

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шебзухова Татьяна Александровна
Должность: Директор Пятигорского института (филиал) Северо-Кавказского
федерального университета
Дата подписания: 23.08.2023 14:46:34
Уникальный программный ключ:
d74ce93cd40e39275c3ba2f58486412a1c8ef96f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

Методические указания

по выполнению лабораторных работ

по дисциплине

«ПРОГРАММИРОВАНИЕ МОБИЛЬНЫХ УСТРОЙСТВ»

для направления подготовки **09.03.02 Информационные системы и технологии**
направленность (профиль) **Информационные системы и технологии обработки
цифрового контента**

Пятигорск
2022

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. Цель и задачи изучения дисциплины.....	3
2. Оборудование и материалы.....	3
3. Наименование лабораторных работ.....	14
4. Содержание лабораторных работ.....	14
Лабораторная работа №1. Установка AndroidStudio.....	14
Лабораторная работа 2. Создание нового проекта.....	23
Лабораторная работа 3. Жизненный цикл Activity.....	46
Лабораторная работа 4. Использование ресурсов приложения.....	50
Лабораторная работа 5. Layout-файл в activity. Смена ориентации экрана.....	53
Лабораторная работа 6. Всплывающие уведомления / toastnotification.....	62
Лабораторная работа 7. Уведомления / pushnotification.....	66
Лабораторная работа 8. Локализация приложения.....	71
Лабораторная работа 9. Переключение между экранами.....	74
Лабораторная работа 10. Организация сервиса в приложении.....	78
Лабораторная работа 11. Сохранение данных в приложении.....	82
Лабораторная работа 12. Знакомство с SQLite. Хранение данных при помощи SQLite.....	86
5. Учебно-методическое и информационное обеспечение дисциплины.....	93
5.1. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины.....	93
5.1.1. Перечень основной литературы:.....	93
5.1.2. Перечень дополнительной литературы:.....	93
5.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине (модулю).....	94
5.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля).....	94

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

ВВЕДЕНИЕ

1. Цель и задачи изучения дисциплины

В лабораторный практикум по дисциплине «Программирование мобильных устройств» включены лабораторные работы по основным разделам этой дисциплины, читаемой на кафедре «Систем управления и информационных технологий». Лабораторные работы ориентированы на приобретение студентами навыков программирования на языке Java для мобильной платформы Android. Работы ориентированы на использование систем автоматизированного проектирования AndroidStudio.

Содержащиеся в практикуме сведения теории, методические указания и рекомендации по выполнению лабораторных работ позволяют использовать его в качестве дополнительного пособия для закрепления курса лекций.

Целью данного лабораторного практикума является поэтапное формирование у студентов знаний, умений и навыков создания приложения для мобильных устройств, изучение языка программирования Java.

Практикум предназначен для студентов Северо-Кавказского федерального университета и может быть полезным для всех желающих ознакомиться с основами программирования мобильных устройств.

Данный вид работы играет важную роль в формировании практических навыков программирования и способствует формированию следующих образовательных компетенций:

Код	Формулировка:
ПК-8	Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО
ПК-9	Способность оценки качества разрабатываемого обеспечения, включая разработку тестов, проведение

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

2. Действителен с 20.08.2021 по 20.08.2022

Для проведения практических занятий необходимо следующее материально-техническое обеспечение: персональный компьютер; проектор; возможность выхода в сеть Интернет для поиска по образовательным сайтам и порталам, интерактивная доска.

3. Наименование лабораторных работ

№ Темы	Наименование работы	Объем часов
	7 семестр	
1	Лабораторная работа 1. Установка AndroidStudio	1,5
2	Лабораторная работа 2. Создание нового проекта	1,5
3	Лабораторная работа 3. Жизненный цикл Activity	1,5
4	Лабораторная работа 4. Использование ресурсов приложения	1,5
	Итого за 7 семестр	6
	Итого	6

4. Содержание лабораторных работ

Лабораторная работа №1. Установка AndroidStudio

До недавнего времени компания Google поддерживала две платформы разработки – AndroidStudio и Eclipse. Поскольку большинство разработчиков мобильных приложений используют среду разработки AndroidStudio и она имеет больше возможностей, от поддержки разработки в Eclipse компания Google отказалась. Таким образом, осталась одна поддерживаемая система – AndroidStudio.

Рассмотрим процесс установки и настройки окружения.

Поскольку разработка приложения для ОС Android осуществляется на языке программирования Java, необходимо скачать и установить пакет JDK (JavaDevelopmentKit) – этот пакет содержит все необходимые библиотеки для нативной разработки на языке программирования Java. Последняя доступная версия располагается на странице компании Oracle, в соответствующем разделе (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>).

Также необходимо скачать и установить пакет разработчика Android – ADT(AndroidDevelopmentTools), который можно найти на официальной странице в сети по адресу: <http://developer.android.com/intl/ru/sdk/index.html>. В последних версиях AndroidStudio уже содержит в себе пакет ADT, по- этому достаточно

только загрузить среду разработки под выбранную платформу. На сегодняшний день поддерживаются следующие операционные системы Linux /

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

Windows / MacOS.

Процесс запуска установки пакета JDK представлен на рис. 1.1.

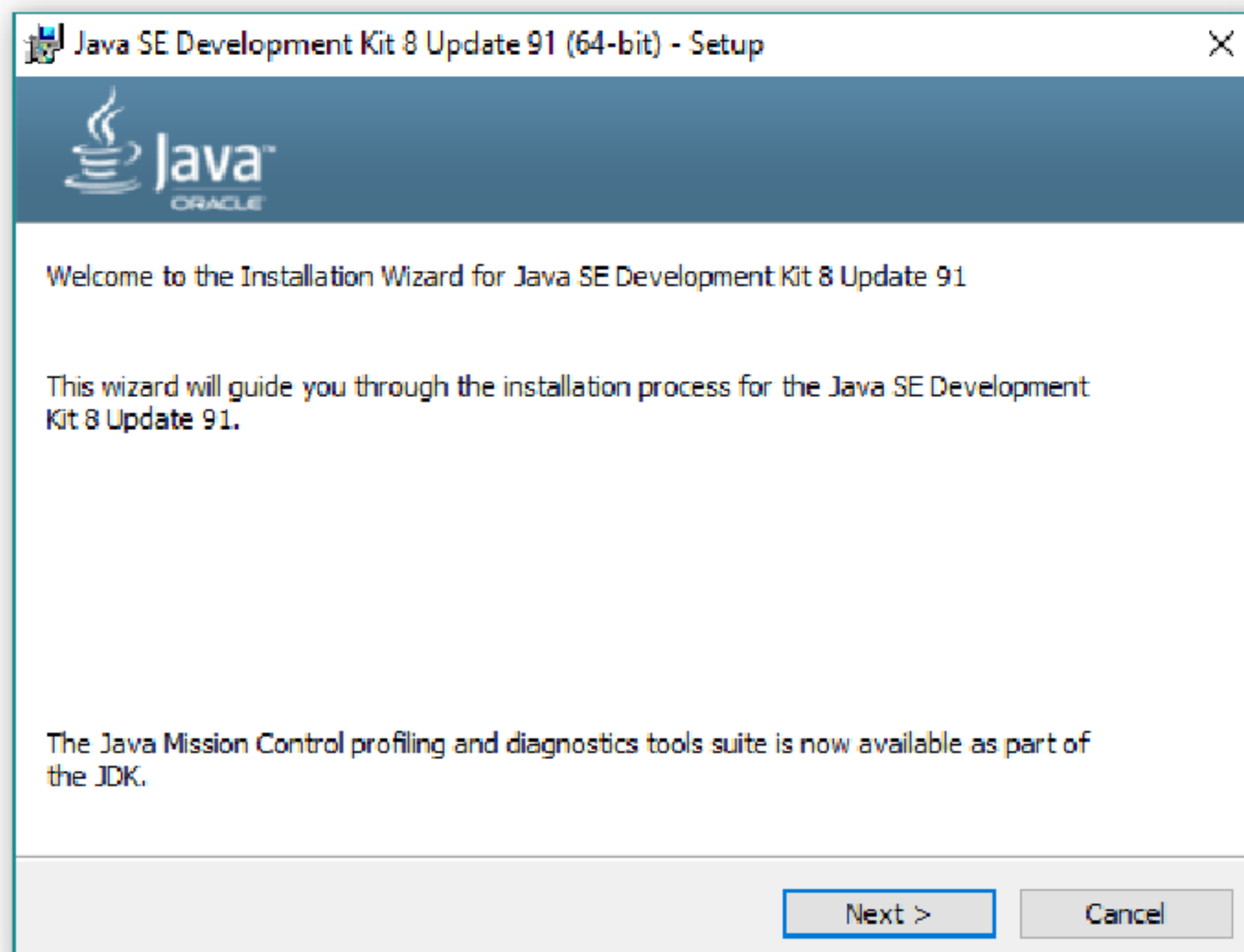
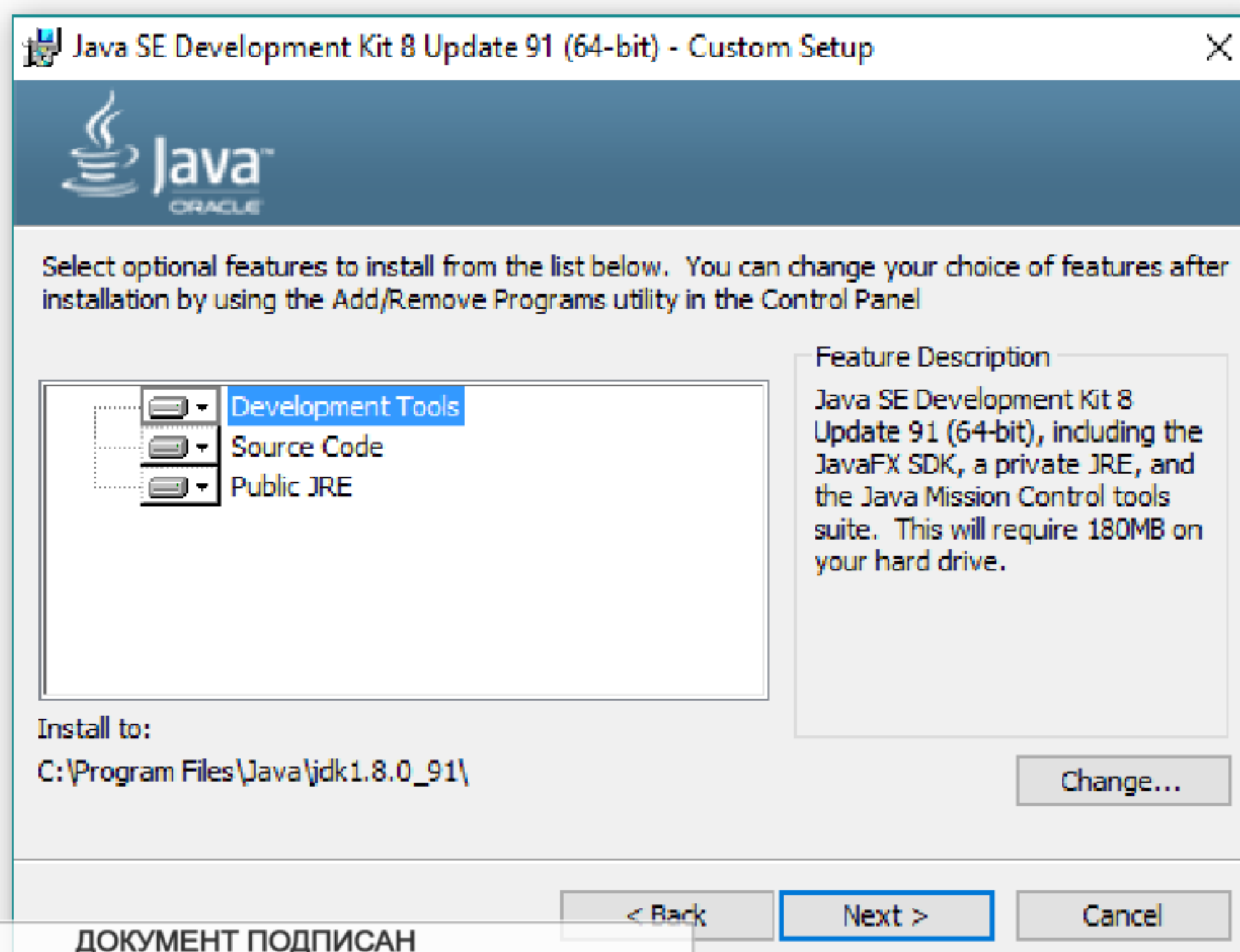


Рис. 1.1. Начало установки JDK

Следующий шаг – установка пакета JDK (рис. 1.2).



ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

Рис. 1.2. Выбор пути для установки JDK

Следом за JDK процесс установки предложит выбрать путь для установки JRE (рис. 1.3).

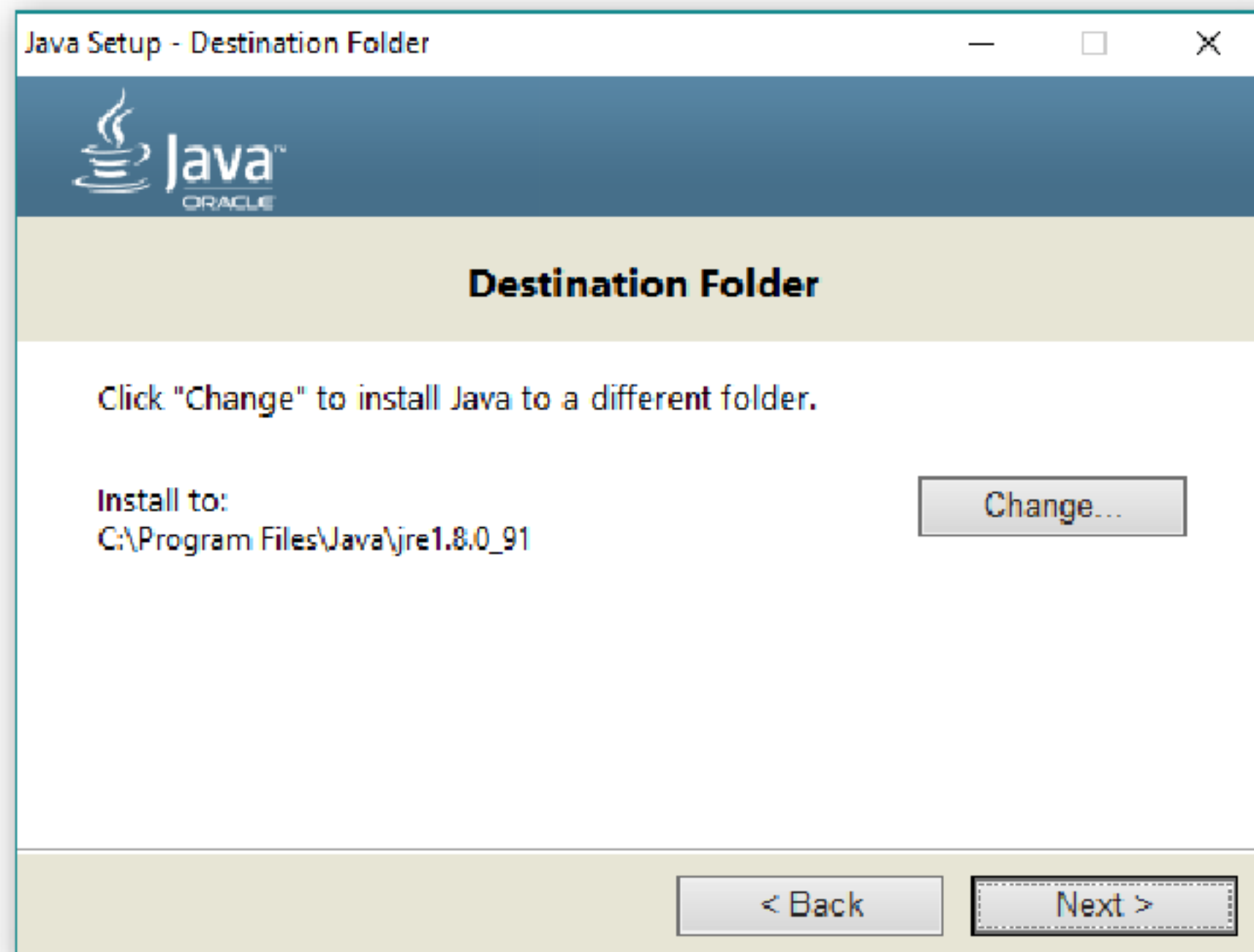


Рис. 1.3. Выбор пути для установки JRE

Следующим шагом будет установка самой AndroidStudio, скачать её можно на сайте www.developer.android.com. Запустив исполняемый файл установки и нажав Next, попадем на экран выбора компонентов для установки (рис. 1.4).

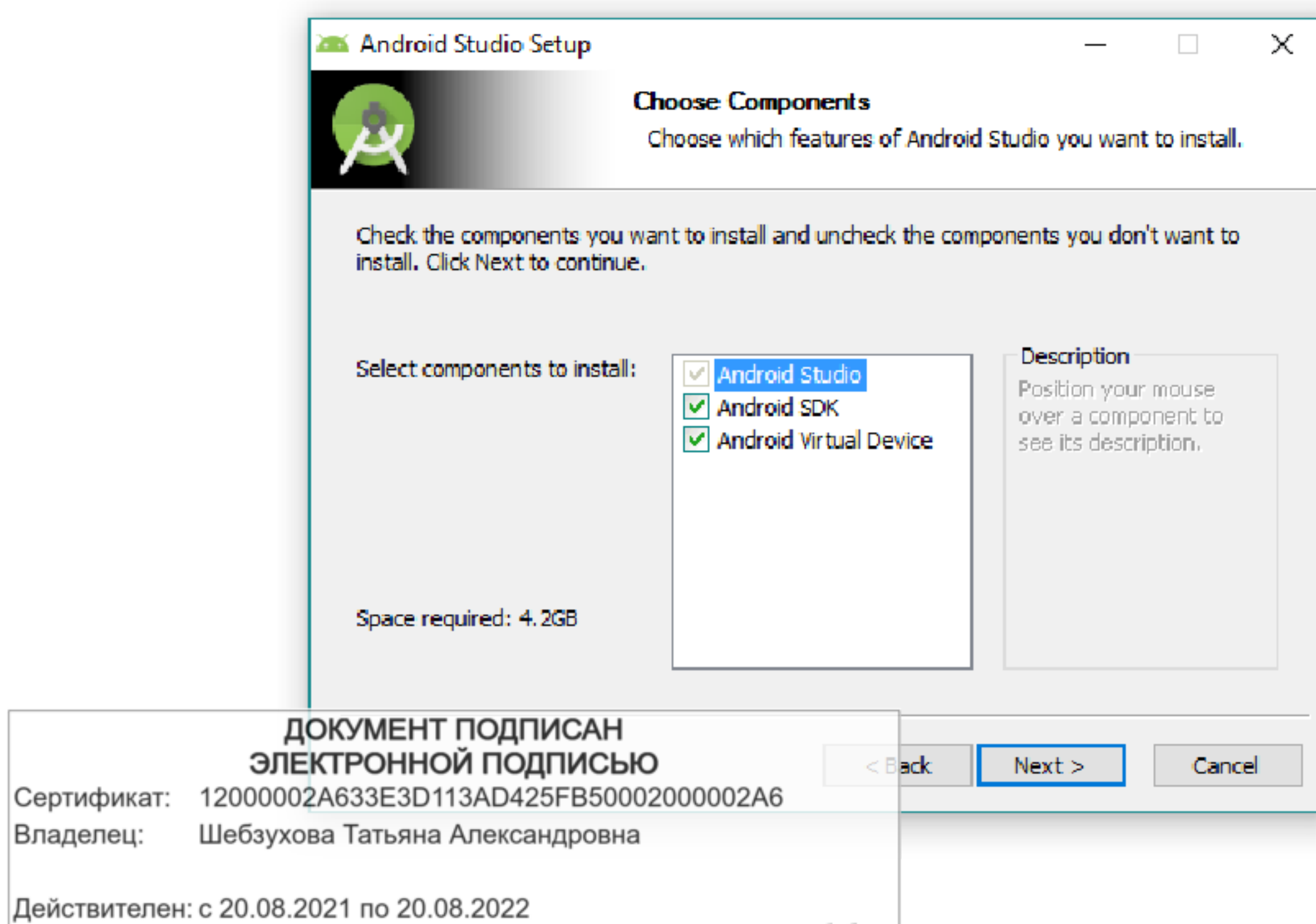


Рис. 1.4. Выбор компонентов для установки

Android Studio – среда разработки (Integrated Development Environment, сокращенное название IDE) на основе популярной IntelliJ IDEA для разработки на языке программирования Java. AndroidSDK – среда разработки приложений для операционной системы Android, включает в себя несколько инструментальных средств, которые позволяют компилировать и отлаживать создаваемые приложения. AndroidVirtualDevice (AVD) – эмулятор мобильного устройства на ОС Android. Используя эмулятор, можно запускать и тестировать приложения без использования реального Android-устройства.

Далее, приняв лицензионное соглашение, требуется выбрать путь, куда необходимо установить AndroidStudio и SDK вместе с AVD.

Важно! В записи пути не должно быть русских символов. Если учетная запись имеет русские буквы в своем имени, следует выбрать другой путь для SDK (рис. 1.5).

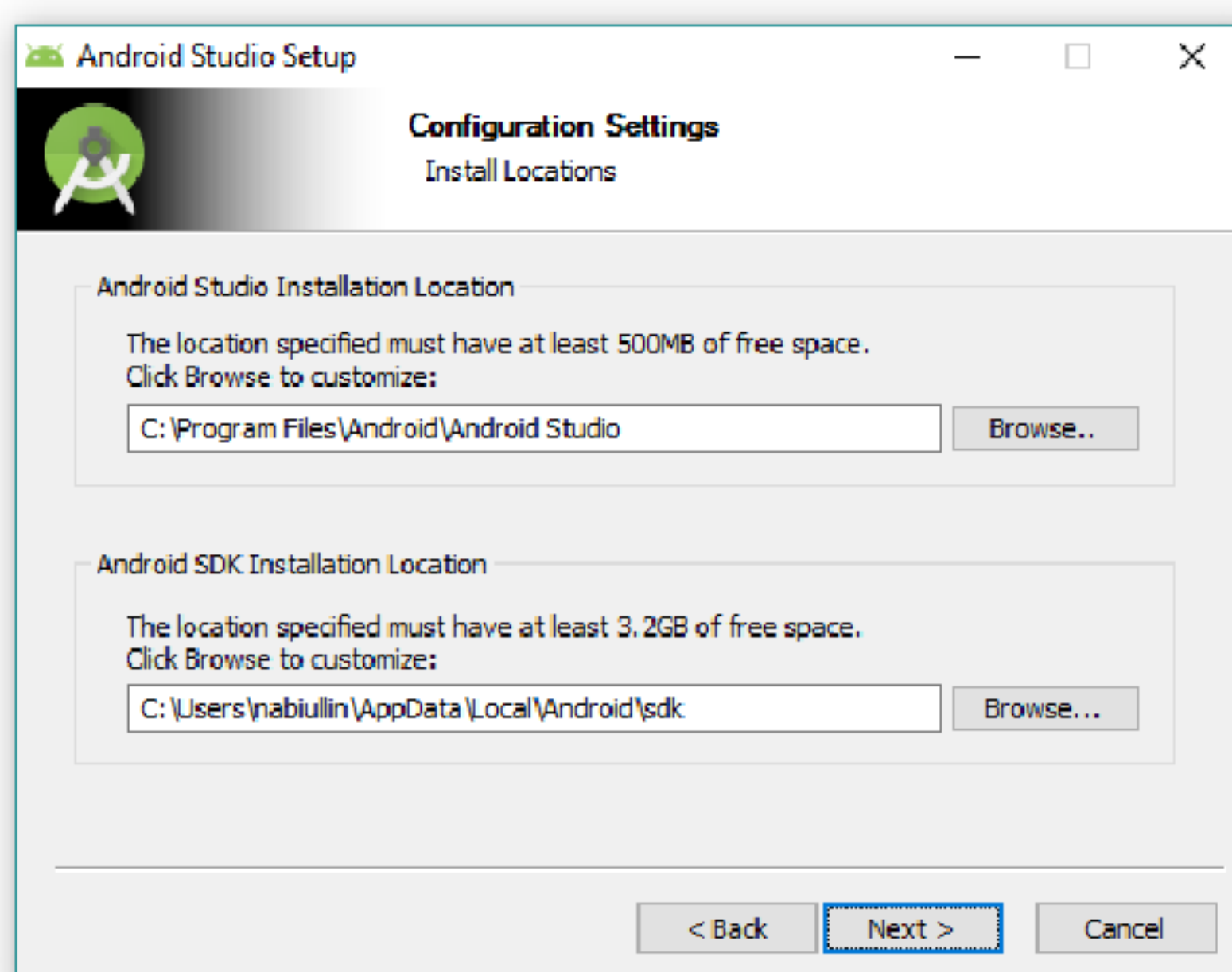
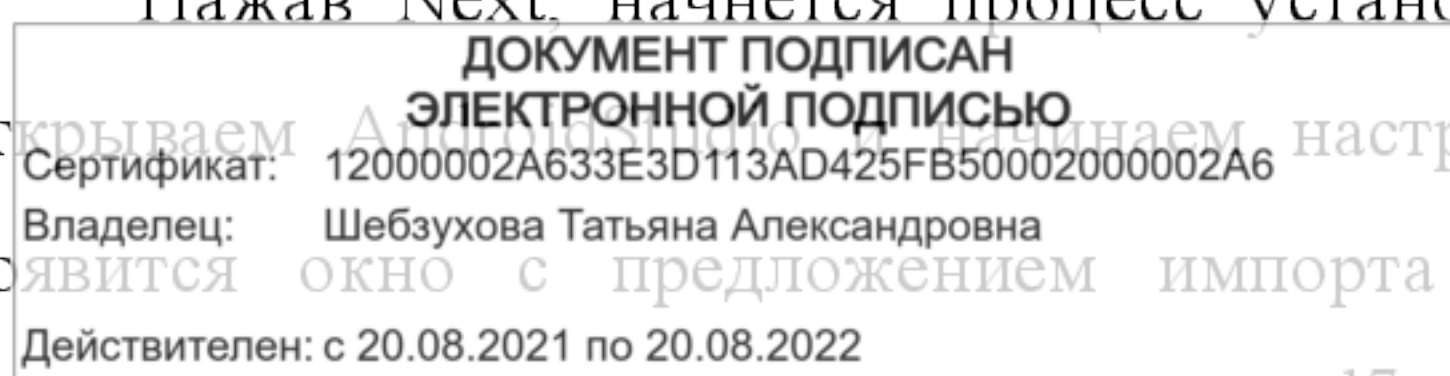


Рис. 1.5. Выбор пути для установки AndroidStudio и SDK вместе с AVD

Нажав Next, начнется процесс установки. После того как он завершится,

открываем Android Studio. Вводим настройку окружения. В первую очередь появится окно с предложением импорта конфигурации предыдущей версии



AndroidStudio (рис. 1.6).

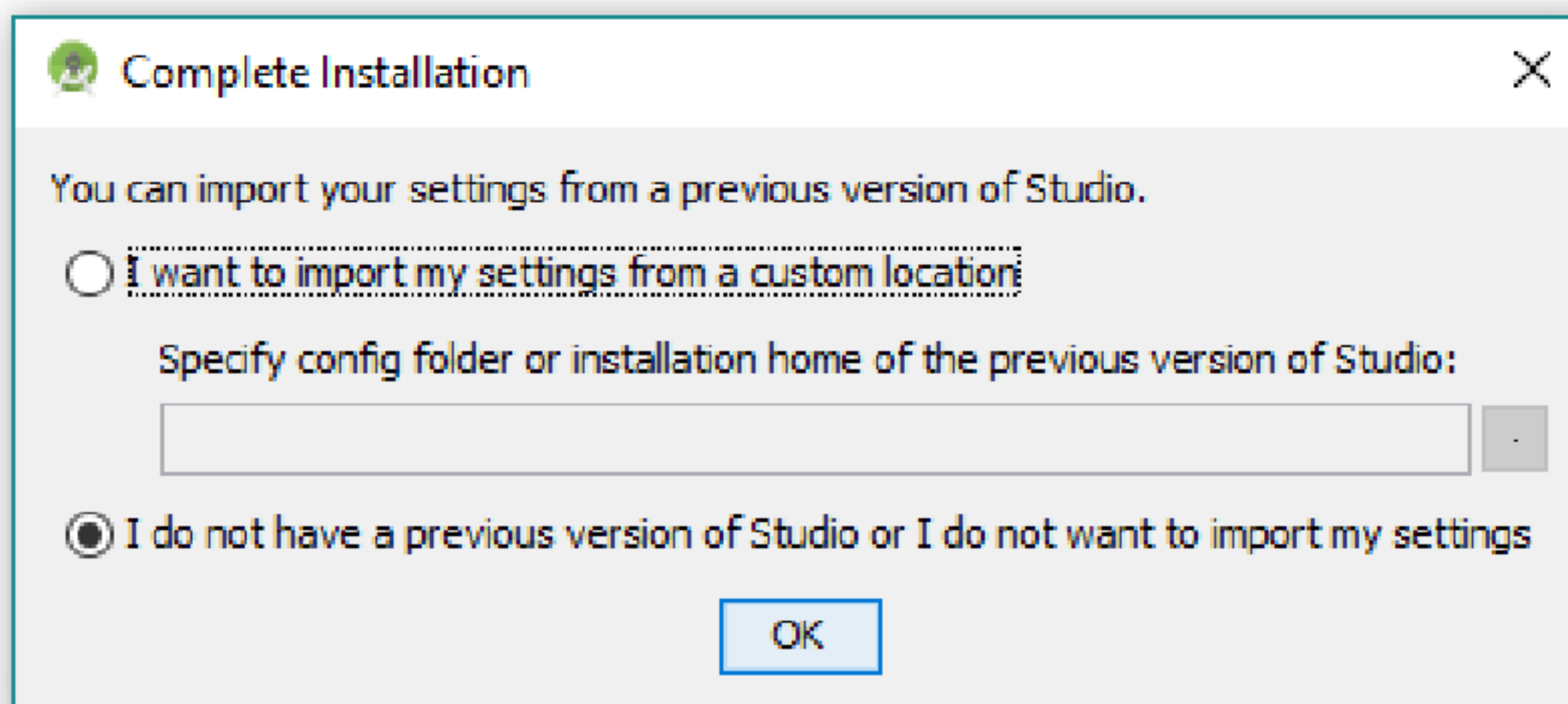
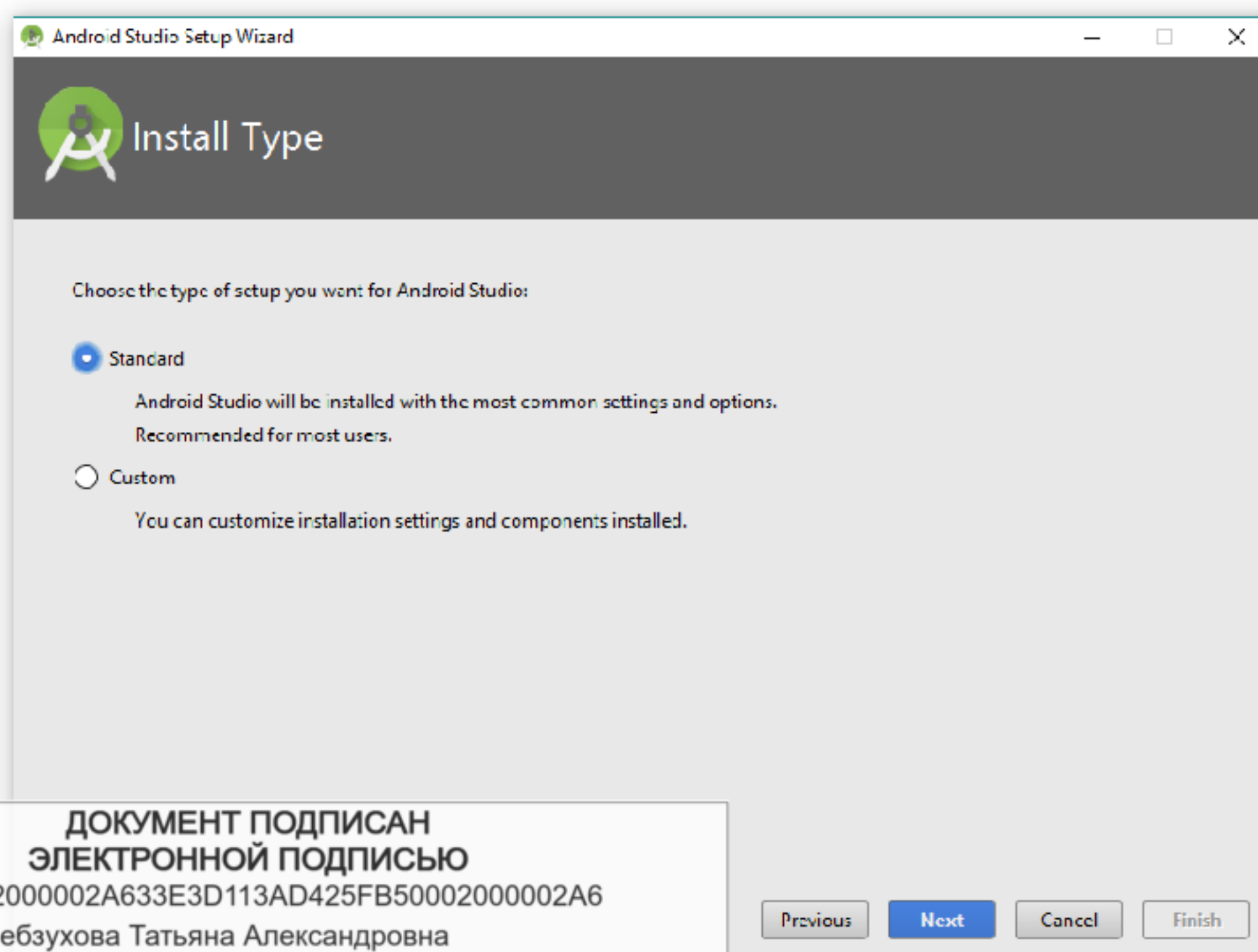


Рис. 1.6. Импорт конфигураций предыдущей версии

Выбираем второй пункт «Я не имею предыдущей версии студии или не хочу импортировать мои конфигурации». Далее откроется окно с предложением настроить AndroidStudio. Нажав Next, попадаем на окно с выбором типа настройки (рис. 1.7). Standard включает в себя стандартные настройки IDE, т. е. будут установлены все минимально необходимые компоненты для программирования. Custom позволит настроить окружение в индивидуальном порядке.



ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

Рис. 1.7. Выбор типа настройки AndroidStudio

По завершении быстрой настройки будут загружены обновления AndroidStudio или SDK, если таковые имеются. Далее откроется начальное окно (рис. 1.8).

В данный момент AndroidSDK обладает необходимым минимумом инструментов и актуальной версией платформы для начала разработки, но если необходимо расширить возможности, можно перейти в менеджер SDK, выбрав опцию – ConfigureAndroidSDK.

Во вкладке SDKPlatforms (рис.1.9) можно выбрать доступные версии AndroidSDK.

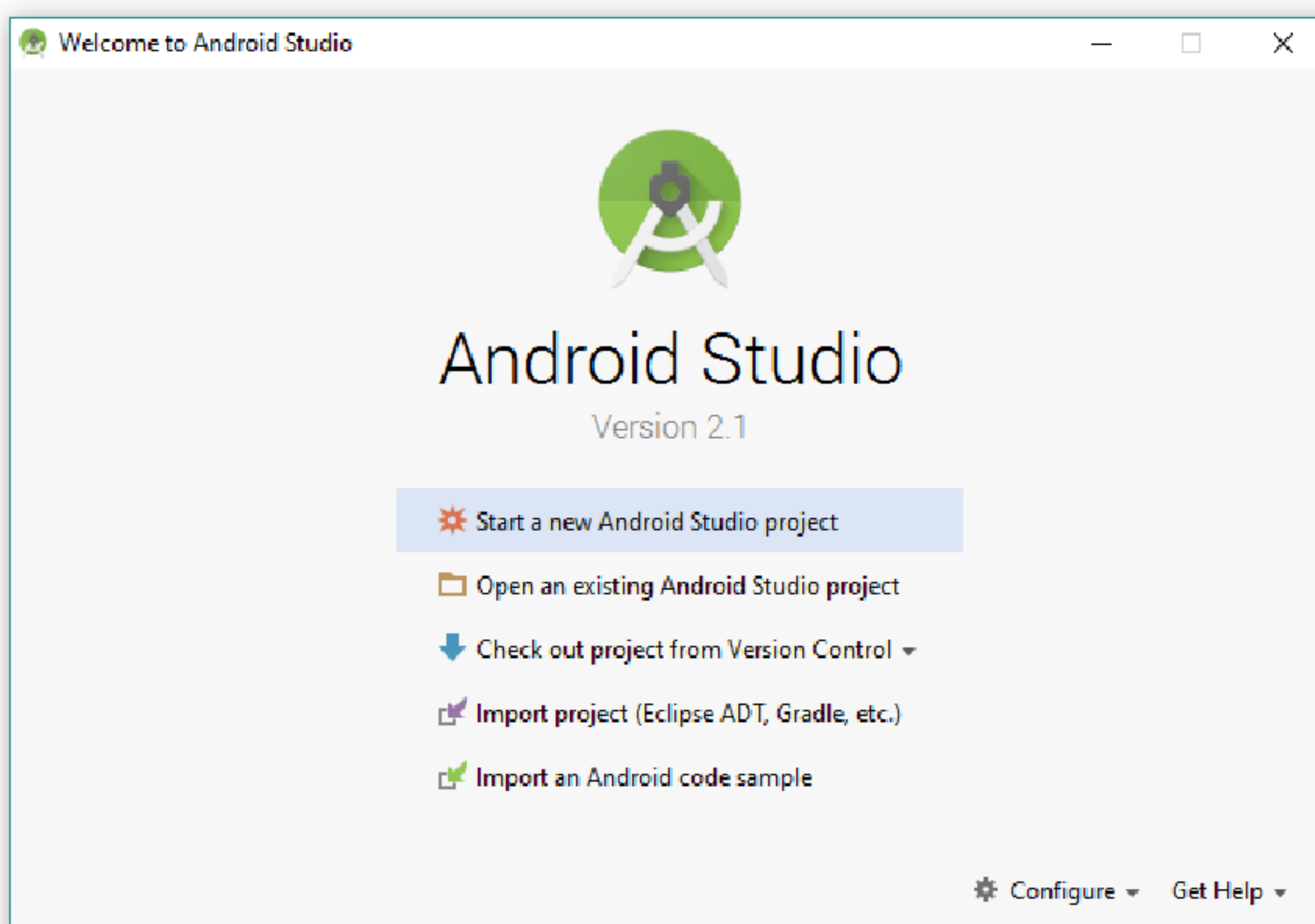


Рис. 1.8. Начальное окно AndroidStudio

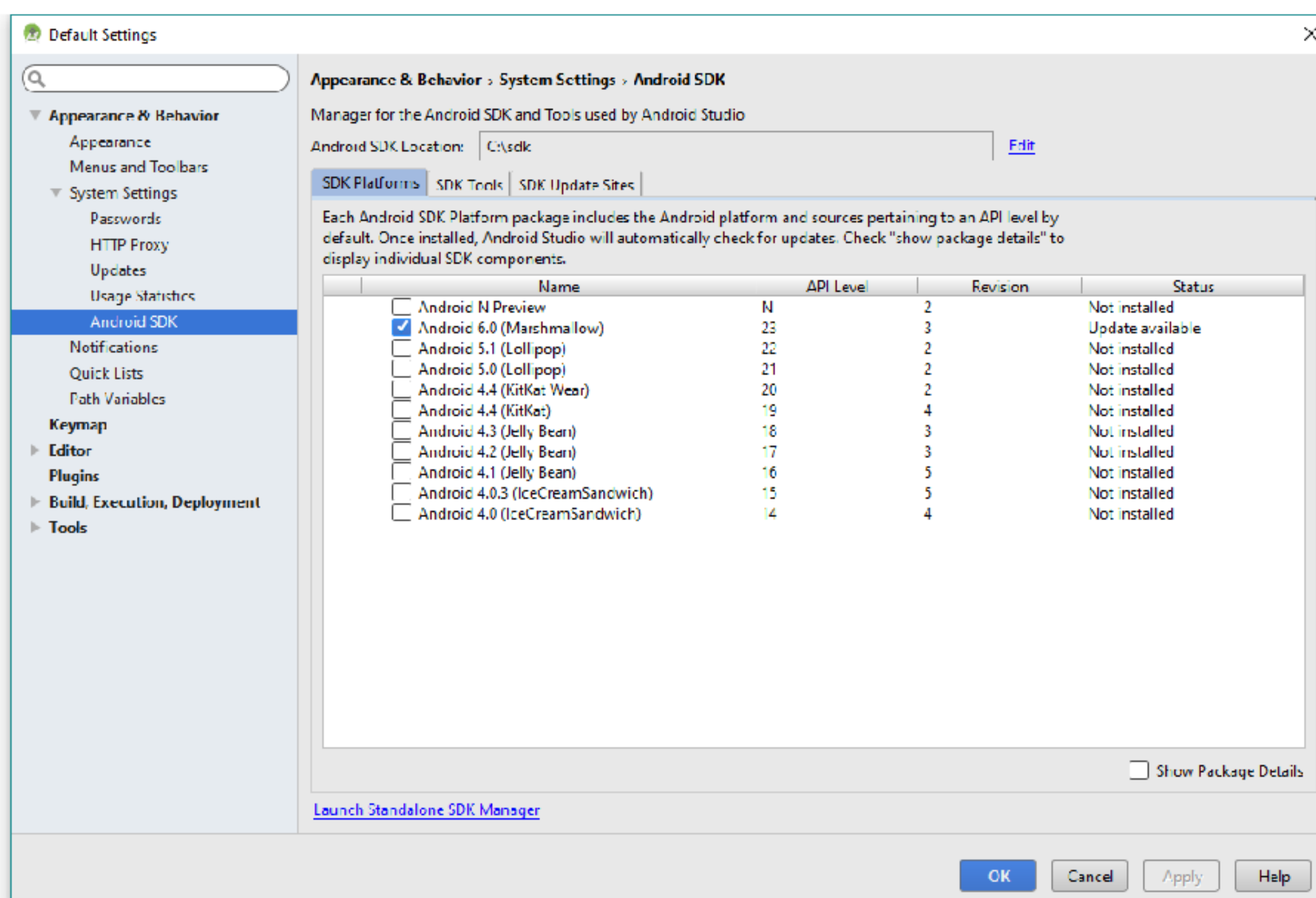


Рис. 1.9. Менеджер платформ

Во вкладке SDKTools (рис. 1.10) необходимо выбрать инструменты, которые могут понадобиться при разработке приложений. Кроме того, в большинстве случаев, если какой-либо из необходимых пакетов не был установлен, AndroidStudio укажет на необходимость установки требуемого пакета.

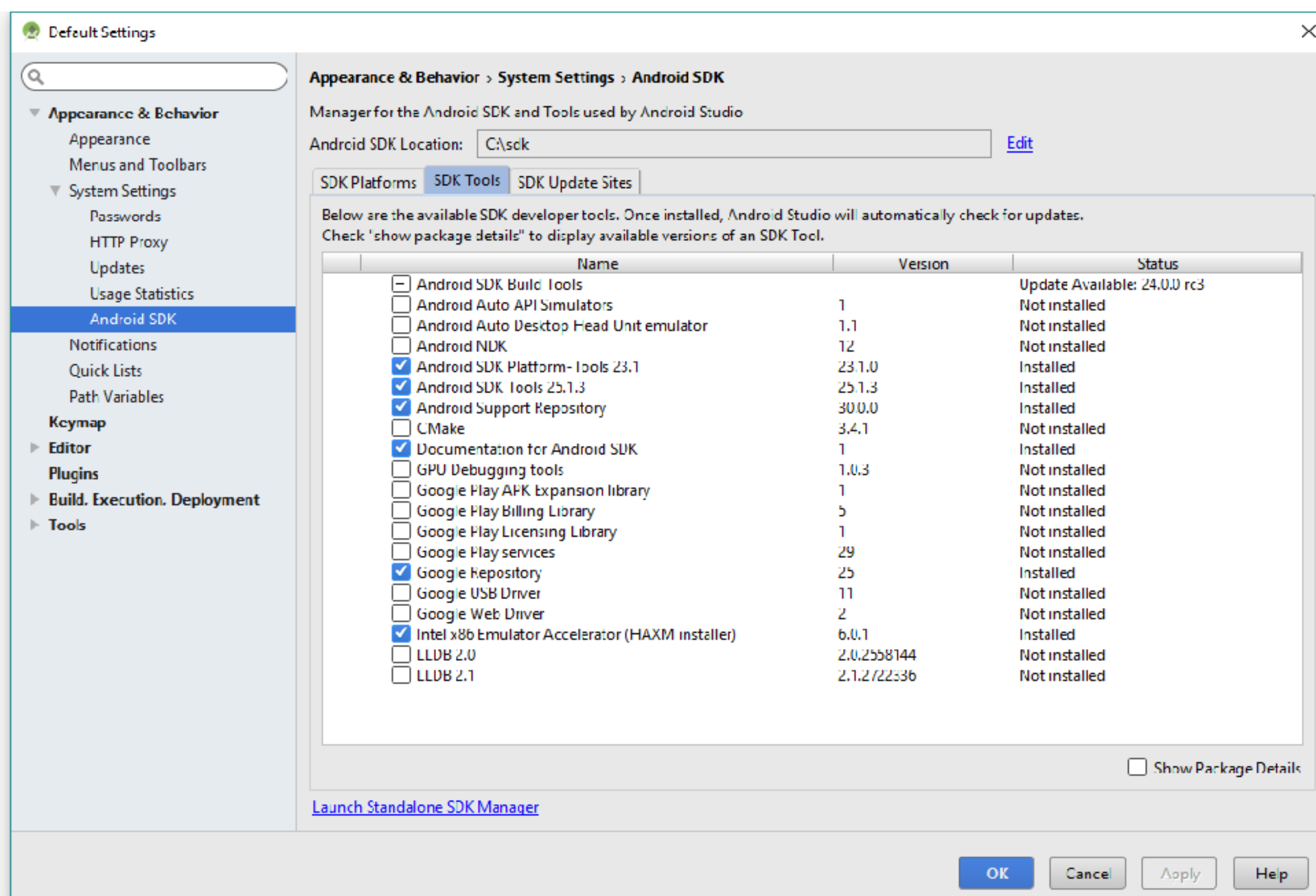


Рис. 1.10. Менеджер инструментов

Для того чтобы скачать какой-либо пакет, необходимо установить галочку напротив пункта, в котором указано название этого пакета, и подтвердить выбор нажатием кнопки ОК (рис. 1.11).

В появившемся окне отразится текст лицензионного соглашения, которое необходимо принять. Для этого выбираем пункт Ассерт (рис. 1.12) и нажимаем Next.

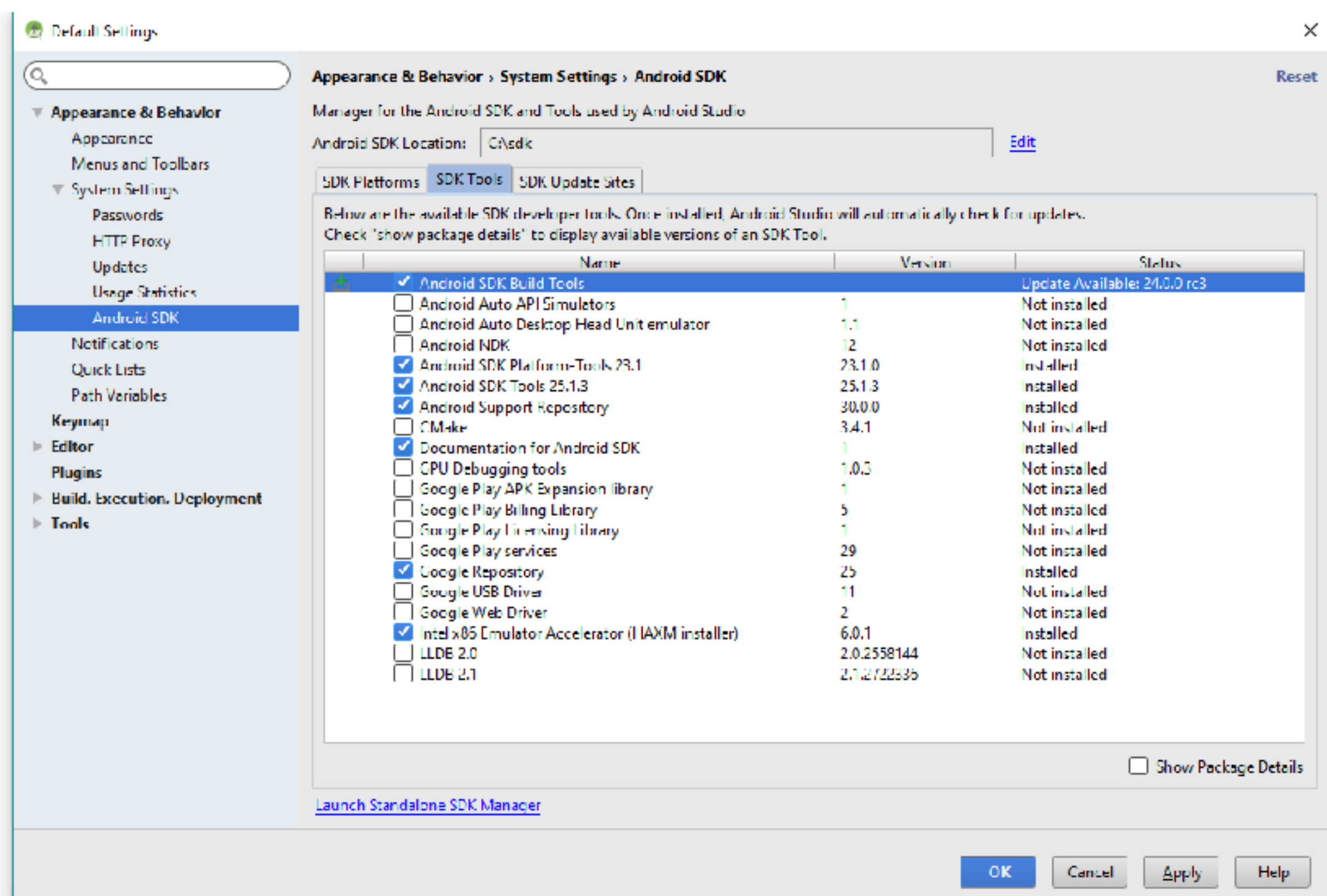


Рис. 1.11. Выбор установочных библиотек

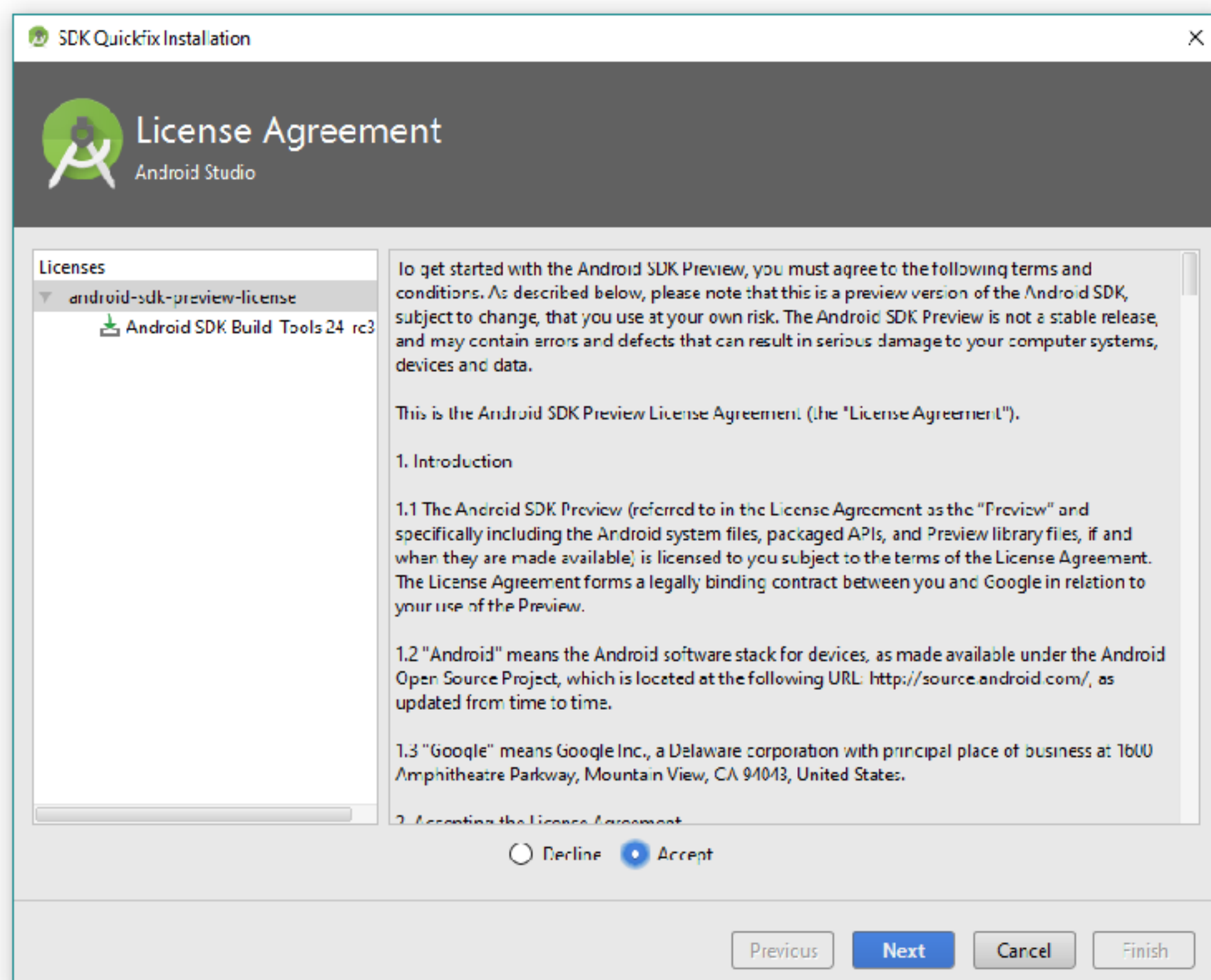


Рис. 1.12. Лицензионное соглашение

Документ подписан
После нажатия на кнопку «Next» начнется скачивание, распаковка и установка
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Необходимого вам компонента SDK.
Действителен: с 20.08.2021 по 20.08.2022

Рабочее окружение готово для работы.

Вопросы:

1. Перечислите все версии ОС Android.
2. Какие средства необходимы для начала разработки под ОС Android?
3. Перечислите основные преимущества и недостатки ОС Android.
4. Под какими ОС возможно разрабатывать программное обеспечение под ОС Android?

Лабораторная работа 2. Создание нового проекта

Для создания проекта необходимо из контекстного меню среды разработки AndroidStudio выбрать File → NewProject. Появится диалоговое окно (рис. 2.1).

Скриншот диалогового окна "Create New Project" в Android Studio. В окне "Configure your new project" заполнены следующие поля:

- Application name: My Application (отмечено как **имя приложения**)
- Company Domain: omgtu.ru (отмечено как **имя домена компании в сети Интернет**)
- Package name: ru.omgtu.myapplication (отмечено как **структура пакета**)
- Project location: /Users/gritsay/development/MyApplication (отмечено как **расположение проекта**)

В нижней части окна расположены кнопки: Cancel, Previous, Next (выделена), Finish.

В нем необходимо указать имя проекта, наименование идентификатора приложения – в нашем случае omgtu.ru (обычно указывается по имени сайта организации), при этом наименование Package name формируется автоматически – в обратной последовательности имени домена с именем проекта в конце строки. Нажимаем кнопку Next и попадаем на диалоговое окно выбора версии SDK, под которую будет собираться проект (рис. 2.2).

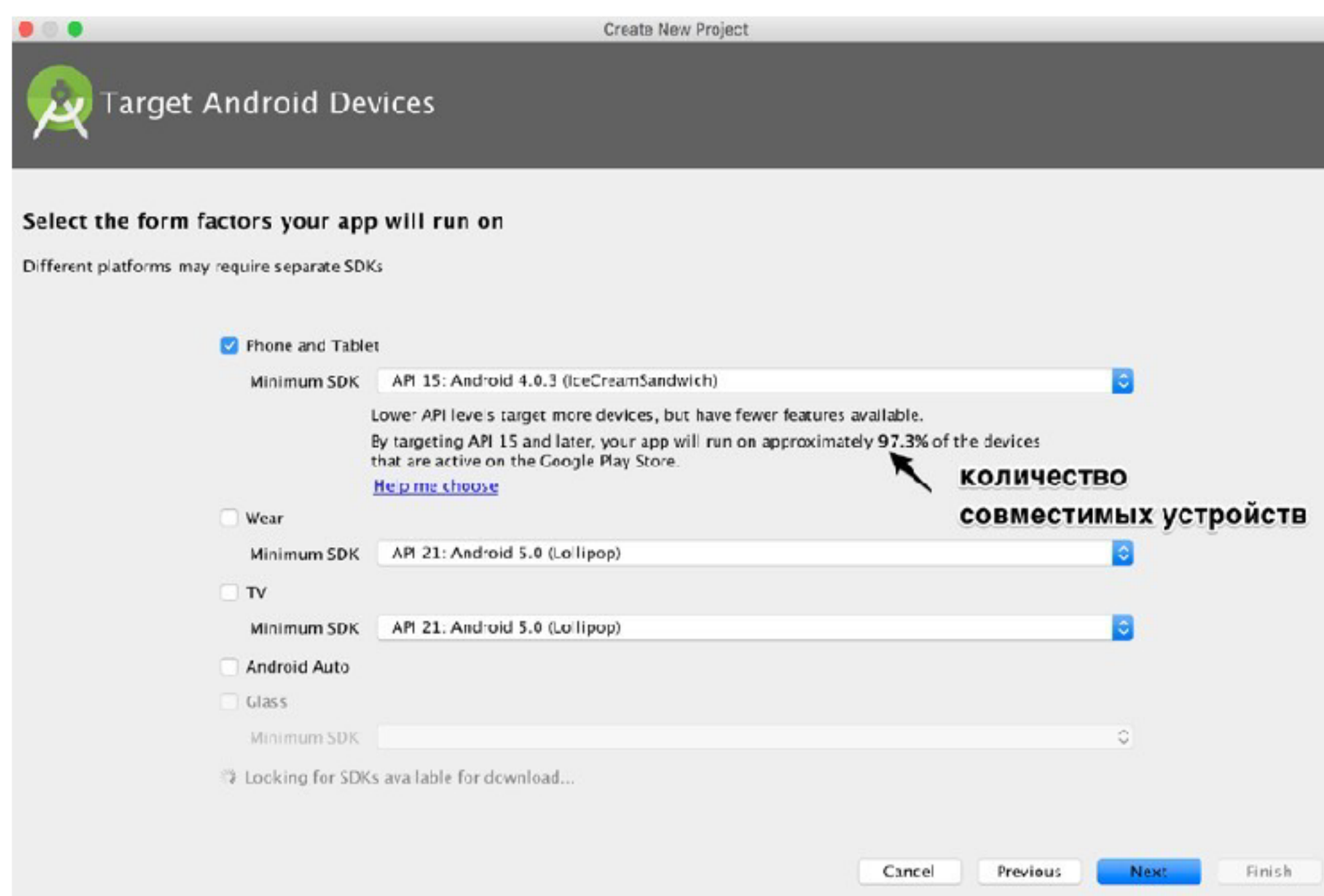
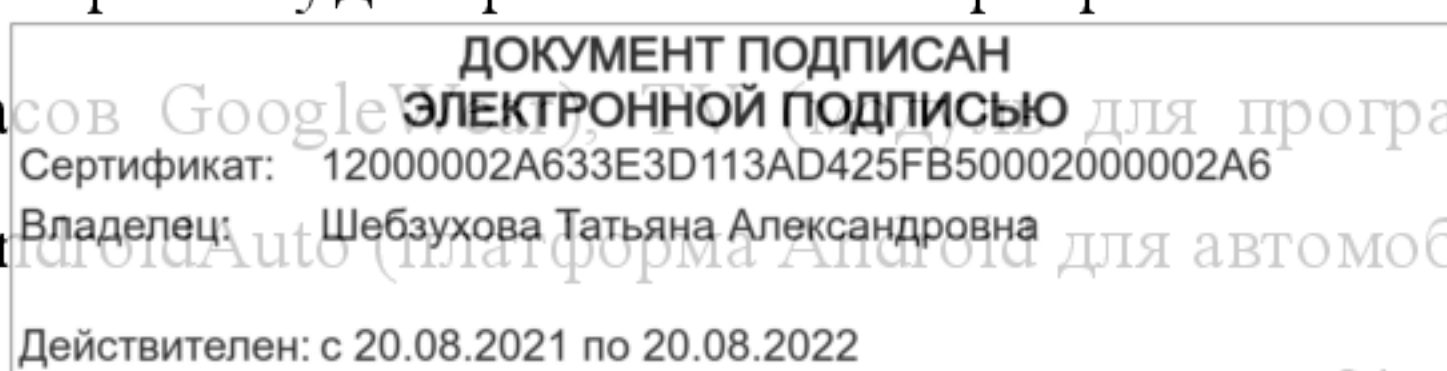


Рис. 2.2. Диалоговое окно выбора SDK для разработки

В нашем случае система сама предложила осуществить поддержку (начиная с операционной системы Android 4.0.3). Обратите внимание на строку, представленную ниже, в которой написано, что выбранная конфигурация будет совместима с 97,3 % существующих Android устройств. Также в интерактивном режиме можно посмотреть, какие еще существуют варианты, кликнув на надписи «Help me choose». В данном окне возможно выбрать дополнительные модули, для которых будет реализована программа Wear (платформа для программирования часов Google Wear OS) и AndroidTV (платформа для программирования устройств AndroidTV) и Android Auto (платформа Android для автомобилей).



Нажав кнопку Next, переходим на следующий экран, где представлена возможность выбора перенастроенной конфигурации элементов управления Activity (рис. 2.3).

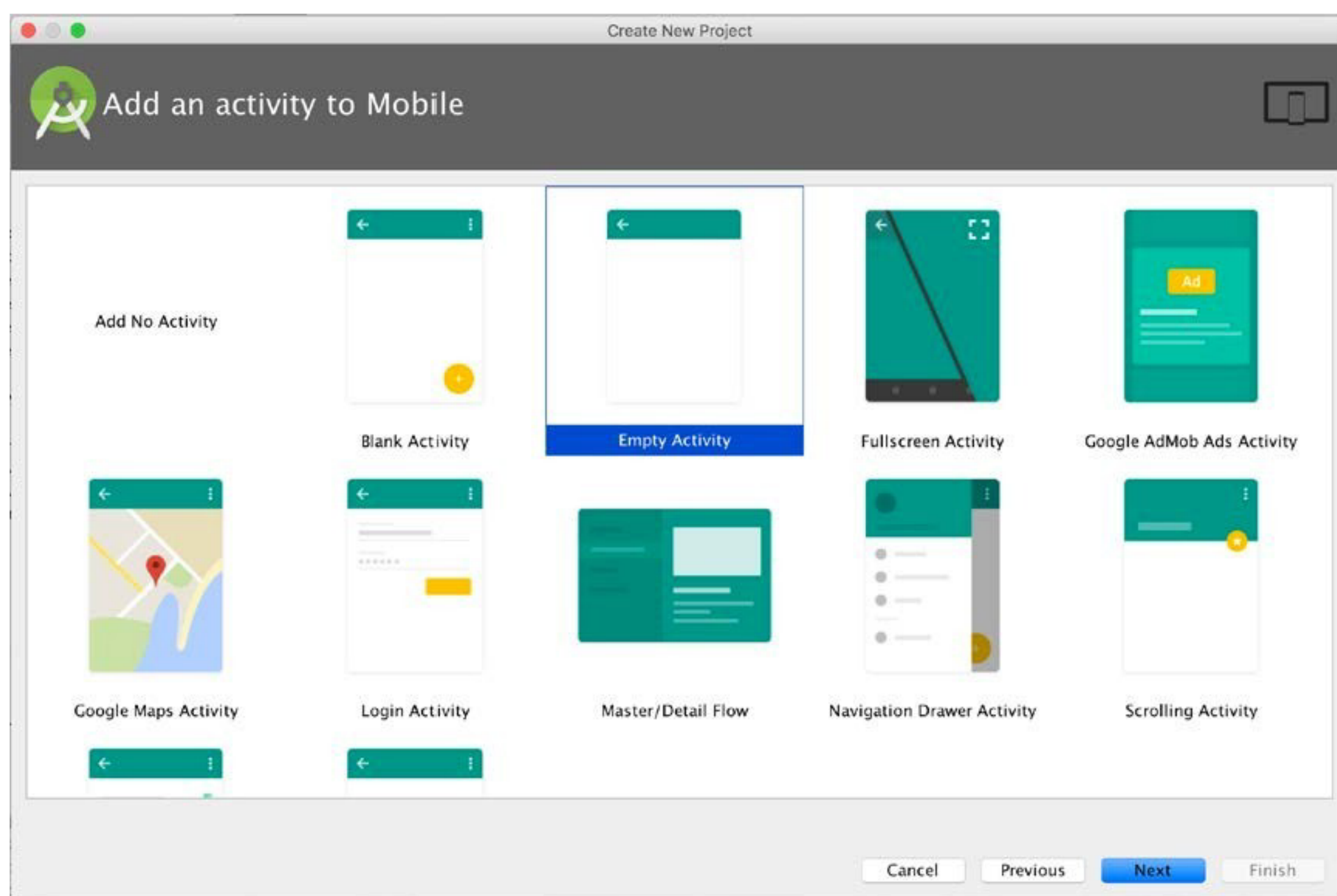


Рис. 2.3. Диалоговое окно выбора элементов Activity

На экране выбора элементов Activity необходимо указать один из подходящих вариантов с расположением элементов управления. В нашем случае элементы управления будут добавлены вручную, поэтому выбираем экранную форму без каких-либо элементов управления «EmptyActivity». Структура созданного проекта отображается в древовидном виде (рис. 2.4), что позволяет упростить навигацию по файлам.

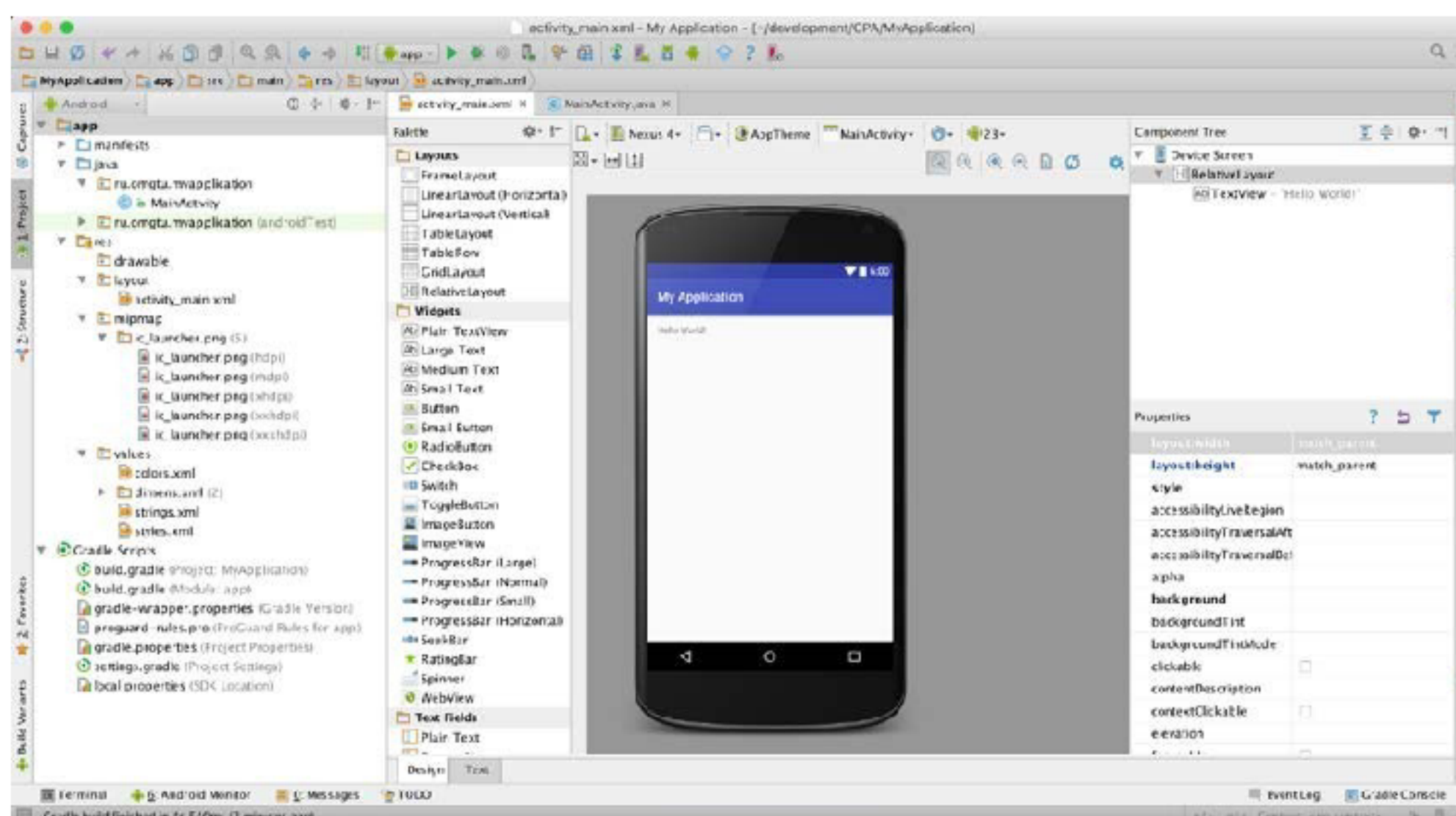


Рис. 2.4. Структура проекта в AndroidStudio

В папке GradleScripts представлены файлы конфигурации. Для проекта, созданного «по умолчанию», существует два файла конфигурации build.gradle – один включает в себя конфигурацию проекта, другой – конфигурацию модуля разрабатываемого приложения.

В папке ru.omg.tu.myapplication располагается единственный в нашем случае файл Activity, папка res содержит в себе все ресурсы приложения, настройки, константы, стили. Так в папке mipmap представлено 5 иконок приложения в формате .png для разных разрешений экранов.

Чтобы проект можно было откомпилировать и запустить, в случае, если используется не прямое подключение к интернету, а через прокси-сервер, необходимо внести в файл gradle.properties следующие строки:

#для протокола
`httpssystemProp.http.proxyHost=hostnamesystemProp.http.proxyPort=hostport`
 #для протокола
`httpssystemProp.https.proxyHost=hostnamesystemProp.https.proxyPort=hostport`

где hostname – имя хоста прокси сервера, hostport – номер порта, по которому

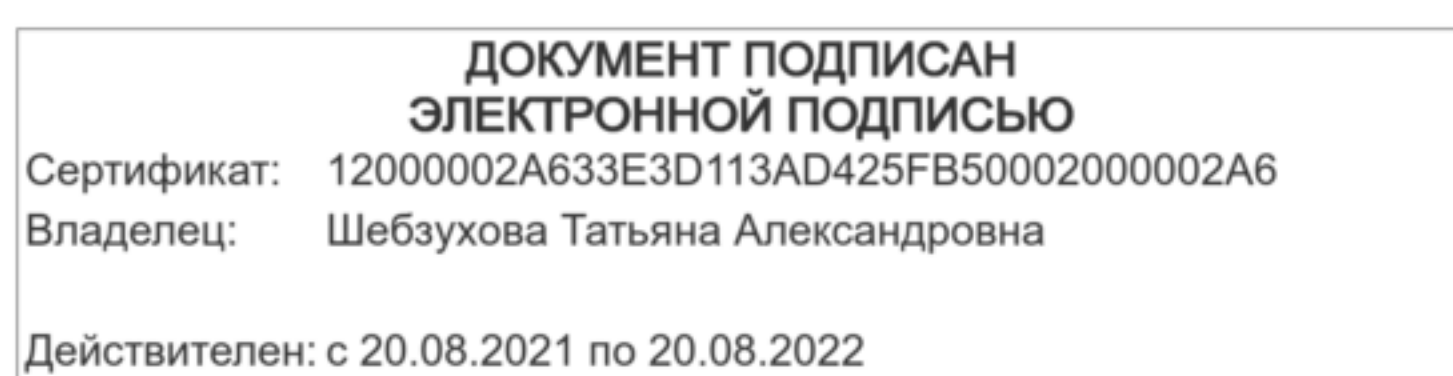
Доступен проект
 ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ
 Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен: с 20.08.2021 по 20.08.2022

build.gradle скрипта, который отображает

команды для сборки проекта. В нашем случае данный файл был сгенерирован автоматически при создании проекта. В секции `defaultConfig` определен идентификатор приложения (каждое приложение, находящееся в магазине приложений PlayMarket, имеет свой уникальный идентификатор), минимальная версия поддерживаемого SDK (`minSdkVersion`). Этот параметр характеризует версию операционной системы, на которой будет работать приложение. Если версия операционной системы устройства будет меньше, чем версия, определенная этим параметром, программа будет недоступна для скачивания с электронного магазина (в случае ее распространения средствами PlayMarket). Параметр `buildToolsVersion` определяет версию SDK пакета ADT, которая будет использоваться для компиляции проекта, а параметр `compileSdkVersion` определяет версию Android устройства, под которую будет собран билд. Свойства `versionCode` и `versionName` не обязательны, но больше как информационные параметры для программиста и системы распространения PlayMarket. Они позволяют не нарушать принципы нумерования версий программ. В свою очередь версионность систем может задаваться релизной политикой, которая определена документально при разработке той или иной системы.

Секция `buildTypes` описывает специальные параметры сборки билда, его тип и опции обфускации.

Секция `dependencies` описывает дополнительные пакеты (зависимости), которые используются в проекте. Данные зависимости можно добавить как вручную в файл `gradle.build`, так и в автоматическом режиме поиском по существующим библиотекам в системе maven и выбором их в диалоговом интерфейсном окне (рис. 2.5).



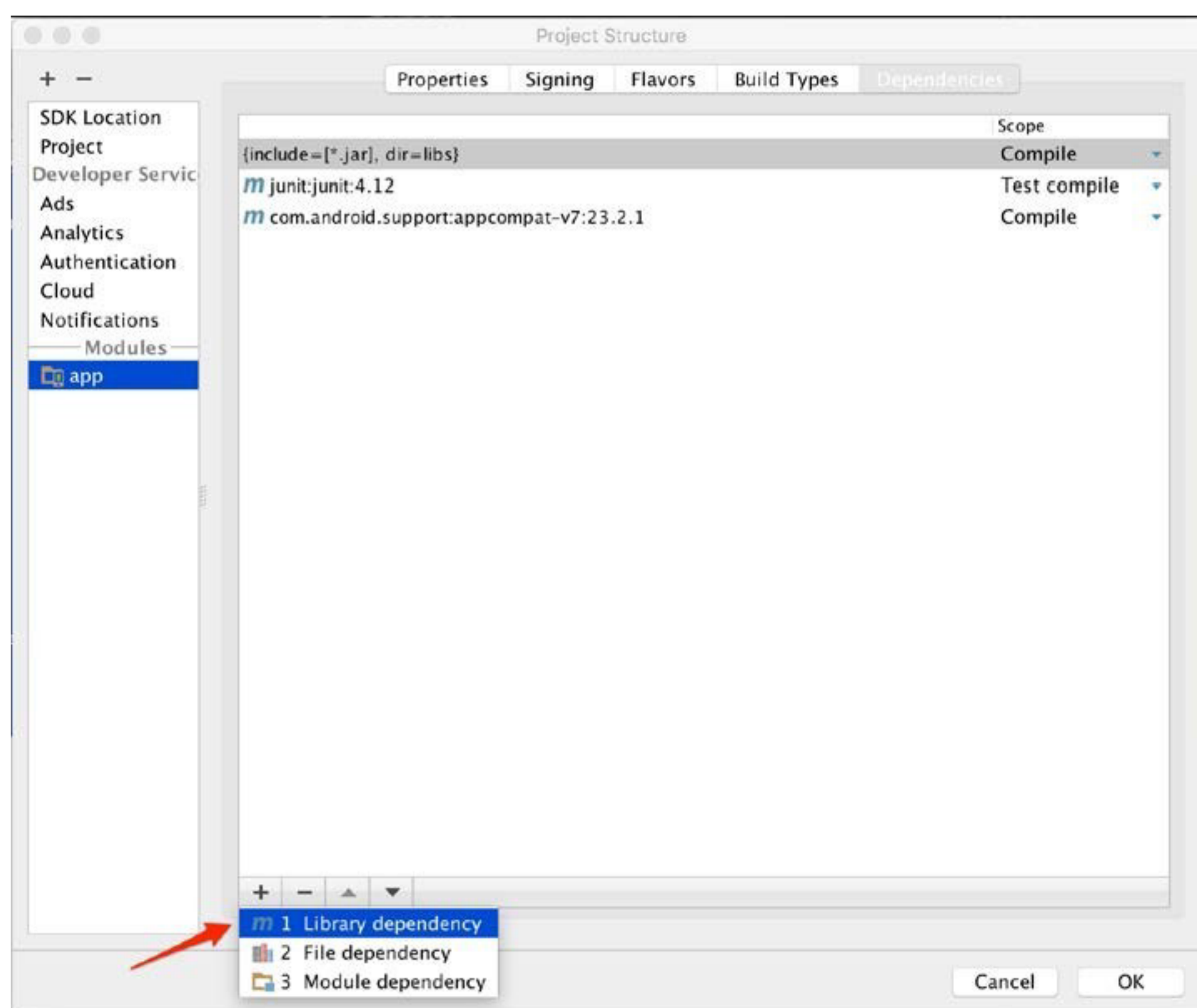


Рис. 2.5. Диалоговое окно добавления зависимостей

Для того чтобы вызвать это диалоговое окно, необходимо выполнить Build → Edit Libraries and Dependencies.

В диалоге нужно нажать «+» и выбрать Library dependency. Разумеется, поиск будет возможен только при наличии доступа в сеть Интернет. В других случаях, например, при наличии предварительно скачанной библиотеки, можно воспользоваться опцией File dependency или Module dependency (зависит от типа библиотеки).

При написании программы под ОС Android существует еще один важный файл конфигурации, который называется AndroidManifest.xml. Пример такого файла приведен в прил. Б. Основное, что на сейчас следует усвоить, – это принцип осуществления входа в приложение, т. е. явное указание основного элемента Activity, который будет выполнен после запуска приложения. В примере это MainActivity, с использованием интент-фильтров и android.intent.category.LAUNCHER.

Таким образом, рассмотрев основные возможности проекта для операционной системы Android, мы можем его скомпилировать. Для этого необходимо выполнить Build →BuildAPK.

Вопросы:

1. Разработку под какие типы устройств поддерживает IDEAndroidStudio?
2. Перечислите все основные файлы проекта, созданного по умолчанию.
3. Каким образом возможно добавление сторонней библиотеки в разрабатываемоеприложение?
4. Что такое арк-файл? Как егополучить?

Лабораторная работа 3. Жизненный цикл Activity

Как мы рассматривали выше, одним из ключевых компонентов системы Android является компонент Activity. Жизненный цикл этого компонента контролируется операционной системой Android, а сам компонент взаимодействует с пользователем посредством графического интерфейса. Например, если пользователь в определенный момент времени работает с каким-либо окном в приложении, этому окну система Android присваивает наивысший приоритет, и наоборот, если пользователь перевел приложение в режим бэкграунд и не использует его, система решает, что приложение необходимо остановить, чтобы выделить дополнительные ресурсы для других активных приложений. В общем случае жизненный цикл Activity представлен на рис.3.1.

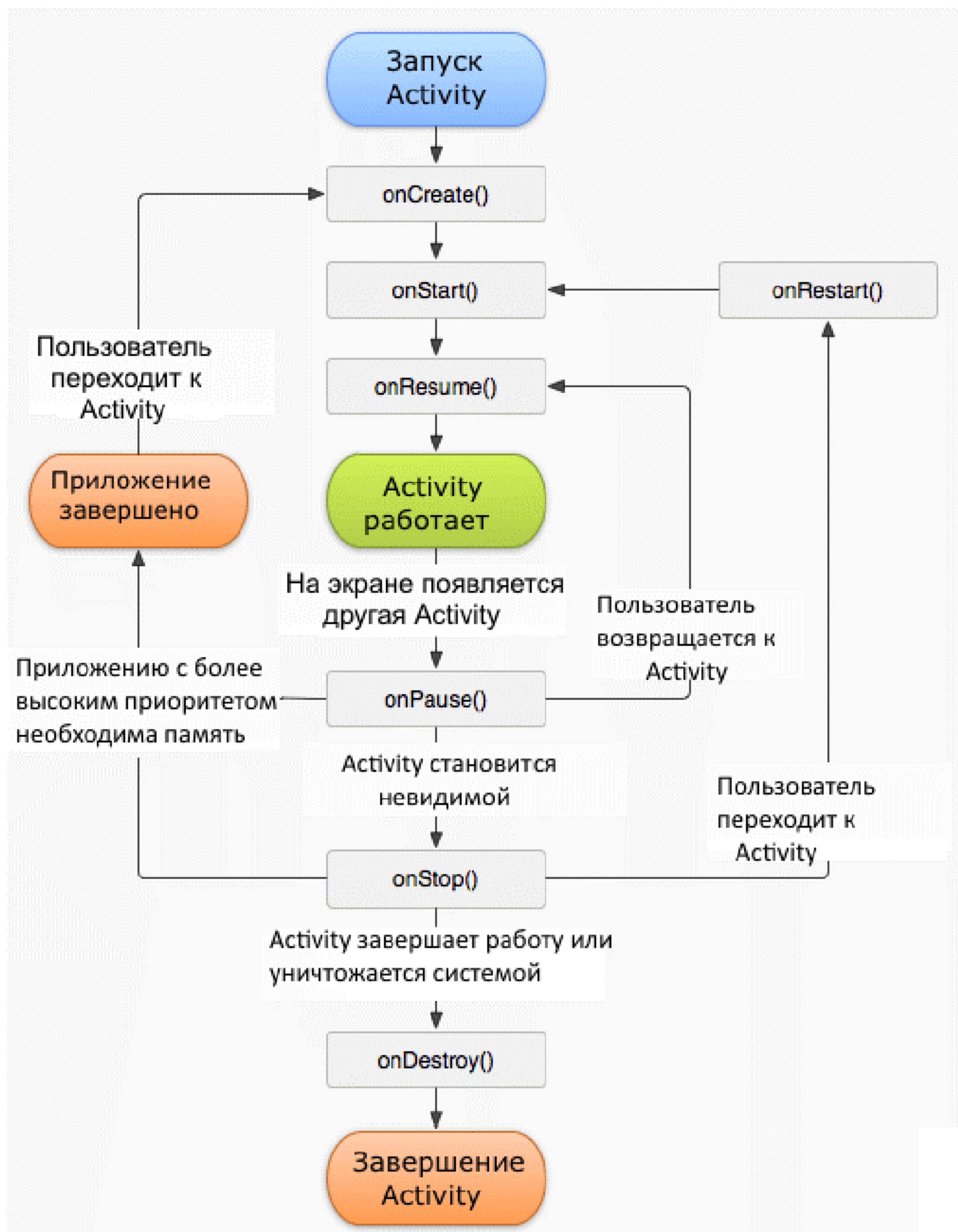


Рис. 3.1. Жизненный цикл Activity

- `protected void onCreate();`
- `protected void onStart();`
- `protected void onRestart();`
- `protected void onResume();`
- `protected void onPause();`
- `protected void onStop();`
- `protected void onDestroy();`

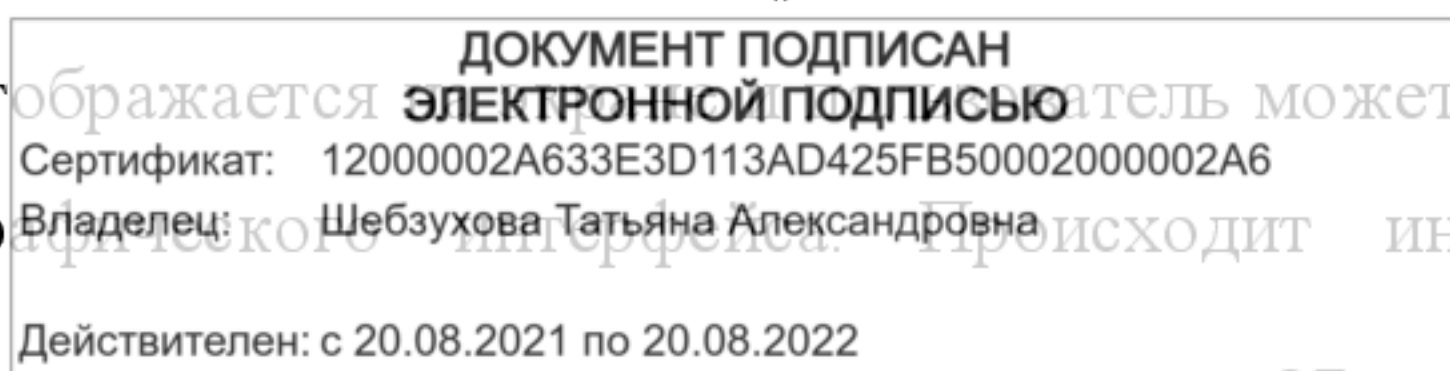
Метод `onCreate()` всегда вызывается при создании и перезапуске Activity (рис. 3.2). Нотация `@Override` обозначает, что переопределяется стандартный метод класса `Activity.class`. Внутри этого метода интерфейс связывается с классом Activity посредством метода `setContentView()`. Этот метод принимает в параметре объект `Bundle`, содержащий текущее состояние пользовательского интерфейса, сохраненное при последнем вызове обработчика `onSaveInstanceState`. Также инициализируются статические данные.

```
@Override
public void onCreate() {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

Рис. 3.2. Пример переопределения метода `onCreate()`

Метод `onStart()` запускается при старте Activity вслед за методом `onCreate()`, но не всегда, поскольку метод `onStart()` также вызывается при возобновлении работы приложения после того, как оно было остановлено. При остановке вызывается метод `onStop()`, это происходит в момент скрывания активности, например, когда открывается другое приложение.

Метод `onResume()` вызывается после метода `onStart()`, когда окно отображается. Пользователь может с ним взаимодействовать посредством графического интерфейса. Происходит инициализация переменной `mConnect`,



которая устанавливает соединение с каким-либо устройством (рис. 3.3). Предположительно это соединение было закрыто в методе onPause().

```
@Override  
public void onResume() { super.onResume();  
    if (mConnect == null) { connectDevice();  
    }  
}
```

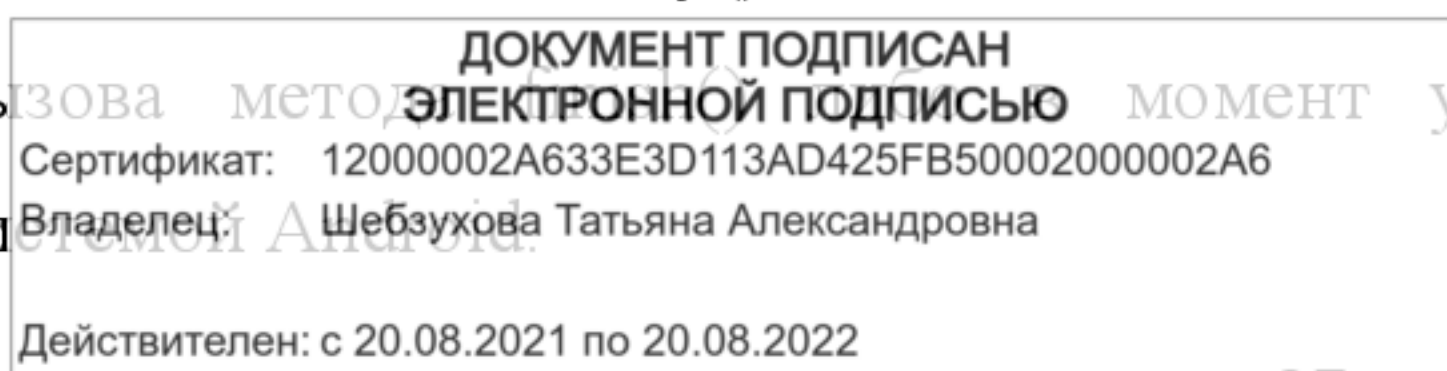
Метод onPause() вызывается, если открывается новая Activity, а текущая помещается в стек неактивных. В этом методе необходимо высвободить ресурсы, занятые текущей активностью – выключить камеру, остановить воспроизведение потокового видео и т. д. Также важно в этом методе размещать быстрый легковесный код, в противном случае, если, например, при вызове данного метода осуществлять запись в БД каких-либо больших данных, процесс перехода к другой активности будет замедлен, что негативно скажется на отзывчивости программного интерфейса.

Метод onStop() вызывается, когда окно становится невидимым пользователю. Это произойдет в том случае, если была запущена другая активность. В этом методе можно размещать операции, которые требуют длительного времени на выполнение, например, обновление пользовательского интерфейса, чтение показаний датчика GPS. При нехватке памяти система Android может сама уничтожить скрытую Activity, минуя метод onStop() с вызовом метода onDestroy().

Метод onRestart() вызывается в тот момент, когда окно возвращается в приоритетный режим после вызова метода onStop(). Т. е. в момент, когда активность была остановлена и снова запущена. Метод onRestart() предшествует вызову метода onStart(). Данный метод можно использовать при повторном вызове активности для восстановления ее параметров и элементов пользовательского интерфейса.

Метод onDestroy() вызывается по окончании работы Activity посредством

Вызова метода уничтожения экземпляра активности



Рассмотрим пример, представленный в прил. В. Необходимо создать новый проект и добавить в него два файла: MainActivity.java и activity_main.xml. При запуске и остановке приложения нужно следить за консолью, в которую переопределенные методы будут писать сообщения (см. таблицу).

Порядок вызова методов в зависимости от действия пользователя

Действие пользователя	Порядок вызовов
Запуск приложения	onCreate() → onStart() → onResume()
Нажатие системной кнопки «Назад»	onPause() → onStop() → onDestroy()
Нажатие системной кнопки «Домой»	onPause() → onStop()
После нажатия кнопки «Домой» повторный вызов приложения	onRestart() → onStart() → onResume()
Экран телефона выключен	onPause() → onStop()
Экран телефона повторно включен	onRestart() → onStart() → onResume()

Понимание жизненного цикла Activity очень важно при разработке под ОС Android, так как это позволяет правильно строить логику приложения.

Вопросы:

1. Что такое элементActivity?
2. Перечислите элементы жизненного циклаActivity.
3. Какие методы Activity вызываются при смене ориентации устройства?
4. Каким образом происходит связывание интерфейса сActivity?

Лабораторная работа 4. Использование ресурсов приложения

В корне проекта существует директория ресурсных файлов res. Ресурсы хранятся в файлах в директории res/values. При создании приложения в файл strings.xml добавляется первый ресурс

app_name, значение которого есть имя приложения. В данном файле нужно описывать строковые константы, которые будут использоваться в приложении (рис. 4.1). Требуется это для мультиязычности приложения и простоты доработки его в дальнейшем.

```
<resources>
    <string name="app_name">Hello world</string>
    <string name="sample">test string</string>
</resources>
```

Рис. 4.1. Пример файла strings.xml

При объявлении ресурса в xml файле имя должно быть уникальным, в классе R создается константа с таким же именем, чтобы мы могли иметь доступ к строковым элементам (рис.4.2).

```
public static final class string {
    public static final int abc_action_bar_home_description=0x7f060000;
    public static final int abc_action_bar_home_description_format=0x7f06000d;
    public static final int abc_action_bar_home_subtitle_description_format=0x7f06000e;
    public static final int abc_action_bar_up_description=0x7f060001;
    public static final int abc_action_menu_overflow_description=0x7f060002;
    public static final int abc_action_mode_done=0x7f060003;
    public static final int abc_activity_chooser_view_see_all=0x7f060004;
    public static final int abc_activitychooserview_choose_application=0x7f060005;
    public static final int abc_search_hint=0x7f06000f;
    public static final int abc_searchview_description_clear=0x7f060006;
    public static final int abc_searchview_description_query=0x7f060007;
    public static final int abc_searchview_description_search=0x7f060008;
    public static final int abc_searchview_description_submit=0x7f060009;
    public static final int abc_searchview_description_voice=0x7f06000a;
    public static final int abc_shareactionprovider_share_with=0x7f06000b;
    public static final int abc_shareactionprovider_share_with_application=0x7f06000c;
    public static final int abc_toolbar_collapse_description=0x7f060010;
    public static final int app_name=0x7f060011;
    public static final int sample=0x7f060012;
    public static final int status_bar_notification_info_overflow=0x7f060013;
}
```

Рис. 4.2. Класс R

Чтобы использовать значение ресурса, требуется в свойствах элемента ввести следующую конструкцию @string/имя_ресурса (рис. 4.3).

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ru.omgtu.nabiullin.helloworld">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

    </application>

</manifest>
```

Рис. 4.3. Пример файла AndroidManifest.xml

Кроме того, можно создать свой файл с константами, для этого следует нажать правой кнопкой на директорию values, выбрать New→Valuesresourcefiles и в созданном файле описать необходимое (рис.4.4).

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="tvTopText">Верхний текст</string>
    <string name="btnTopText">Верхняя кнопка</string>
    <string name="tvBottomText">Нижний текст</string>
    <string name="btnBottomText">Нижняя кнопка</string>
    <color name="llTopColor">#336699</color>
    <color name="llBottomColor">#339966</color>
</resources>
```

Рис. 4.4. Пример своего ресурсного файла

В нашем случае строковые ресурсы string и color находятся в одном файле, однако делать этого не стоит (здесь это сделано исключительно в учебных целях). Принято строковые константы и константы, которые задают цвет,

описывать в файле R.java. Следом можно убедиться в том, что константы созданы. Для этого достаточно посмотреть файл R.java.

Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

Если необходимо использовать значения ресурсов в коде программы, то их можно получить с помощью следующего метода:

```
getResources().getString(R.string.app_name);
```

Имена ресурсов, глобальных для всех файлов, хранятся в папке `res/values`. Т.е. невозможно в разных файлах создать ресурс с одним именем и типом.

Имена файлов ресурсов могут быть произвольными, и файлов можно создавать сколько угодно. В `R.java` попадут все ресурсы из этих файлов.

Вопросы:

1. Для чего необходимы файлы ресурсов приложения?
2. Как получить доступ к элементу файла ресурса приложения?
3. Можно ли вносить какие-либо изменения в файл `R.java`?

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

Методические указания

для обучающихся по организации и проведению самостоятельной работы
по дисциплине «ПРОГРАММИРОВАНИЕ МОБИЛЬНЫХ УСТРОЙСТВ»
для студентов направления подготовки **09.03.02 Информационные системы и
технологии**
направленность (профиль) **Информационные системы и технологии обработки
цифрового контента**

Пятигорск, 2022

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

1. Общие положения

Самостоятельная работа – планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

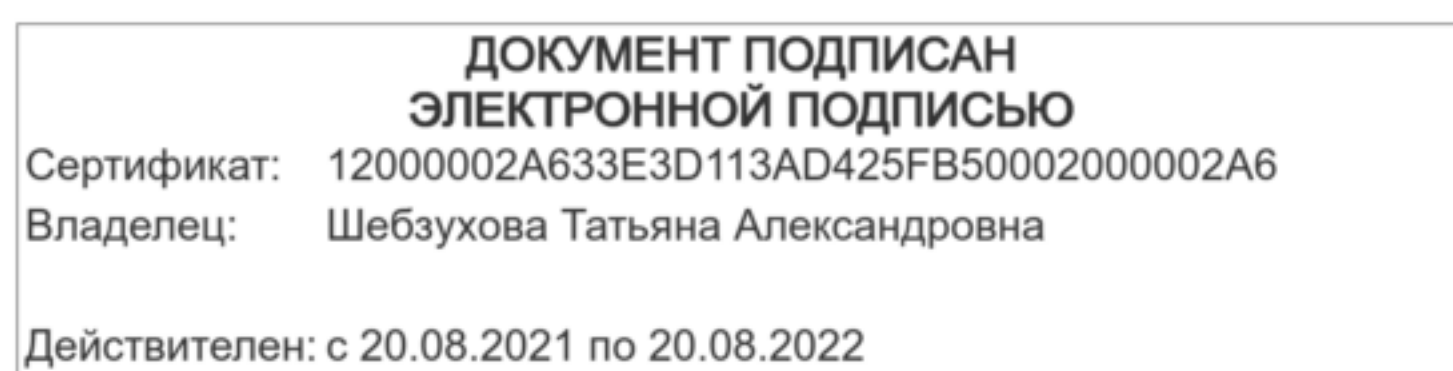
К основным видам самостоятельной работы студентов относятся:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- написание докладов;
- подготовка к семинарам, практическим и лабораторным работам, их оформление;
- составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);
- выполнение учебно-исследовательских работ, проектная деятельность;
- подготовка практических разработок и рекомендаций по решению проблемной ситуации;
- выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;
- компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;
- выполнение курсовых работ (проектов) в рамках дисциплин;
- выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).



2.Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование универсальных компетенций.

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельной работы и лабораторных занятий.

3.Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе (астр.)		
			СРС	Контактная работа с преподавателем	Всего
7 семестр					
ПК-5(ИД-1, ИД-2) ПК-6(ИД-1, ИД-2) ПК-13(ИД-3, ИД-4)	Самостоятельное изучение литературы и источников	Собеседование	76,32	8,48	84,8
ПК-5(ИД-1, ИД-2) ПК-6(ИД-1, ИД-2) ПК-13(ИД-3, ИД-4)	Самостоятельное изучение литературы и источников	Собеседование	1,08	0,12	1,2

Действителен: с 20.08.2021 по 20.08.2022

ИД-2) ПК-13(ИД-3, ИД-4)					
ПК-5(ИД-1, ИД-2) ПК-6(ИД-1, ИД-2) ПК-13(ИД-3, ИД-4)	Написание реферата/доклада	Защита доклада	9	1	10
Итого за 7 семестр			86,4	9,6	96
Итого			86,4	9,6	96

4.Порядок выполнения самостоятельной работы студентом

4.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют **три установки в чтении научного текста:**

1. **Информационная установка** – найти, выделить искомую информацию (например, для решения задачи).
2. **Усваивающая установка** – усвоить информацию полностью или частично.
3. **Критическая установка** – критически проанализировать материал и т.п.

Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта.

2. Выделите главное, составьте план.

3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора.

4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

4.2. Методические рекомендации по подготовке к практическим занятиям

Для того чтобы практические занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на лабораторных занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно для дальнейшего усвоения материала).

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы

(задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

4.3. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала – умение отвечать на вопросы для собеседования.

Вопросы для собеседования

Базовый уровень

1. Понятие защищенной информационной системы.
2. Свойства защищенной ОС.
3. Безопасность информационных систем в нормативных документах.
4. Классификация защищенности ОС по международным стандартам.
5. Политика безопасности, формальное представление политик.
6. Классификация изъянов защиты.
7. Категории изъянов защиты в ОС.
8. Мобильное программирование, платформы для разработки.
9. Основы работы с сенсорным вводом. Обработка нескольких касаний. Использование изображений.
10. Акселерометр и служба определения местоположения, вторичные потоки выполнения, обработка асинхронных операций и доступ к Веб-сервисам.
11. Особенности использования `pivot` и `panorama`.
12. Краткая история ОС Android.
13. Intel для Android: партнерство и инструментарий разработчика.
14. Архитектура приложений для Android. Ресурсы приложения. Пользовательский интерфейс. Инструментарий разработки приложений для Android.
15. Обзор шагов разработки типичного приложения под Android. Особенности разработки с использованием эмулятора. Отладка кода в эмуляторе и на реальных приложениях. Пример простейших программ Android-приложения. Запуск приложения на эмуляторе.
16. Тестирование приложения с помощью Dalvik Debug Monitor Server (DDMS).
17. Планирование покадровой анимации, анимирование, анимация шаблонов, видов, использование класса `Camera`.
18. Проверка безопасности, работа со службами, основанными на местоположении, использование сервисов AIDL.
19. Подготовка APK-файла для загрузки, локализация приложения, подготовка ярлыка приложения, подготовка APK-файла для загрузки, работа пользователя с Android Market.

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

20. Работа с инструментами Intel для оптимизации отладки Android-приложений
21. Программный стек мобильных платформ.
22. Архитектура мобильных приложений.
23. Приемы для улучшения производительности и уменьшения потребления памяти для мобильных приложений.
24. Основные составляющие манифеста приложения.
25. Жизненный цикл мобильного приложения.
26. Разработка интерфейсов, не зависящих от разрешения и плотности пикселей.
27. Для чего предназначены файлы с расширением .xap? 8. Что такое Zune? Применение.
28. Что такое графическая схема Silverlight?
29. Что представляет собой игровая платформа XNA?
30. Что входит в состав среды разработки Microsoft Visual Studio 2010 Express for Windows Phone?
31. Какие компоненты включает в себя Windows Phone SDK?
32. Что представляет из себя WP7 приложение?
33. Что представляет собой файл приложения ApplicationIcon.png?
34. Что представляет собой файл приложения Background.png?
35. Что представляет собой файл приложения SplashScreenImage.jpg?
36. Что представляет собой файл приложения App.xaml
37. Что представляет собой файл приложения App.xaml.cs:
38. Что представляет собой файл приложения AppManifest.xml?
39. Что представляет собой файл приложения AssemblyInfo.cs?
40. Что представляет собой файл приложения MainPage.xaml?
41. Что представляет собой файл приложения MainPage.xaml.cs?
42. Что представляет собой файл приложения WMAppManifest.xml?
43. Для чего предназначен язык разметки приложений XAML?

Повышенный уровень

44. Как выглядит эмулятор Windows Phone 7 в Visual Studio?
45. Какие существуют экраны в Windows Phone 7?
46. Перечислите основные кнопки приложений Windows Phone 7?
47. Какие существуют режимы экранов? 29. Как добавить новые страницы в приложение? 30. Какие существуют подходы для создания ссылок между страницами?
48. Что представляет собой класс NavigationService?
49. Как переопределить функциональность кнопки "Назад"?
50. Что такое Microsoft XNA?
51. Какие Вы знаете сервисы Xbox Live?
52. Как использовать шрифты Ascender Corporation в приложениях Windows Phone 7?
53. Что представляет собой класс Game1.cs?
54. Как определить координаты точки?
55. Что представляет собой метод Draw?
56. Что представляет собой метод Update?
57. Что такое приложение Rich Internet application?
58. Что такое Windows Presentation Foundation?
59. Опишите структуру класса MainPage.
60. Intents и Activities. Принципы работы Intent-фильтров.
61. Адаптеры и привязка данных.
62. Работа с интернет-ресурсами.
63. Диалоговые окна: создание и использование.
64. Курсоры, Content Values. Получение данных из SQLite.

65. Фоновые процессы, уведомления и сигнализация.
66. Фоновые процессы и карты сервисов: конфигурирование и использование.
67. Сенсорные датчики: SensorManager.

Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

68. Анимация и спецэффекты.
69. Акселерометр, датчик ориентации и компас: регулировка и программные функции.
70. Межпроцессное взаимодействие.
71. Основные права и полномочия для запуска приложений на устройстве.

4.4. Методические рекомендации по написанию научных текстов (докладов, рефератов, эссе, научных статей и т.д.)

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста доклада предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление.

Структура доклада:

- Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.

- Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ограничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.

- Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса

- Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.

- Приложение (при необходимости).

Требования к оформлению:

– текст с од...
**ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ**
 Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 – кегль шрифта 14;
 Действителен: с 20.08.2021 по 20.08.2022

- межстрочное расстояние 1,5;
- поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;
- реферат должен быть представлен в сброшюрованном виде.

Порядок защиты доклада:

На защиту доклада отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите доклада приветствуется использование мультимедиа-презентации.

Доклад оценивается по следующим критериям: соблюдение требований к его оформлению; необходимость и достаточность для раскрытия темы приведенной в тексте доклада информации; умение студента свободно излагать основные идеи, отраженные в докладе; способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

Критерии оценки:

Оценка «отлично» выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания доклада современные научные материалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

Оценка «хорошо» выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

Оценка «удовлетворительно» выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

Оценка «неудовлетворительно» выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

Описание шкалы оценивания

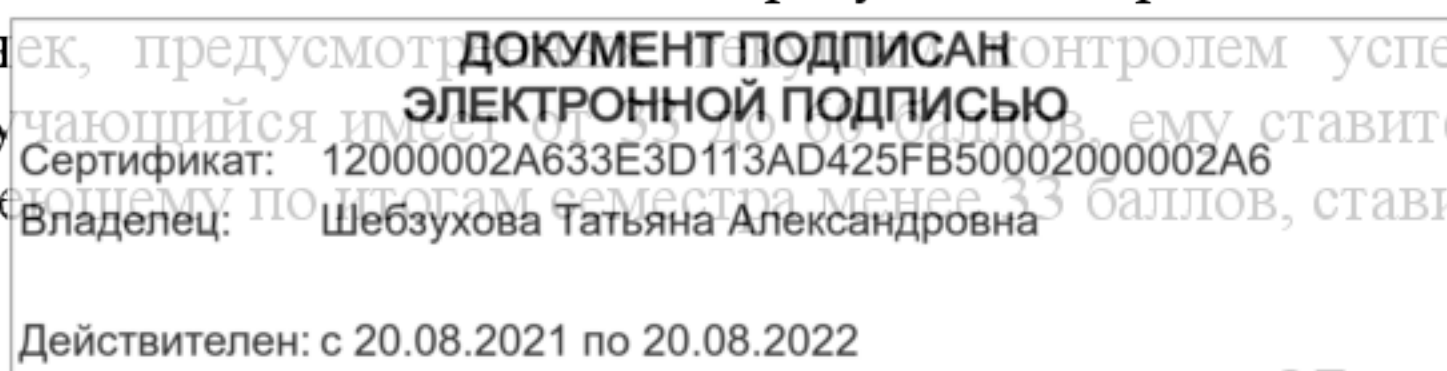
Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

4.5. Методические рекомендации по подготовке к зачетам

Процедура зачета как отдельное контрольное мероприятие не проводится, оценивание знаний обучающегося происходит по результатам текущего контроля.

Зачет выставляется по результатам работы в семестре, при сдаче всех контрольных точек, предусмотренных учебным планом, успешно сдавшим студенту ставится отметка «зачтено». Если по итогам семестра обучающийся имеет от 33 до 39 баллов, ему ставится отметка «зачтено». Обучающемуся, имеющему по итогам семестра менее 33 баллов, ставится отметка «не зачтено».



Количество баллов за зачет (Sзач) при различных рейтинговых баллах по дисциплине по результатам работы в семестре

Рейтинговый балл по дисциплине по результатам работы в семестре ($R_{сем}$)	Количество баллов за зачет ($S_{зач}$)
$50 \leq R_{сем} \leq 60$	40
$39 \leq R_{сем} < 50$	35
$33 \leq R_{сем} < 39$	27
$R_{сем} < 33$	0

Контроль самостоятельной работы студентов

Контроль самостоятельной работы проводится преподавателем в аудитории.

Предусмотрены следующие виды контроля: собеседование, оценка выполнения доклада и его презентации.

Подробные критерии оценивания компетенций приведены в Фонде оценочных средств для проведения текущей и промежуточной аттестации.

Список литературы для выполнения СРС

Основная литература:

Чезарини, Ф. Программирование в Erlang / Ф. Чезарини, С. Томпсон. - М. : ДМК Пресс, 2012. - 487 с. - (Функциональное программирование). - ISBN 978-5-94074-617-1 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=232035>

2. Операционная система Android / . - М. : МИФИ, 2012. - 64 с. - ISBN 978-5-7262-1780-2 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=231690>

3. Михеева, Е. В. Информационные технологии в профессиональной деятельности : учеб. пособие / Е.В. Михеева. - 14-е изд., стер. - М. : Академия, 2016. - 384 с.

4. Гохберг, Г. С. Информационные технологии : учебник / Г.С. Гохберг, А.В. Зафиевский, А.А. Короткин. - 9-е изд., перераб. и доп. - М. : Академия, 2014. - 240 с.

5. Хлебников, А. А. Информационные технологии : учебник / А. А. Хлебников. – М. : КноРус, 2014. – 472 с.

Дополнительная литература:

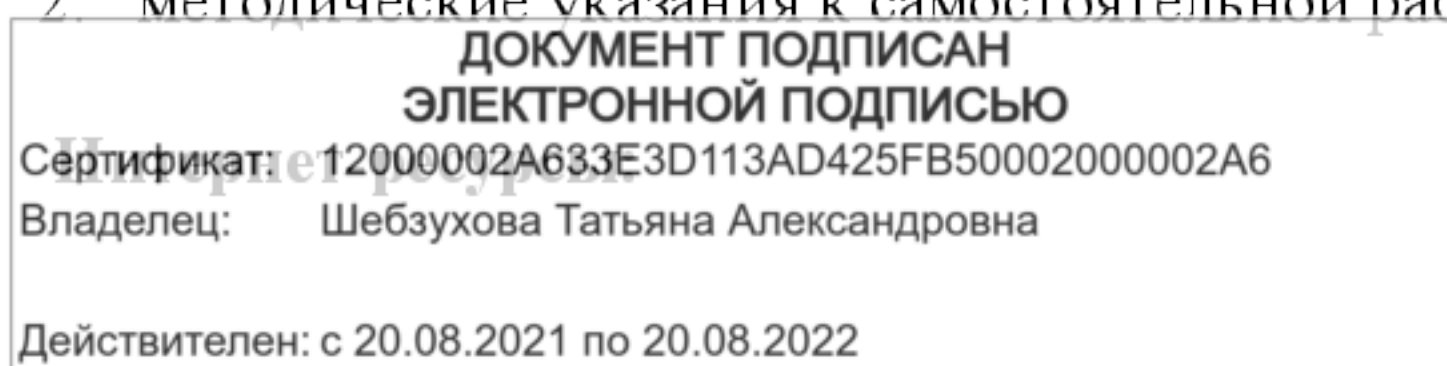
Васильев, А. Н. Java. Объектно-ориентированное программирование : [учеб. пособие] / А.Н. Васильев. - СПб. : Питер, 2012. - 400 с. : ил. - (Учебное пособие). - Прил.: с. 379-395. - Библиогр.: с. 377.

2. Винокуров Н.А. Практика и теория программирования. В 2 Кн. Кн. 1 Ч. I и II [Текст]: учеб. издание/ Н.А. Винокуров, А.В. Ворожцов. – М.: Физматкнига, 2008. – 192 с.

3. Винокуров Н.А. Практика и теория программирования. В 2 Кн. Кн. 2 Ч. III и IV [Текст]: учеб. издание/ Н.А. Винокуров, А.В. Ворожцов. – М.: Физматкнига, 2008. – 288 с.

Методическая литература:

1. методические указания к лабораторным работам;
2. методические указания к самостоятельной работе.



<http://www.biblioclub.ru/> - электронная библиотека

2. <http://www.uts-edu.ru/> - «Электронные курсы»

Программное обеспечение:

Android Studio and SDK tools (бесплатный), Java IDE NetBeans (бесплатный)

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022