

М. А. Федотенко

Разработка МОБИЛЬНЫХ приложений

Первые
шаги

Под редакцией
В. В. Тарапаты



Москва
Лаборатория знаний

Оглавление

От автора	6
Благодарности	7
Введение	8
Загрузка и установка Android Studio	9
Разработка Android-приложений	13
Глава 1. «Hello, World!», или Первое приложение	14
1.1. Запуск Android Studio и создание проекта	14
1.2. Знакомство с интерфейсом Android Studio	20
1.2.1. Структура проекта Android Studio	22
1.3. Работа в режиме дизайна	27
1.3.1. TextView — текстовые элементы	27
1.3.2. Resources — библиотеки ресурсов	30
1.3.3. ID — уникальный идентификатор	32
1.3.4. ImageButton — изображение-кнопка	34
1.4. Работа в режиме кода	40
1.4.1. AndroidManifest — файл манифеста	47
1.5. Сборка проекта	49
1.6. Тестирование приложения	50
1.6.1. Файл APK	50
1.6.2. Эмулятор	53
1.6.3. USB-отладка	59
Итоги главы 1	61
Глава 2. Основы проектирования интерфейса	62
2.1. Макеты	62
2.1.1. FrameLayout — расположение элементов друг над другом	65
2.1.2. LinearLayout (vertical) — линейное расположение элементов по вертикали	70
2.1.3. LinearLayout (horizontal) — линейное расположение элементов по горизонтали	73
2.1.4. GridLayout — сеточное расположение элементов	77
2.1.5. Строковые ресурсы	81
2.2. Ориентация экрана	83
2.3. Разработка приложений для планшетов	87
2.4. Приложение «Калькулятор»	90
Итоги главы 2	102
Глава 3. Способы оповещения пользователей	104
3.1. Всплывающие сообщения	104
3.2. Диалоговые окна	109
3.2.1. Диалоговые окна с множественным выбором ..	113

3.3. Уведомления	114
3.3.1. Удаление уведомлений	118
3.3.2. Большая иконка	119
3.3.3. Приоритет уведомлений	119
3.3.4. Звуковое и световое оповещение	120
3.3.5. Уведомление, отображающее ход выполнения	121
3.3.6. Уведомления на экране блокировки	121
3.4. Звуковые эффекты	122
3.5. Приложение «Маленький принц»	126
Итоги главы 3	139
Глава 4. Дизайн и юзабилити мобильных приложений	140
4.1. Дизайн и юзабилити	140
4.2. Логотип приложения	142
4.2.1. Назначение и роль логотипа	142
4.2.2. Виды логотипов	143
4.2.3. Создание логотипа	144
4.2.4. Установка логотипа приложения	151
4.3. Загрузочный экран приложения	152
4.3.1. Анимация элементов и класс AnimationUtils	157
4.3.2. ProgressBar — индикатор загрузки	159
4.4. Темы и стили	161
4.4.1. Стили	161
4.4.2. Темы	164
4.5. Меню. Виды меню	167
4.5.1. Меню-шторка	174
4.5.2. Фрагменты	178
4.5.3. Главное меню	183
4.6. Навигация. Переключение между несколькими экранами	184
Итоги главы 4	194
Глава 5. Работа с текстом, изображениями и жестами	195
5.1. Работа с текстом	195
5.1.1. Длинный текст	196
5.1.2. ScrollView — контейнер с возможностью прокрутки	198
5.1.3. ScrollingActivity — прокручиваемый экран	201
5.2. Обработка касаний и жестов	204
5.2.1. Двойное нажатие	206
5.2.2. Долгое нажатие	207
5.2.3. Отпущенное одиночное нажатие	207
5.2.4. Скроллинг и свайпинг	208
5.3. Работа с камерой	208

5.4. Приложение «Скетчбук творческой личности»	212
Итоги главы 5	225
Глава 6. Интернет и базы данных	226
6.1. Интернет	226
6.2. Базы данных	228
6.2.1. Подключение БД к проекту Android Studio	238
6.3. Инструмент Firebase	246
6.4. Приложение «Посторонним вход воспрещен»	249
Итоги главы 6	262
Глава 7. Сторонние приложения и встроенные инструменты	263
7.1. Работа со сторонними приложениями	263
7.1.1. Телефон	263
7.1.2. Браузер	264
7.1.3. Электронная почта	264
7.1.4. Магазин приложений Google Play	265
7.1.5. Фонарик	266
7.1.6. Switch — переключатель	268
7.2. Конвертация сайта в мобильное приложение	270
7.3. Чтение QR-кодов	275
7.4. Работа с картами Google	283
7.4.1. Установка маркера	287
7.4.2. Изменение типа и настроек карты	289
7.4.3. Определение текущего местоположения	290
7.5. Приложение «Вокруг света за 80 дней»	291
Итоги главы 7	298
Глава 8. Итоговый проект «Общалка». Уровень: продвинутый Android-разработчик	299
Итоги главы 8	312
Заключение	314
Приложения	315
Магазины приложений	315
Девять шагов к идеальному приложению	318
Спортивное ориентирование по проекту Android Studio	319
Исправление типичных ошибок	321
Глоссарий	328

От автора

*Посвящается
Виталию Ивановичу Кулишу*

Кажется, что сегодня буквально все можно сделать мобильным приложением. А вы как раз давно хотели стать разработчиком мобильных приложений, но не знали, с чего начать? Смотрите на многочисленные приложения и думаете, что это сложно?

Легко ли создавать приложения для Android™?¹

Если задать этот вопрос в Интернете — большинство ответов будет «сложно». Но в магазине приложений Google Play миллионы приложений. Если бы это было очень сложно, их было бы тысячи; невероятно сложно — сотни, причем их разработка была бы «по плечу» только крупным компаниям. Но их миллионы! А значит, вы вполне можете стать одним из сотен тысяч разработчиков по всему миру и создать новый продукт, который будут использовать и любить миллионы пользователей.

Не стоит оглядываться на пессимистов!

В чем отличие этой книги от остальных книг по Android-разработке? Она проста и понятна для начинающих, яркая и увлекательная.

Следуя подробным пошаговым инструкциям, вы уже на первом уроке создадите приложение, которое можно запустить и которым можно поделиться с друзьями и не только.

Думаете, нужно заранее досконально изучить несколько языков программирования? Нет, для начала будет достаточно понимания их основ, которое вы получите в процессе чтения данной книги². При этом вы сможете называться настоящими Android-разработчиками, в отличие от тех, кто пользуется различными конструкторами приложений.

Вам не придется на протяжении всей книги идти к одному-единственному результату. Результатом изучения нескольких уроков будет готовый продукт (калькулятор, скетчбук или викторина), который можно совершенствовать и далее³. А в конце вы создадите собственный мессенджер и, при желании, сможете развить его до уровня WhatsApp или Viber.

Этому мало где учат. Далеко не во всех, даже ведущих, университетах есть курсы Android-разработки, а значит, с уверенно-

¹ Android является товарным знаком Google Inc.

² Основы программирования также можно изучить с помощью книг серии «Школа юного программиста» издательства «Лаборатория знаний».

³ Примеры приложений из этой книги можно найти на ее странице на сайте издательства «Лаборатория знаний».

стью можно сказать, что большинство разработчиков — самоучки. Что же мешает вам?

Задача этой книги — дать вам основные инструменты, с помощью которых вы сами сможете создать собственный ВКонтакте, или... Кто знает?

Успехов вам, будущие Android-разработчики!

Благодарности

Искренняя признательность профессору МПГУ Надежде Николаевне Самылкиной, благодаря которой написание этой книги стало возможным.

Благодарю моего научного руководителя, Марину Леонидовну Соболеву, за ее поддержку и бесценные советы.

Спасибо моим любимым студентам за проявленный интерес и помощь в испытании проектов этой книги на практике.

Отдельная благодарность моим коллегам, Виктору Викторовичу Тарапате и Алене Антоновне Салаховой, а также замечательному коллективу издательства «Лаборатория знаний» — всем, кто работал с рукописью и способствовал ее улучшению.

И самая главная благодарность всем, кто прочитает эту книгу.

Сердечно благодарю мою дорогую семью за всестороннюю помощь и веру в мои силы.

Введение

Эта книга посвящена основам разработки мобильных приложений под Android в интегрированной среде разработки¹ **Android Studio**.

Android Studio — официальная среда разработки мобильных приложений для устройств с операционной системой Android от компании Google. Это доступный и универсальный инструмент. Он обладает широкими возможностями и достаточно прост в освоении.

Работа в Android Studio с помощью этой книги станет намного проще и гораздо интереснее. Например, в разработке мобильных приложений под Android задействованы сразу четыре языка программирования и разметки (Java, XML, SQL и Groovy), однако мы не будем тратить время на изучение основ каждого языка в отдельности — все произойдет в процессе обучения разработке мобильных приложений, а на выходе получится довольно внушительный набор IT-навыков.

Глава 1 этой книги мало похожа на стандартную главу учебника, в ней нет блока вводной теории, вопросов, дополнительных заданий. Она содержит только один урок, скорее напоминающий руководство пользователя IDE Android Studio. Задача главы 1 — провести краткий экскурс в мир разработки мобильных приложений, показать весь процесс от запуска Android Studio до запуска приложения на смартфоне.

Все последующие главы содержат по несколько уроков. В каждом уроке создается мини-приложение с набором основных функций современных мобильных приложений. После прохождения нескольких уроков изученные функции объединяются в одно полноценное приложение. Завершает книгу итоговый проект, в котором мы с помощью всех полученных навыков создадим собственный мессенджер.

Развернутое оглавление книги поможет легко ориентироваться в расположении описания изученных функций. Если в процессе реализации десятого приложения мы забудем, как создавать APK, достаточно заглянуть в оглавление и глоссарий. Его можно найти в Приложениях, так же как Руководство по исправлению основных ошибок и подробное описание структуры проекта Android Studio.

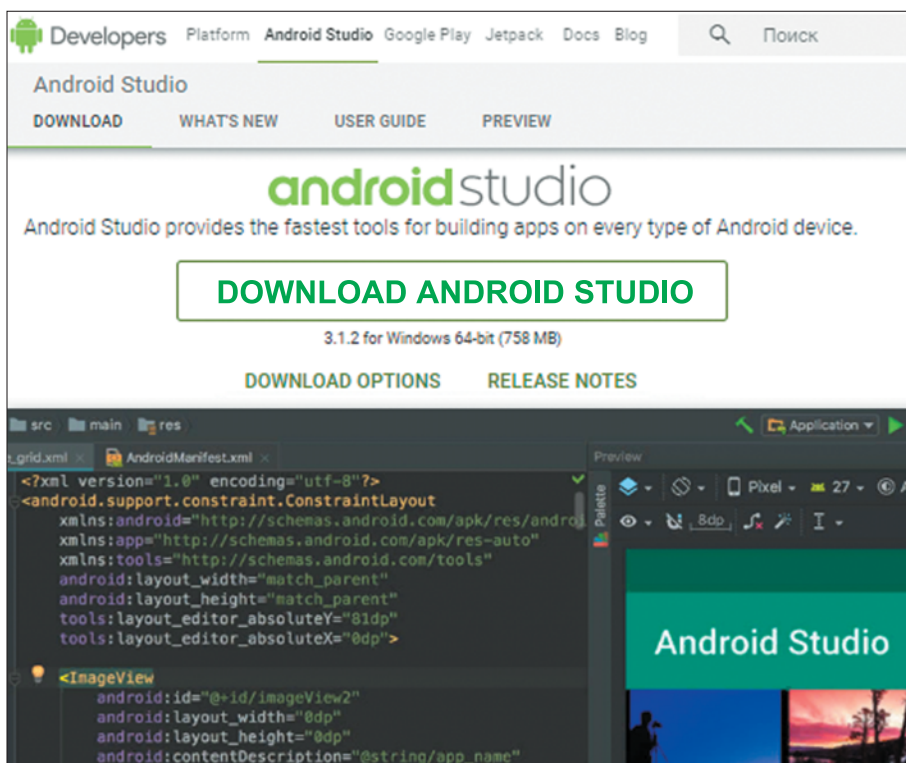
Готовы стать настоящим Android-разработчиком?
Давайте пройдем этот путь вместе!

¹ Интегрированная среда разработки (Integrated development environment, IDE) — среда разработки, которая уже включает в себя все необходимые для разработки средства, такие как компилятор, текстовый редактор, средство автоматизации сборки и отладчик.

Загрузка и установка Android Studio

Интегрированная среда разработки **Android Studio** является свободно распространяемой. Текущая версия доступна для загрузки на официальном сайте разработчика <https://developer.android.com/studio/index.html>

На открывшейся странице сайта есть кнопка **Download Android Studio** (*скачать Android Studio*). Она запускает скачивание текущей версии для 64-разрядной Windows:



Если на компьютере установлена другая операционная система, нужно пролистать страницу вниз и найти раздел **System Requirements** (*системные требования*). В нем содержатся заявленные разработчиком системные требования:

Для **Windows**:

- Microsoft® Windows® 7/8/10 (32- или 64-разрядная версия);
- минимум 3 Гб ОЗУ (рекомендуется 8 Гб) плюс 1 Гб для эмулятора Android (виртуального устройства для тестирования приложений);
- минимум 2 Гб доступного места на диске (рекомендуется 4 Гб);
- минимальное разрешение экрана 1280×800.

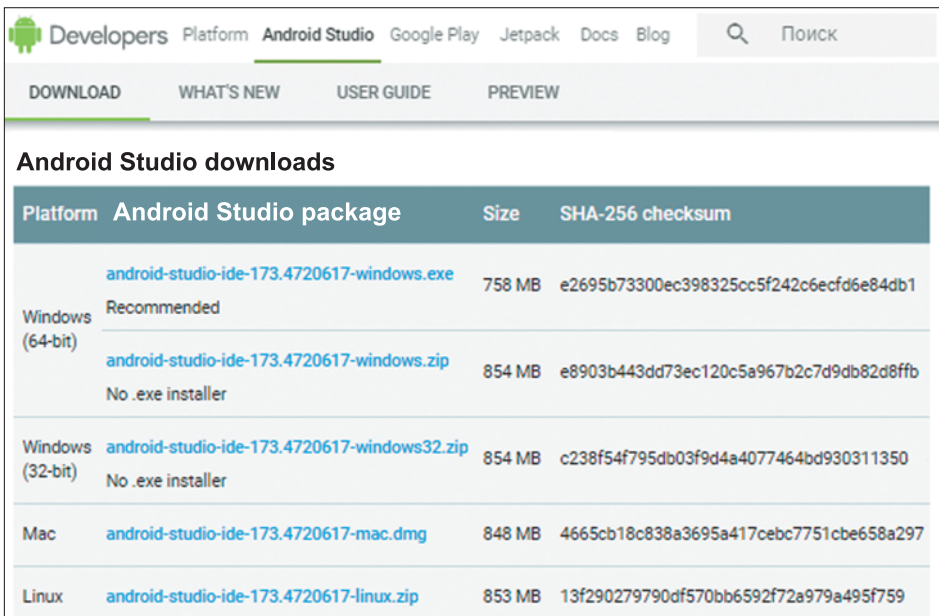
Для Mac OS:

- Mac® OS X® 10.10 (Yosemite) или выше;
- минимум 3 Гб ОЗУ (рекомендуется 8 Гб) плюс 1 Гб для эмулятора Android;
- минимум 2 Гб доступного места на диске (рекомендуется 4 Гб);
- минимальное разрешение экрана 1280×800.

Для Linux:

- рабочий стол GNOME или KDE;
- библиотека GNU C (glibc) 2.19 или более поздней версии;
- минимум 3 Гб ОЗУ (рекомендуется 8 Гб) плюс 1 Гб для эмулятора Android;
- минимум 2 Гб доступного места на диске (рекомендуется 4 Гб);
- минимальное разрешение экрана 1280×800.

Перед разделом системных требований идет раздел **Android Studio downloads** (*загрузки*), в котором представлены ссылки для скачивания **Android Studio** для разных платформ:



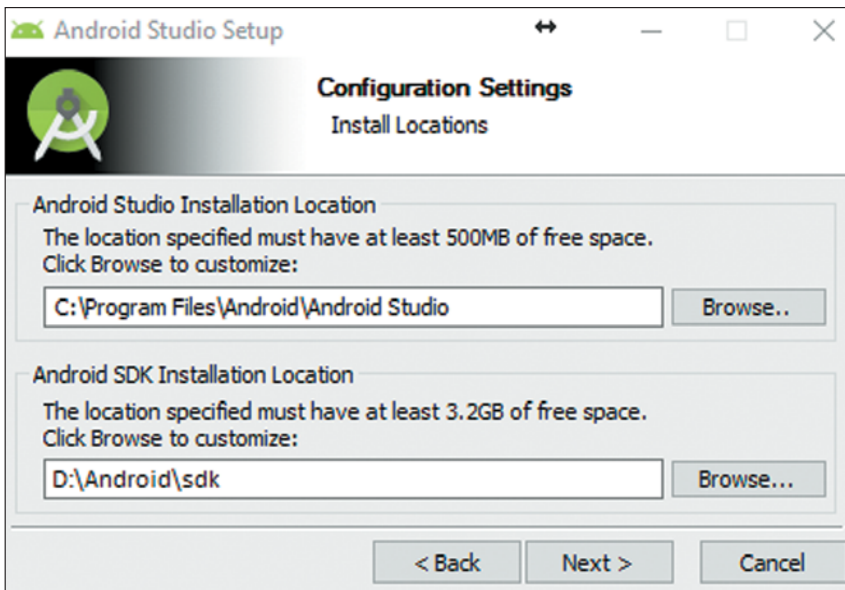
Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	android-studio-ide-173.4720617-windows.exe Recommended	758 MB	e2695b73300ec398325cc5f242c6ecfd6e84db1
	android-studio-ide-173.4720617-windows.zip No .exe installer	854 MB	e8903b443dd73ec120c5a967b2c7d9db82d8ffb
Windows (32-bit)	android-studio-ide-173.4720617-windows32.zip No .exe installer	854 MB	c238f54f795db03f9d4a4077464bd930311350
Mac	android-studio-ide-173.4720617-mac.dmg	848 MB	4665cb18c838a3695a417cebc7751cbe658a297
Linux	android-studio-ide-173.4720617-linux.zip	853 MB	13f290279790df570bb6592f72a979a495f759

Выбираем свою операционную систему и скачиваем соответствующий установочный пакет из поля **Android Studio package** (*пакет Android Studio*).

В загруженном установочном файле уже имеются все необходимые для разработки инструменты: сама **IDE Android Studio** и **Android SDK**¹. Таким образом, дополнительно ничего скачивать и устанавливать не требуется.

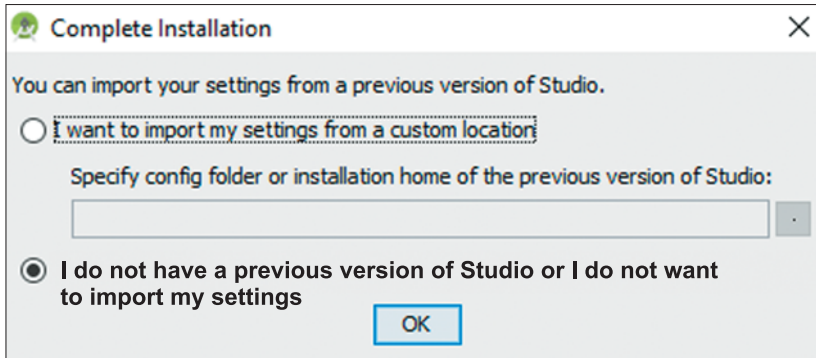
Процесс установки **Android Studio** стандартный, все опции оставляем и принимаем без изменений. Обратит внимание стоит на два момента:

1. При выборе путей установки **Android Studio** и **Android SDK** следует учесть, что:
 - для установки SDK нужно минимум 3.2 Гб свободной памяти на диске;
 - выбранные пути не должны содержать букв кириллицы — это важно.



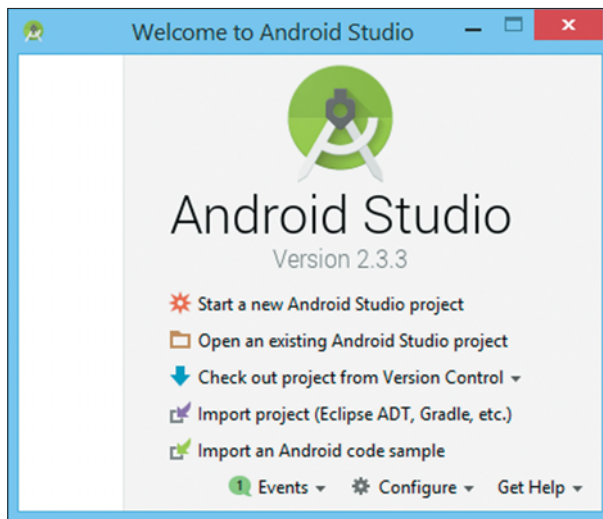
2. При первом запуске **Android Studio** задаст вопрос об импорте конфигурации. Выбираем второй пункт — **I do not have a previous version of Studio or I do not want to import my settings** (*у меня нет предыдущей версии Android Studio или я не хочу импортировать текущие настройки*):

¹ Android SDK (Software Development Kit) — набор инструментов для разработки программного обеспечения для операционной системы Android. Включает в себя эмуляторы устройств, документацию и все необходимые для разработки пакеты.



После этого **Android Studio** начнет самостоятельно загружать **Android SDK** из сети Интернет. Если загрузка SDK с первого раза не получится, **Android Studio** предложит повторить попытку — обязательно нажимаем **Retry** (*попробовать снова*).

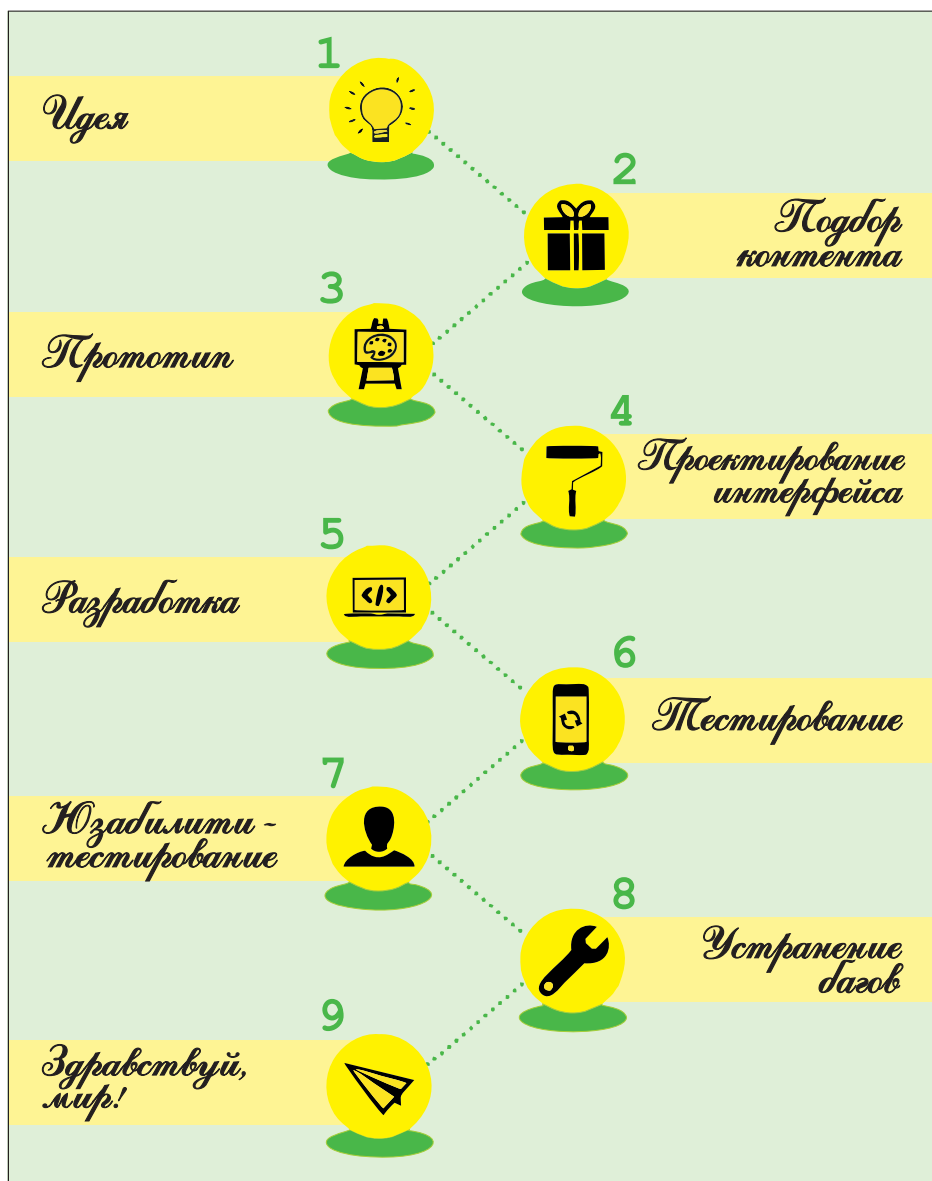
Когда установка будет полностью завершена, появится приветственное окно **Android Studio**:



На панели слева будут отображаться уже созданные проекты (при первом запуске она, естественно, будет пуста), в основной рабочей области — возможности для работы с проектами (создание нового проекта, открытие уже существующего, загрузка и импорт проектов), а внизу — **Events** (*события*, например оповещения об обновлениях), **Configure** (переход к настройкам **Android Studio**) и **Get Help** (меню помощи).

Это означает, что теперь у нас есть все инструменты, необходимые для того, чтобы стать **Android-разработчиками**. Вперед!

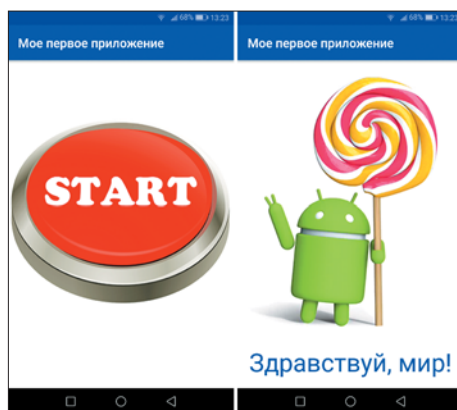
РАЗРАБОТКА ANDROID-ПРИЛОЖЕНИЙ



Глава 1. «Hello, World!», или Первое приложение

Все программисты начинают освоение нового языка программирования с первой программы — «Hello, World!» (в переводе с англ. — «Здравствуй, мир!»). И это не только базовый прием, показывающий программисту особенности нового языка и структуру программы. Это традиция, впервые она была введена в книге «Язык программирования Си» Брайана Кернигана и Денниса Ритчи в 1978 году.

И мы последуем этой традиции при знакомстве с **Android Studio**. Ведь «Hello, World!» — это не просто абстрактное приветствие, на самом деле это оптимистичное и довольно амбициозное заявление: «Здравствуй, мир! Встречай нового Android-разработчика!»¹.



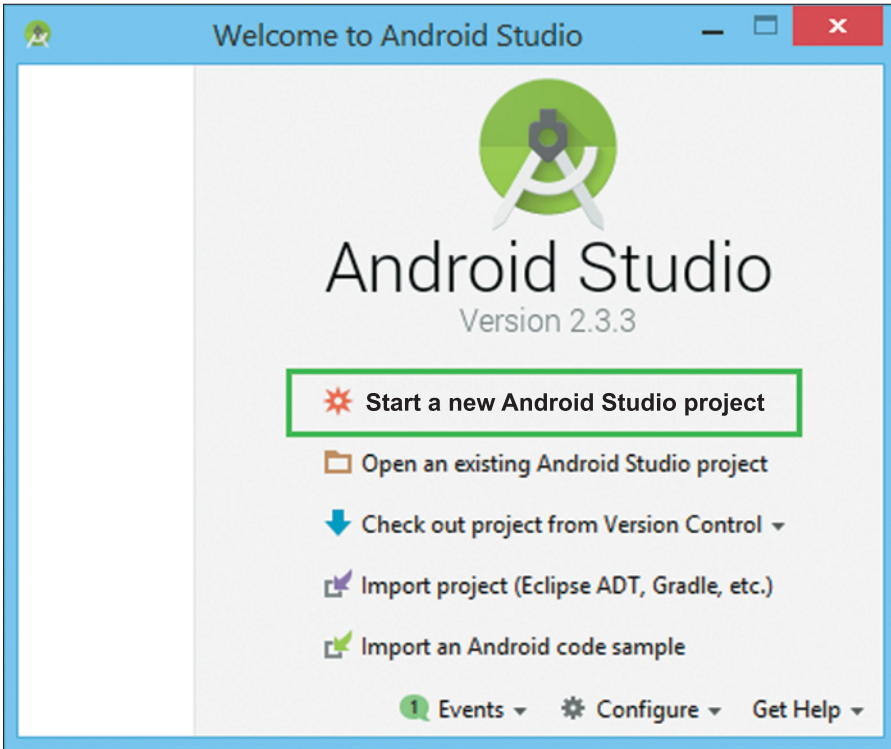
Концепция приложения: при запуске приложения на экране отображается кнопка старта, при нажатии на которую появляется приветственный текст и картинка.

1.1. Запуск Android Studio и создание проекта

Для создания нашего первого (и любого последующего) приложения необходимо создать новый Android Studio проект.

¹ Изображение робота с леденцом — логотип ОС **Android 5.0 Lollipop**.

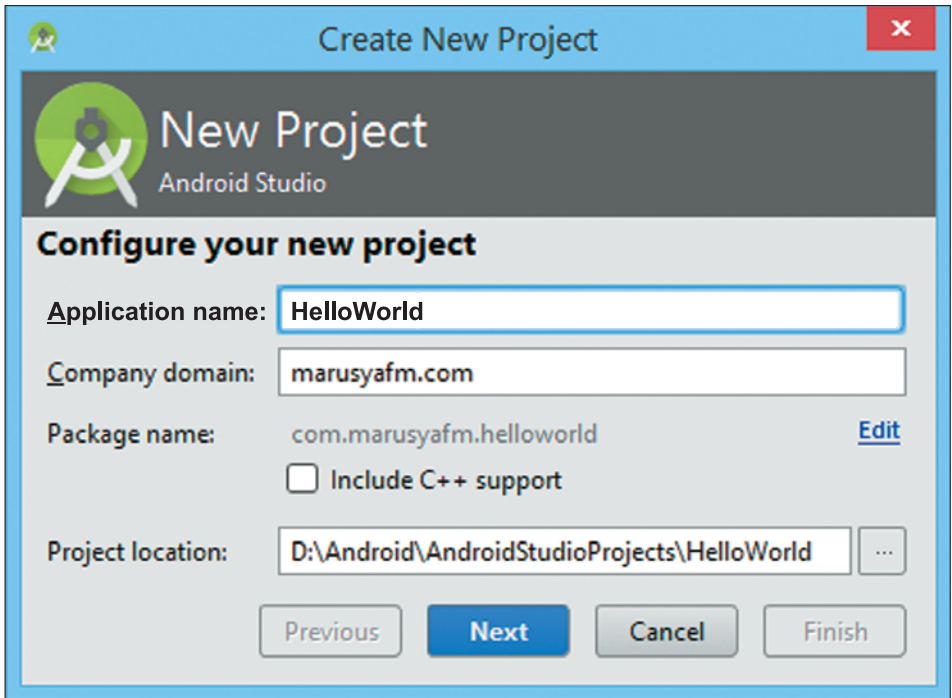
Чтобы создать проект, в приветственном окне Android Studio выбираем первый пункт — **Start a new Android Studio project** (*начать новый проект Android Studio*).



В открывшемся окне задаем **Application name** (*имя приложения*) — HelloWorld. При первом запуске приложения это имя будет отображаться в «шапке» самого приложения. В дальнейшем его легко изменить, поэтому то, что мы введем сейчас, останется только в качестве названия проекта Android Studio и не будет отображаться ни в магазине приложений Google Play, ни в меню смартфона, ни где-либо еще.

Важно!

- Имя приложения должно содержать только буквы латиницы, цифры и знаки препинания.
- Имя приложения должно иметь смысл и напоминать о содержании приложения. Для первого приложения это кажется несущественным, но в дальнейшем будет трудно ориентироваться в стандартных MyApplication1, MyApplication123 и MyApplication5678 и так далее.



В **Company domain** (*домен компании*) по умолчанию записывается <имя_компьютера>.com. Профессиональные разработчики (особенно крупные компании) указывают в этом поле свой сайт. Если собственного сайта нет — оставляем без изменений.

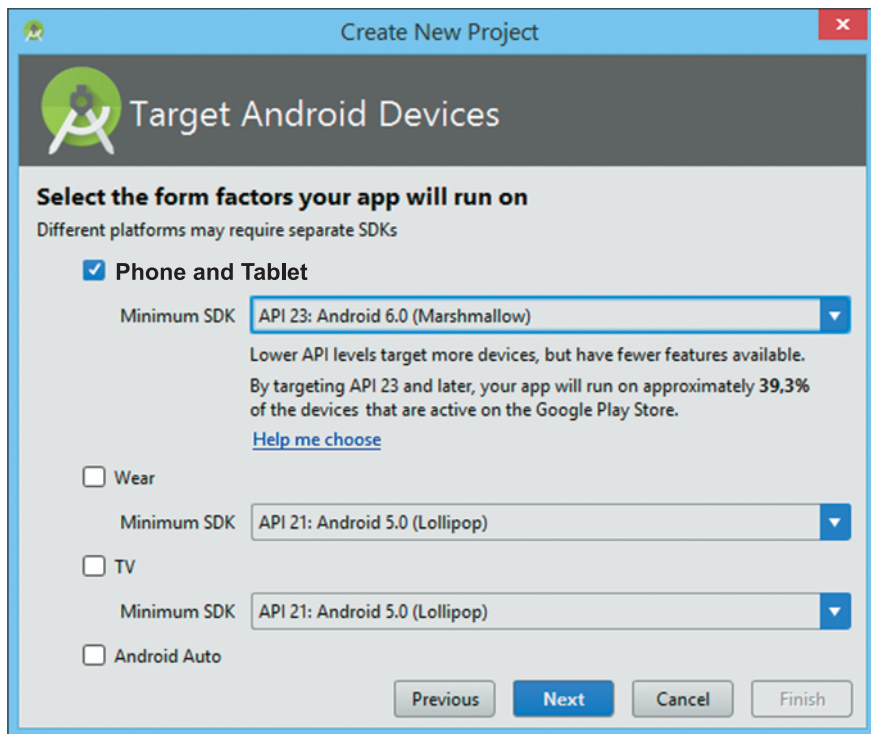
Также при желании можно изменить **Project location** (*расположение проекта — путь к нему*).

Важно!

Путь не должен содержать букв кириллицы и пробелов.

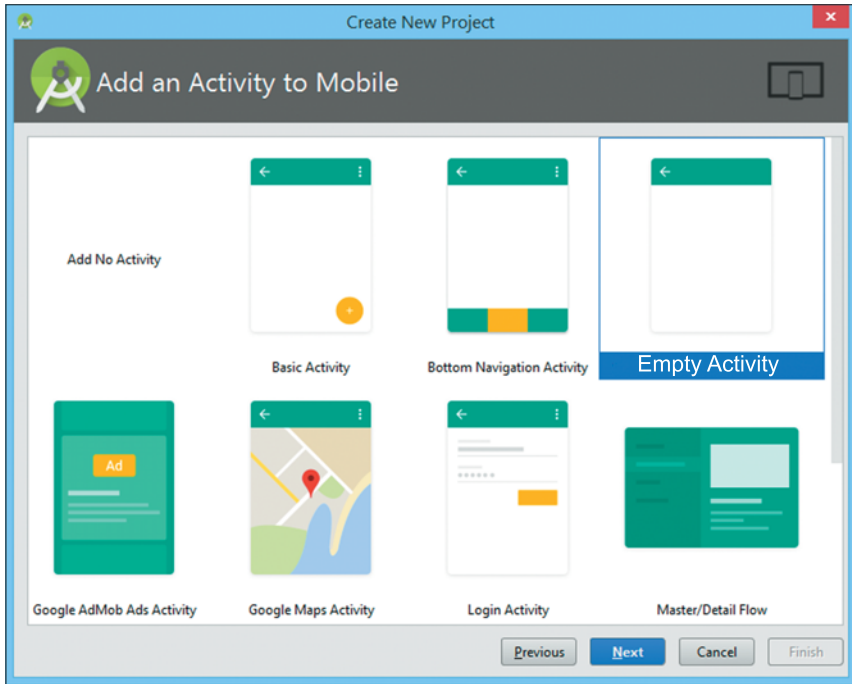
Задаем новый путь (или же оставляем путь, созданный **Android Studio** по умолчанию) и переходим к следующему шагу — нажимаем кнопку **Next**.

В появившемся окне нужно выбрать устройства, для которых будет написано приложение. В среде **Android Studio** можно создавать приложения для самых разных устройств: смартфонов и планшетов, умных часов, телевизоров, автомобилей. Все они выглядят, устроены и функционируют по-разному, а значит, и проекты, и программный код будут иметь свои особенности. На наших уроках речь пойдет о приложениях для смартфонов и планшетов, поэтому выберем **Phone and Tablet** (*смартфоны и планшеты*).

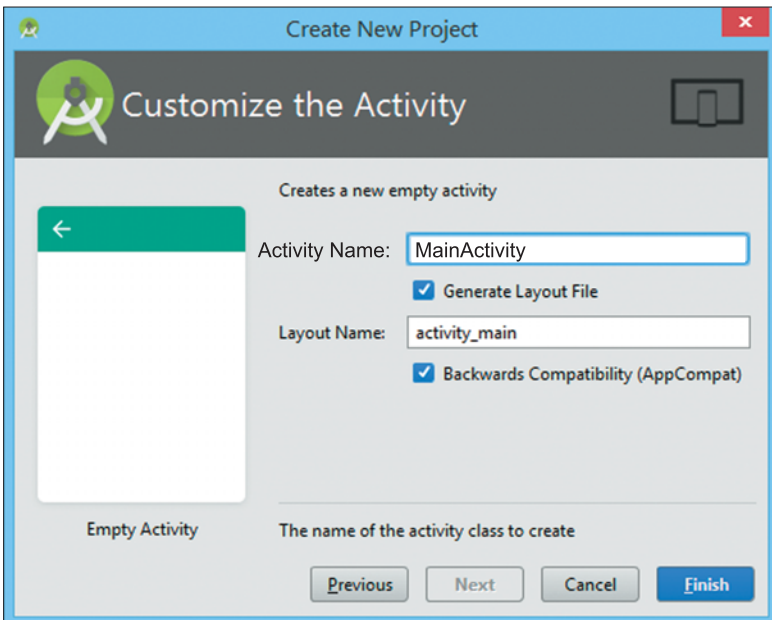


Затем выбираем минимальную версию операционной системы Android. При выборе системы из списка отображается информация о том, что выбор более старой версии операционной системы позволит приложению работать на большей доле смартфонов, чем при выборе недавно анонсированной версии. Однако при этом используется меньше новых функций. Этот параметр впоследствии можно будет изменить в настройках проекта. Выберем, например, **Android 6.0 (Marshmallow)**. **Android Studio** оповещает нас, что приложение, написанное для этой версии, будет работать приблизительно на 39,3% устройств. Эта цифра гораздо ниже реальной, но при разработке приложений для широкой аудитории лучше выбрать версию системы, охватывающую 100% устройств. Переходим к следующему шагу.

Далее нужно выбрать тип Activity. **Activity** («*активность*», *активная область экрана приложения*) — это каждый отдельно взятый экран нашего приложения. Позже мы сможем добавить в приложение сколько угодно Activity, и все они могут быть разного типа. С этой целью **Android Studio** предлагает ряд готовых шаблонов: стандартный экран, пустой экран, широкоформатный экран, экран настроек, экран для работы с картами Google, форма авторизации и другие:



На этапе создания приложения нужно выбрать тип только для одной Activity — экрана, который будет начальным. Выбираем **Empty Activity** (*пустой экран*) и переходим к последнему шагу.



Здесь мы оставляем все параметры без изменений, в том числе **Activity Name** (*имя Activity*) по умолчанию **MainActivity** (*главная Activity*). Завершаем настройку проекта нажатием кнопки **Finish**.

Теперь **Android Studio** понадобится некоторое время на запуск и загрузку всех необходимых для разработки инструментов, и потом еще на создание и сборку проекта. Этот процесс довольно долгий (может занимать до 15 мин) и требует активного подключения к сети Интернет.

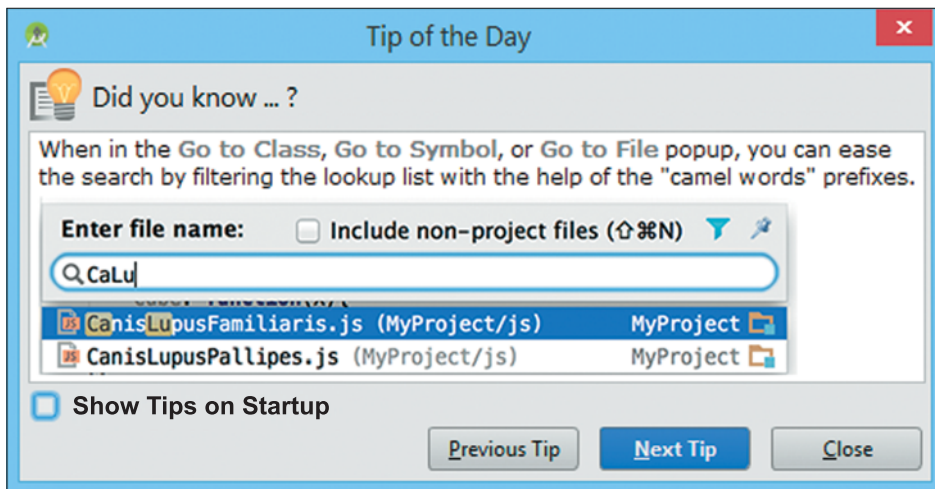
Важно!

На этом этапе главное правило — **не начинать работу до того, как Android Studio сообщит о своей готовности**. Когда процесс сборки проекта и загрузки всех модулей завершится, значок **Android Studio** на панели задач будет подсвечен и на нем появится зеленая галочка. Это значит, что мы можем начинать.



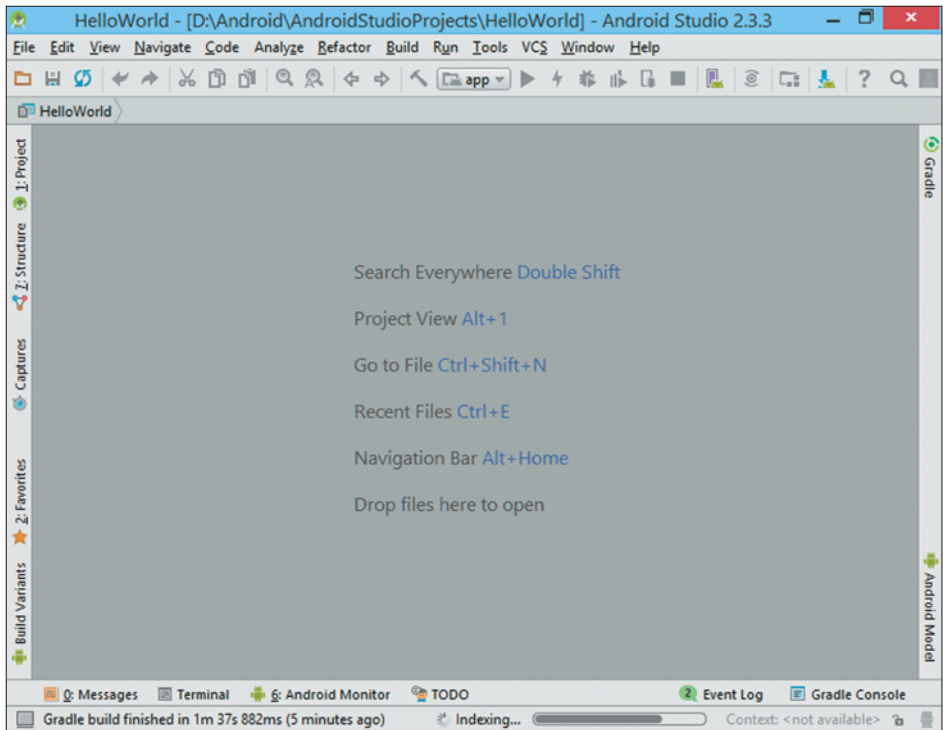
А пока процесс запуска в самом разгаре (он происходит в несколько этапов), поговорим об интерфейсе, возможностях и основных инструментах **Android Studio**.

На первом этапе запуска **Android Studio** загружает непосредственно среду разработки. В начале, как и в любой другой среде разработки, появляется всплывающее окно с подсказками. Снимем галочку (чекбокс) **Show Tips on Startup** (*показывать подсказки при загрузке*), чтобы не видеть это окно при каждом запуске **Android Studio**, и закроем окно.

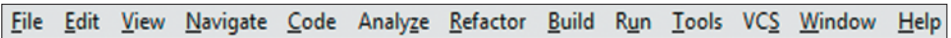


1.2. Знакомство с интерфейсом Android Studio

В заголовке окна мы видим **название нашего проекта** (HelloWorld), его **расположение** (путь к проекту) и **текущую версию Android Studio** (2.3.3).



Ниже располагается **Главное меню** со стандартным для любой среды программирования набором пунктов:



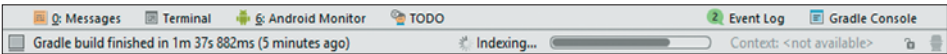
Панель быстрого доступа содержит кнопки для перехода к наиболее востребованным пунктам Главного меню, таким как: создать, сохранить или обновить проект; отменить или повторить действие; вырезать, копировать, вставить; собрать проект, скомпилировать проект, запустить проект (все эти понятия мы рассмотрим в следующих уроках):



Панель навигации отображает путь к текущему открытому файлу. С ее помощью можно легко перемещаться между файлами и папками, образующими довольно сложную структуру проекта **Android Studio**:

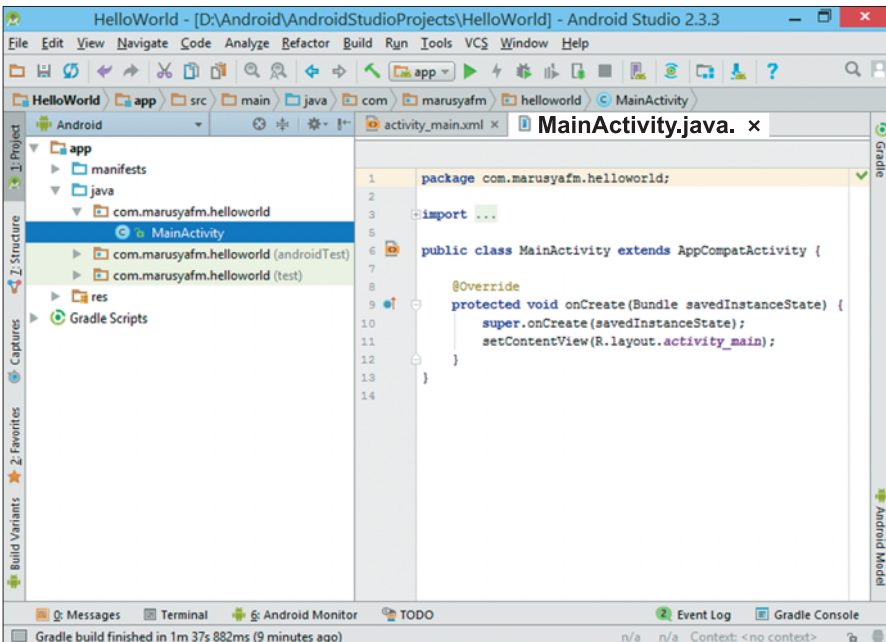


В самом низу окна расположена **Панель выполнения и отладки**. С помощью элементов, расположенных на этой панели, можно отслеживать и контролировать все процессы, выполняемые **Android Studio**:



Рабочая область (темно-серая область по центру) перестраивается и изменяет свою структуру в зависимости от типа открытого файла проекта. Появляются дополнительные панели свойств, палитры компонентов, области предварительного просмотра и т. д.

За это время подгрузился и открылся файл **MainActivity.java**.



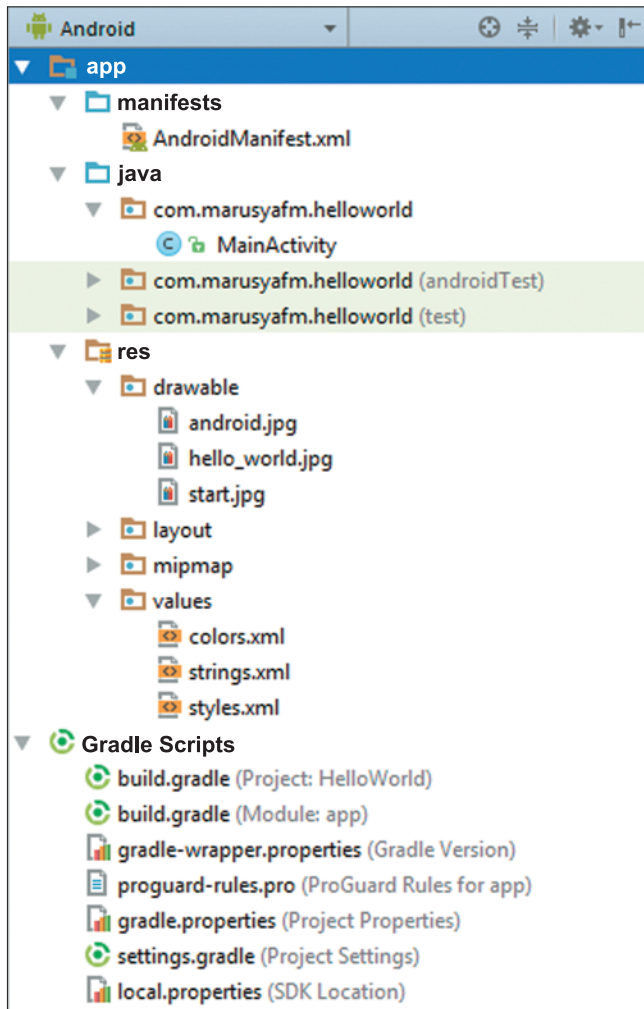
Файлы Java содержат в себе только код на языке программирования Java™ — для них не отображается никаких дополнительных панелей и областей.

Пояснение

Программный код для наших приложений мы будем писать на языке Java. Для того чтобы начать разрабатывать мобильные приложения, совершенно необязательно досконально изучать его заранее. Основ программирования на Java, которые мы получим на этом и последующих уроках, будет вполне достаточно.

Теперь мы можем познакомиться со структурой проекта Android Studio.

1.2.1. Структура проекта Android Studio



Изначально структура проекта Android Studio¹ включает в себя следующие папки:

1. **app** — корневая папка, в которой хранятся все файлы нашего приложения. Имеет следующую подструктуру:

- **manifests** — в этой папке хранятся **манифесты** — файлы, в которых на языке разметки XML записана основная информация о конфигурации приложения: отображаемое название приложения, структура приложения, иконка и ее миниатюра, полномочия и системные требования. Эта информация нужна операционной системе Android (для установки приложения и его отображения в памяти и на экране смартфона, для проверки версии обновления), а также магазинам приложений, таким как Google Play;
- **java** — место хранения исходного программного кода нашего приложения — все Java-классы, необходимые для работы и тестирования;
- **res** — папка ресурсов. В ней хранятся все картинки (папка **drawable**), файлы xml-разметки (папка **layout**), служебные иконки (папка **mipmap**) и описания всех заданных в проекте значений различных переменных (папка **values**).

2. **Gradle Scripts** — корневая папка для скриптов системы автоматической сборки **Gradle**. Скрипты написаны на языке Groovy и хранятся в папке **Gradle Scripts**. **Android Studio** производит сборку приложения с помощью этой системы в фоновом режиме, выполняя Gradle-скрипты без необходимости нашего вмешательства как разработчиков.

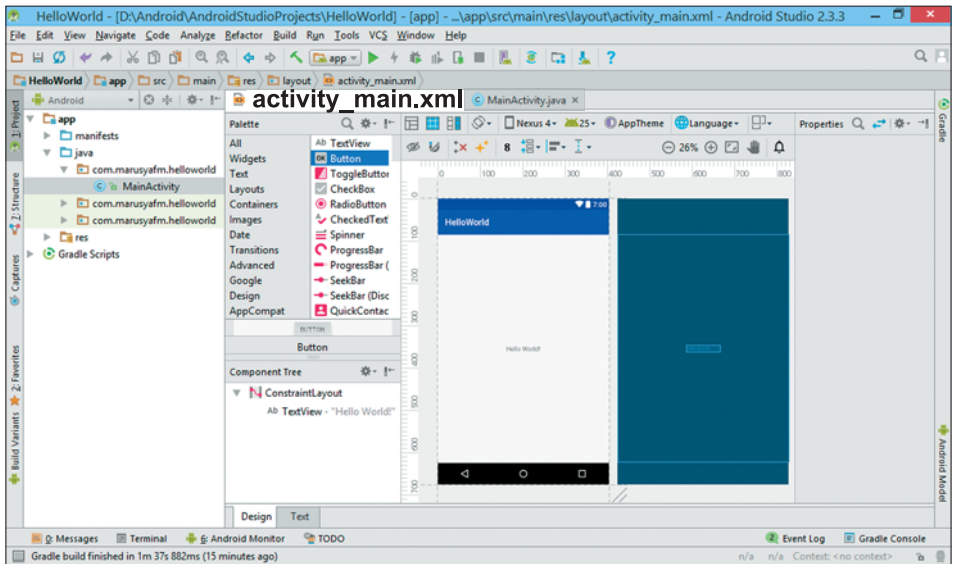
Пояснение

Сборка приложения — процесс получения готового продукта (приложения) из исходного кода, который, как мы видим, написан на разных языках программирования и разметки и «разбросан» по разным файлам, образующим сложную структуру проекта Android Studio.

Процесс сборки включает в себя **компиляцию** (преобразование кода, написанного разработчиком, в машинный код — набор команд, понятных компьютеру) и **компоновку** (преобразование всех файлов проекта в один исполняемый файл, который затем можно будет запустить на смартфоне).

¹ Подробное описание структуры проекта Android Studio см. в Приложении в конце книги.

Тем временем процесс запуска **Android Studio** завершен и наш проект принял следующий вид — открылся файл xml-разметки **activity_main.xml**:



А значит, все готово, и мы можем приступать к разработке. **Файлы xml** хранят в себе описания всех элементов приложения (кнопок, надписей, картинок) и их настроек на языке разметки **XML**.

Пояснение

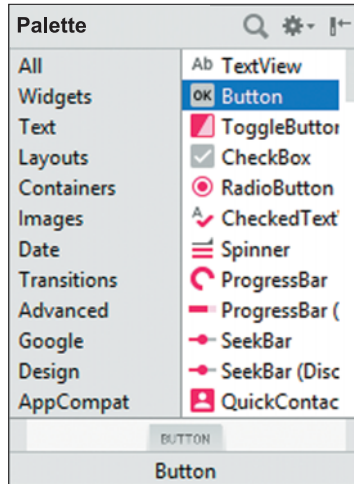
XML (eXtensible Markup Language) — расширяемый язык разметки, созданный для описания данных. В **Android Studio** на языке XML описываются элементы интерфейса (дизайн) приложения, а также некоторые ресурсы (цвета, строки, стили).

Файл XML представляет собой описание элементов интерфейса и их атрибутов (свойств), заключенных в парные теги:

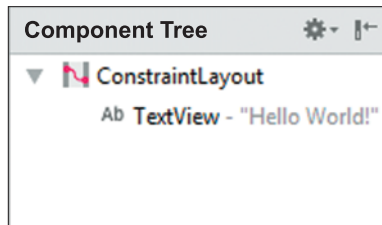
```
<открывающий тег>...</закрывающий тег>
```

Рабочая область для xml-файлов в **Android Studio** содержит шесть элементов:

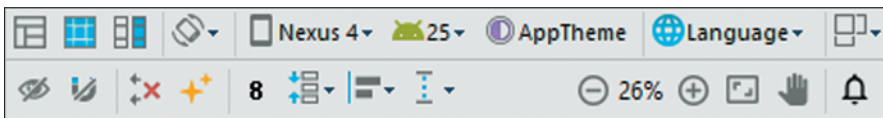
1. **Palette** (*палитра элементов*). В ее левой части перечислены названия типов элементов (все элементы, виджеты, текстовые элементы, шаблоны и изображения), а в правой — все элементы выбранного типа. Для того чтобы добавить элемент на экран, достаточно перетащить его из палитры элементов:



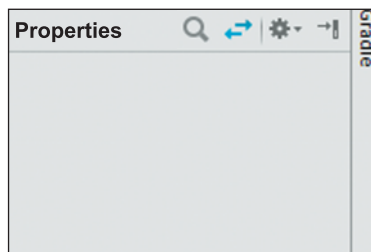
2. **Component Tree** (*дерево компонентов*). В нем отображаются названия и ID (*идентификаторы*) всех элементов, расположенных на экране.



3. **Панель быстрого доступа**. На ней собраны кнопки для быстрого перехода к настройкам отображения Activity:



4. **Панель Properties (Attributes, панель свойств)**. Сейчас она пуста, но при выборе элемента на ней будут отображаться все свойства этого элемента.





Автор книги, Федотенко Мария Александровна, профессиональный Android-разработчик, имеет 5-летний опыт работы в ведущих IT-компаниях России и Германии.

В настоящее время преподает программирование и разработку мобильных приложений в МПГУ.

Активная участница движения «Women in Tech».

Программирование – это грамотность XXI века!

Книги новой серии «Школа юного программиста» издательства «Лаборатория знаний» построены на методике пошагового обучения программированию. Следуя этой методике, любой желающий, от школьника до студента вуза, сможет научиться писать программы, разрабатывать мобильные приложения и компьютерные игры и даже освоить технологии машинного обучения и нейросетей.

В серию войдут следующие учебные пособия:

- «Учимся вместе со Scratch: программирование, игры, робототехника» (5–6 классы)
- «Scratch 2.0: от новичка к продвинутому пользователю. Пособие для подготовки к Scratch-Олимпиаде» (1–11 классы)
- «Творческие задания в среде Scratch. Рабочая тетрадь для 5–6 классов»
- «Scratch 2.0: творческие работы на вырост. Рабочая тетрадь для 7–8 классов»
- «Создаем игры с Kodu Game Lab» (4–5 классы)
- «Python для начинающих – от основ до ООП и приложений» (7 класс)
- «Олимпиадное программирование на Python» (7–8 классы)
- «С# – новый учебный курс программирования от основ до продвинутого уровня» (8–9 классы)
- «Разработка мобильных приложений. Первые шаги» (8–9 классы)
- «Web-разработка: создай свой идеальный сайт. Обучаемся тонкостям HTML, HTML5, CSS3, SQL, PHP, JavaScript» (8–10 классы)
- «Основы искусственного интеллекта и нейросетей» (10–11 классы, студенты) и другие.

