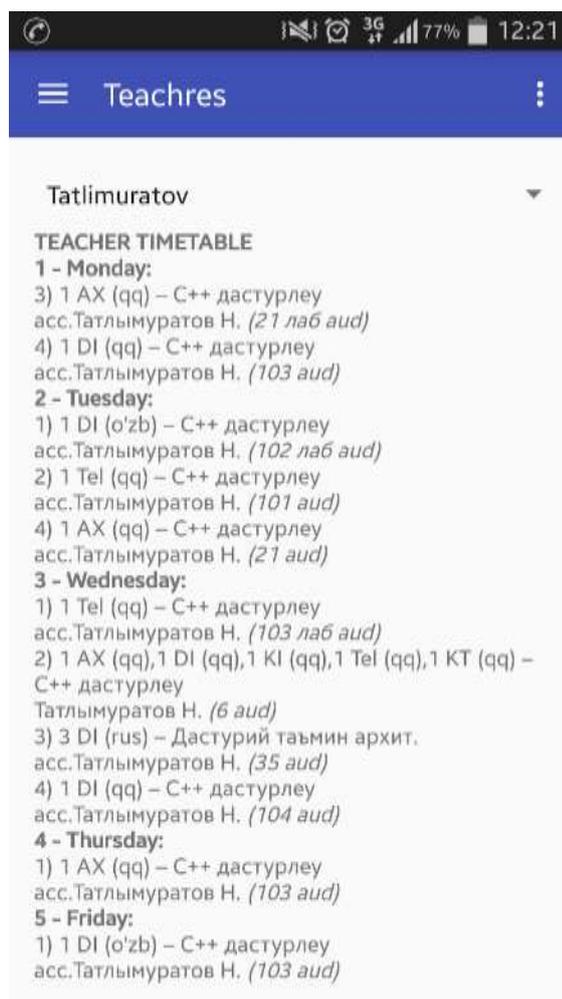


Fig.3.13. Teacher's timetable

The resulting APK file size is about 1.4 megabytes, so users can easily and fastly download it from Play Market or from official website (timetable.tatunf.uz).

Finally we created timetable application and we used the latest technology to create an application that meets the latest design and usability requirements.

If we are going to publish this file in stores like Google Play, we have to generate APK file signed with digital key. It is necessary to make sure that you are the creator of the application. After signing app, it should be aligned with zip-align tool which is located in Java bin folder. It usually takes several hours to check your application and approve it for publishing. If there are no problems during checking, your app will be available for everyone in Google

CONCLUSION

Nowadays, the development of applications for the mobile devices plays a big role for many customers of different services. With its help we can get possibility to use different services, make purchases without much effort and time. No matter how popular and solid the bank servicing payment cards is, it will in future lose significantly to the one that has adopted the mobile application. That is why many companies that understand the value of business applications for smartphones is increasing very fast.

In this qualification work, we created modern mobile application that automates the process of timetable delivery to students and teachers of Nukus branch of the Tashkent University of Information Technologies named after Muxammad al-Xorazmiy. This is one of the first applications of this kind, and we hope that this application may be useful for other universities, after appropriate adaptation of database structures.

Application has intuitive interface, and both teachers and students can easily use it. In this day almost all students of our branch use Telegram and Android versions of timetable application, and we think that this is a big achievement for us. Another advantage of this application that it works in Tas-ix network, so users have no problems with accessing data and there are no delays in delivering it.

We are going to get a certificate of authorship for this application and introduce it to other branches of Tashkent university of Information technologies. While creating this application I learned the principles of creating client-server applications, learned design guidelines of creating Material Design applications and sending http requests using built-in http libraries.

I hope that this application will be useful for students and teachers of our university and will possible facilitate the work of dean's office.

REFERENCES: