МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по выполнению лабораторных работ

по дисциплине «Программирование мобильных устройств»

для студентов направлений 09.03.03 «Прикладная информатика» и 09.03.02 «Информационные системы и технологии».

Ставрополь

2021**ВВЕДЕНИЕ**

Целью дисциплины «Практикум по программированию мобильных устройств» является формирование набора профессиональных компетенций будущих бакалавров по направлению подготовки 09.03.03 «Прикладная информатика».

Задачами дисциплины являются:

* ознакомление студентов с видами пакетов прикладных программ и классами программного обеспечения;
* обучение студентов постановке задач, корректному и эффективному использованию инструментальных средств;
* приобретение студентами навыков самостоятельного и последовательного применения пакетов прикладных программ в профессиональной деятельности;
* формирование логического мышления.

# **Лабораторная работа №1.** **Знакомство с Android Studio**

**Цель работы:** Изучение интерфейса Android studio и создание первого простого приложения.

**Формируемые компетенции:** ПК-7, ПК-8

**Теоретическая часть**

Необходимо вспомнить основные принципы ООП.

Существует огромное количество компонентов, из которых можно собрать приложение. В Android studio они представлены, как виджеты. Основные из них:

* TextView – поле, в котором будет отображаться текст;
* Button – кнопка;
* ProgressBar – индикатор прогресса;
* EditText – поле ввода данных;
* ChekBox – особый тип кнопки, который может быть в одном из двух состояний (checked или unchecked);
* RadioButton – подобный CheckBox тип кнопки, с одним исключением, что используется в RadioGroup, где Checked можно присвоить лишь одному экземпляру RadioButton;
* Toast – небольшое всплывающее сообщение;
* ListView – список строк.

**Ход лабораторной работы:**

Процесс создания приложения состоит из следующих этапов.

1. Скачиваем и устанавливаем Android studio https://developer.android.com/studio

Процесс установки включает в себя многократное нажатие «Далее». Следует учесть, что имя пользователя компьютера должно быть только на английском языке. Дело в том, что среда Android Studio не поддерживает формат UTF-8, из-за чего возникнут проблемы во время установки.

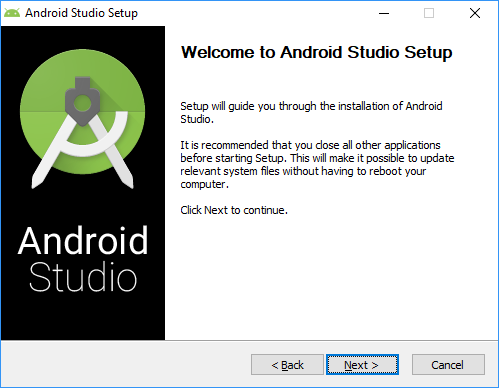


Рисунок 1 - Главное окно установки

Установка является самой сложной частью программирования на Android. Установка разбивается на несколько этапов:

* установка основного компонента;
* установка SDK-менеджера;
* установка компонентов для сборки;
* установка эмулятора.

1. Запускаем Android studio и создаём приложение.

Android Studio обладает некоторыми предустановленными шаблонами, чтобы проектировать приложения. Сейчас мы рассмотрим самое простое приложение, поэтому, выберите Empty Activity.

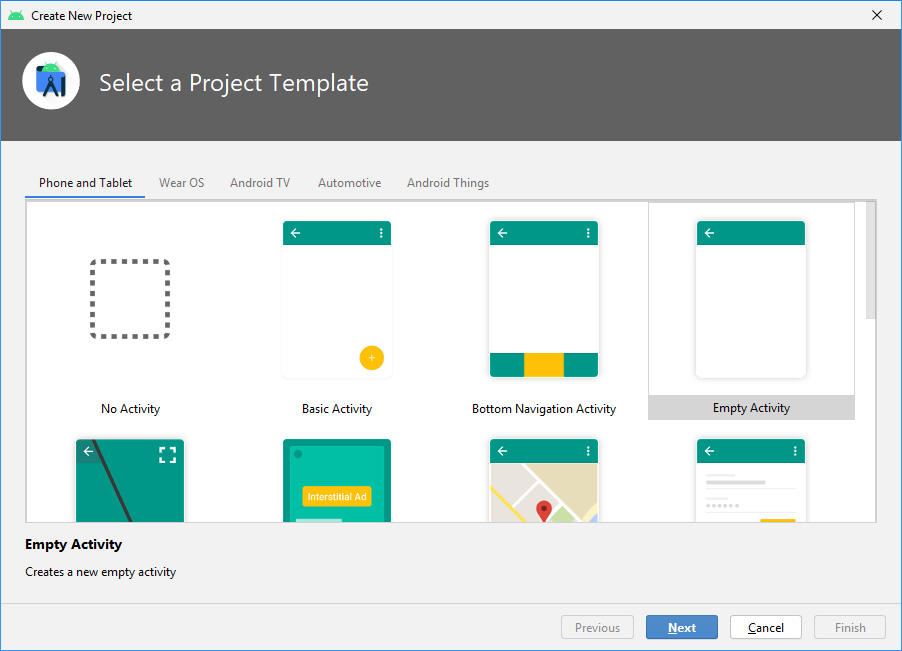


Рисунок 2 – Выбор макета приложения

Нажав «Next», перейдём к окну выбора места сохранения проекта.

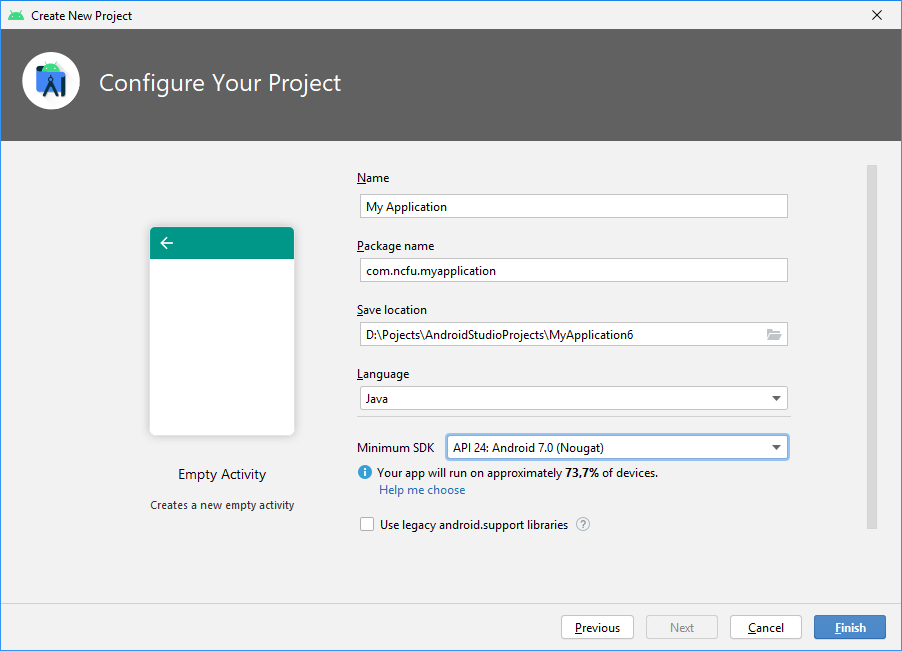


Рисунок 3 – Выбор места создания проекта

1. Далее нажимаем «Финиш».

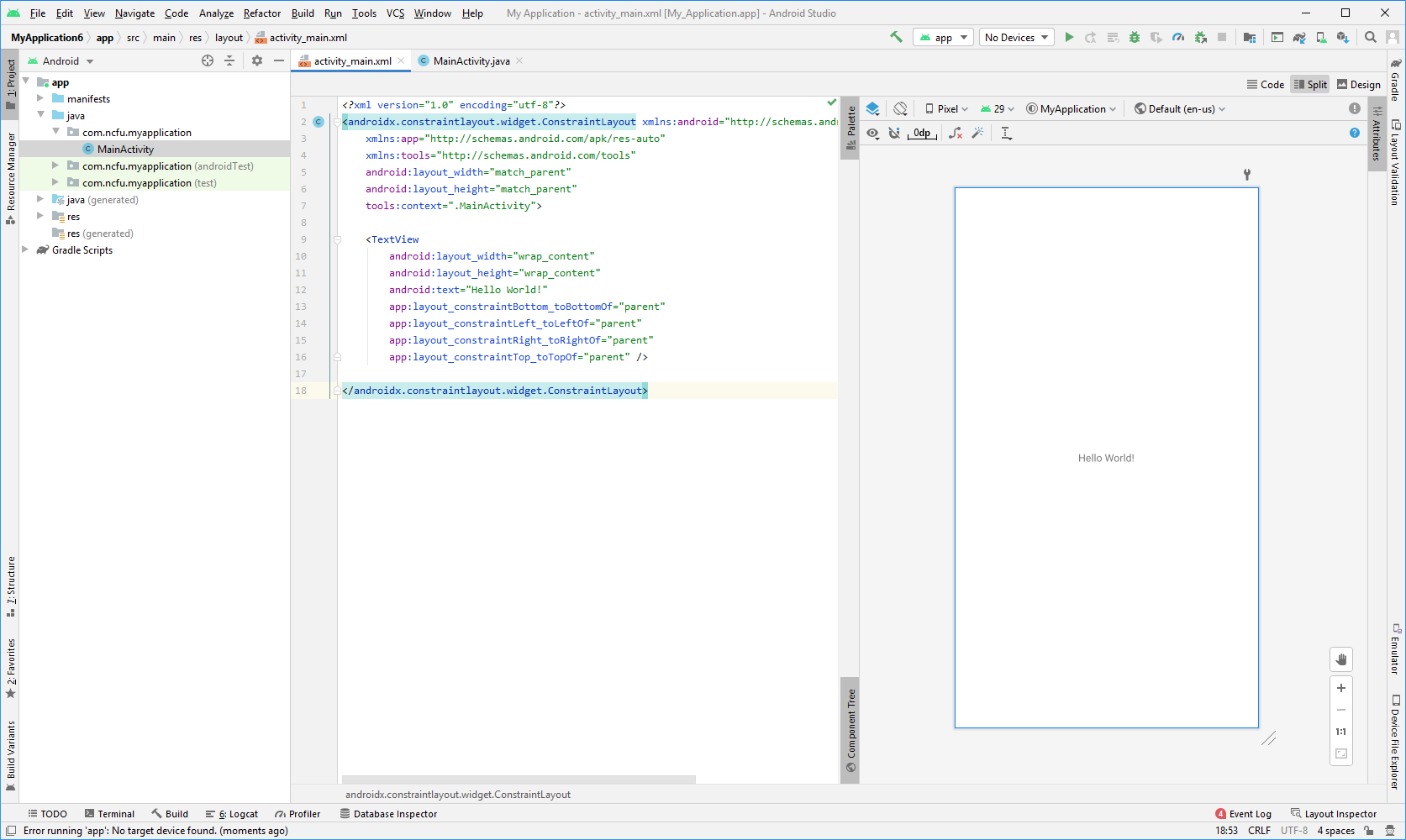


Рисунок 4 – Главное окно приложения

Прежде чем начать программировать, запустите приложение, чтобы проверить, что все компоненты были установлены правильно. В верхней строке меню выберите No Devices→AVD Manager.

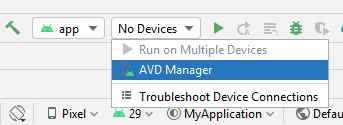


Рисунок 5 – Выбор виртуального устройства

Во время первого запуска необходимо выбрать эмулятор (Если Вы собираетесь тестировать своё приложение на своём андроид-устройстве, необходимо: включить свойства разработчика на устройстве, поставить галочку рядом с пунктом «Разрешить отладку по USB»).

Так как мы хотим проверить сейчас на эмульяторе, рассмотрим данный процесс подробнее.

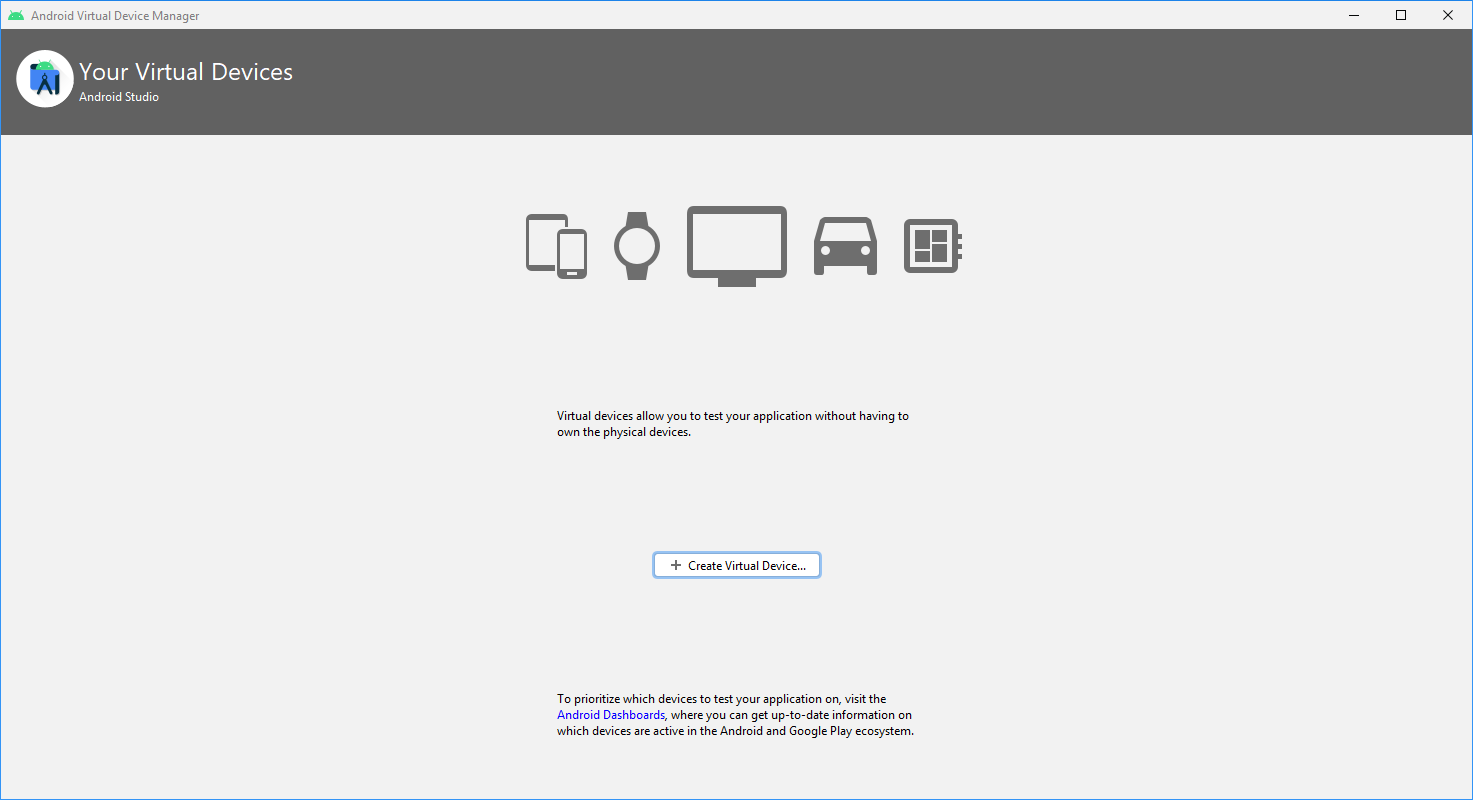


Рисунок 6 – Выбор типа устройства эмулятора

Нажав на «Create Virtual Device», выберите эмулятор под те характеристики, которые Вам более удобны. Например, создадим своё собственное устройство. Найдём в интернете любое устройство на Android (например, Samsung F700 Galaxy Z Flip 8). Перепишем характеристики в новом окне. Нажмите «New Hardware Profile».

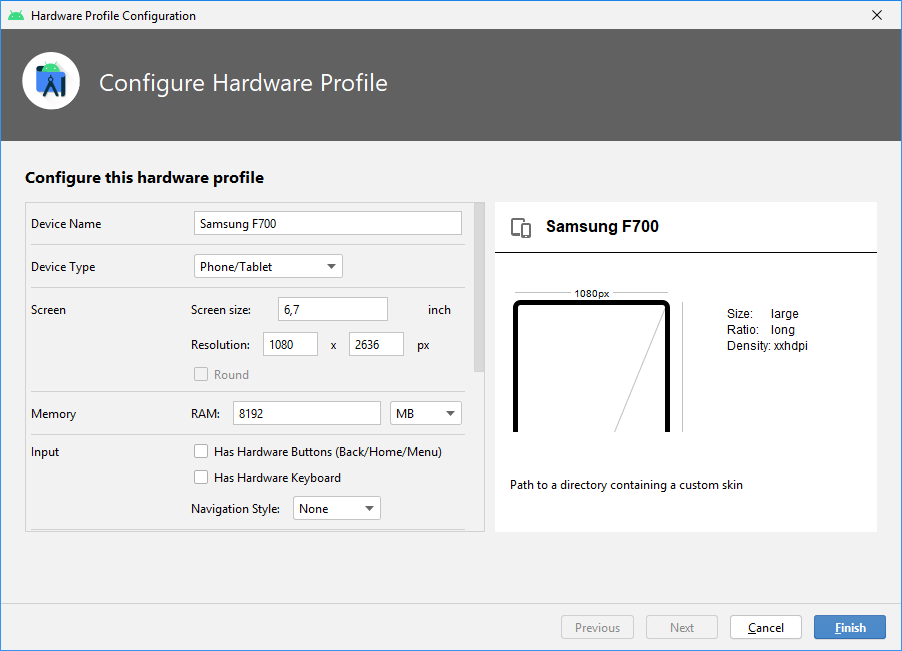


Рисунок 7 – Внесение изменений эмулятора

Теперь необходимо выбрать версию Android для нашего устройства. Скачаем последнюю актуальную версию Android 11.

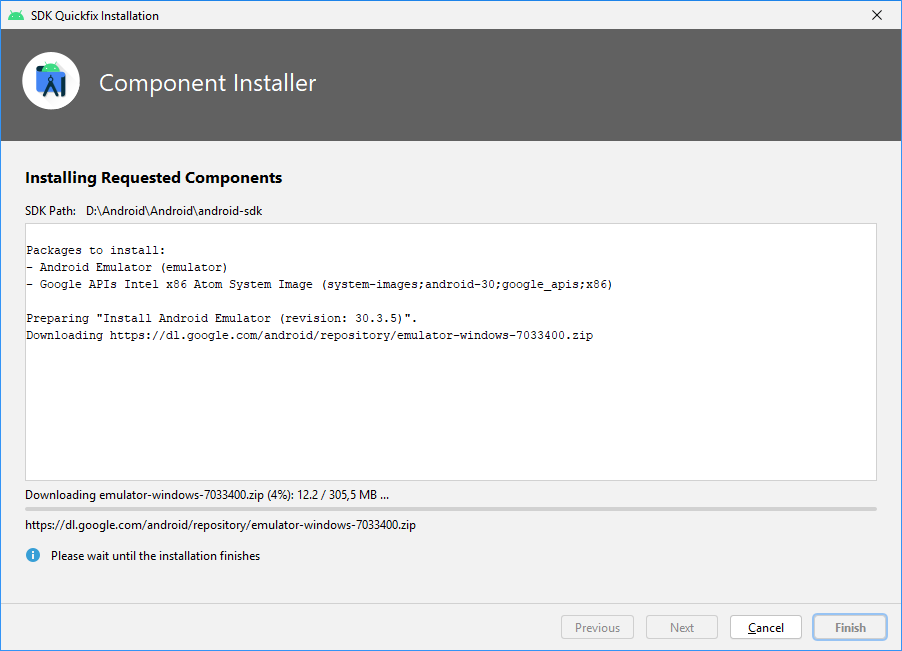


Рисунок 8 – Скачивание Android 11

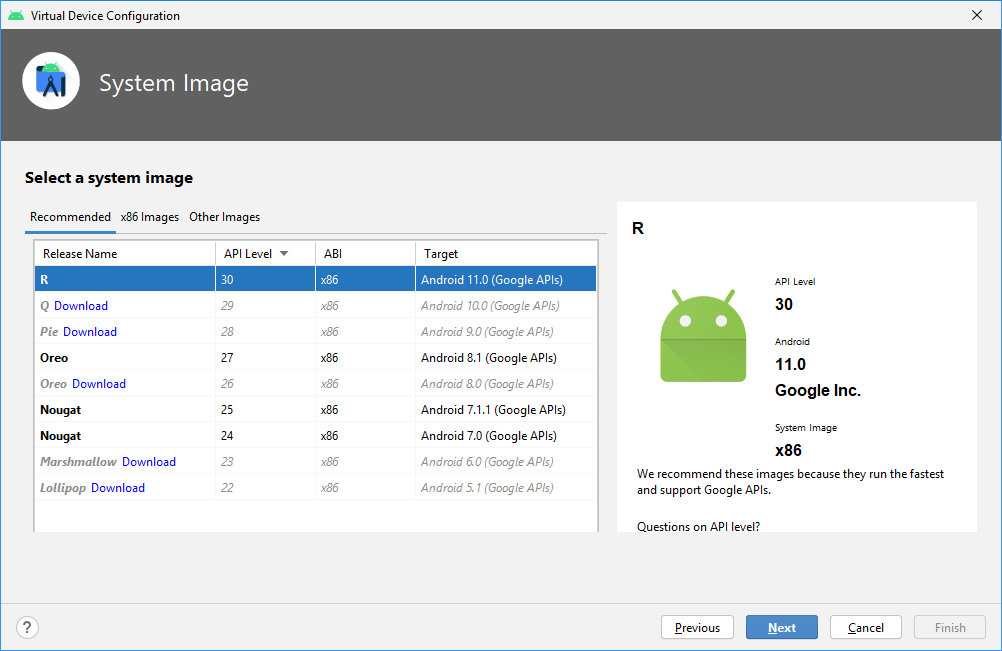


Рисунок 9 – Выбор версии Android

После создания эмулятора оно будет доступно в списке AVD.



Рисунок 10 – Выбор устройства

Для запуска приложения нажмите комбинацию Shift+F10 или на зеленый треугольник.

Если никаких ошибок во время установки и запуска эмулятора не было, то результатом будет такое же окно, как на рисунке 11.

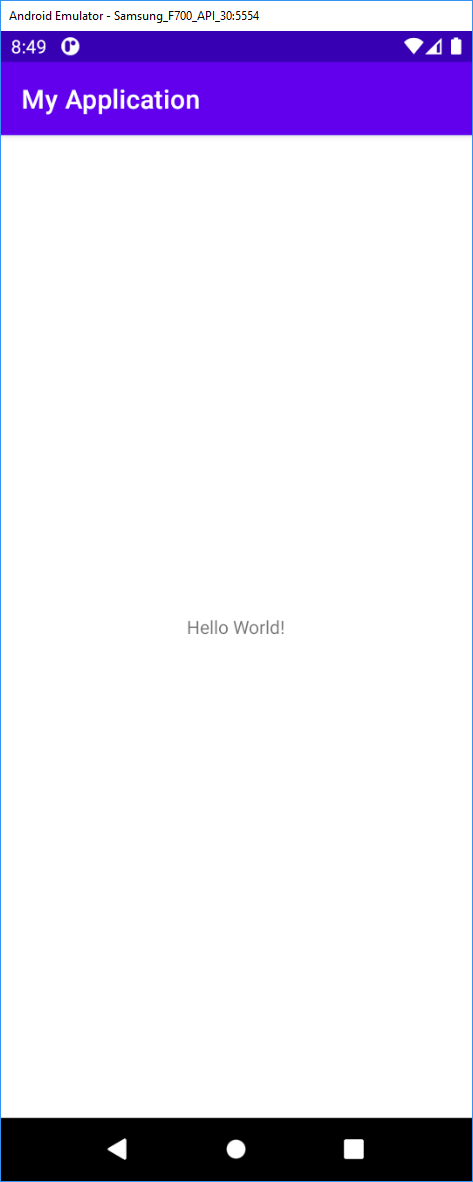


Рисунок 11 – Главное окно эмулятора

Интерфейс студии интуитивно понятен:

1. Дерево решений. Предоставляет доступ к различным файлам проекта. В папке app/java/com.example../ располагаются файлы на языке java. В папке app/res/layout/ файлы, которые отвечают за интерфейс вашей программы (активность). Папка app/res/drawable будет содержать файлы для рисования (подробнее во второй лабораторной работе). Папка app/res/values/ содержит файлы, в которых хранится информация о константах приложения (например, цвет фона, название приложения, и тд).
2. Панель управления проектом. Содержит стандартные возможности работы с проектом, такие как: открыть, сохранить, запустить и тд.
3. Рабочая область. Здесь будет располагаться Ваш код.

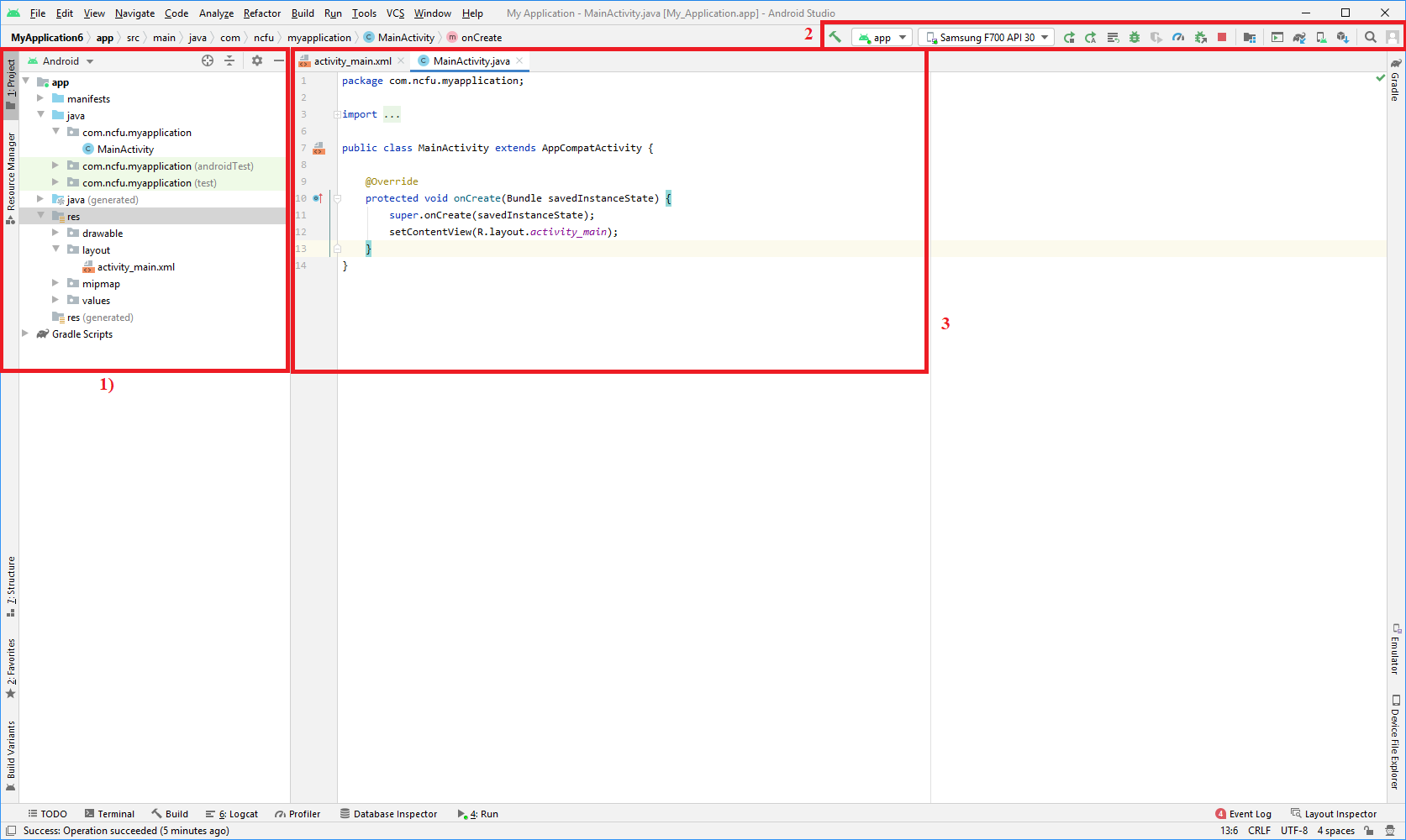


Рисунок 12 – Интерфейс Java

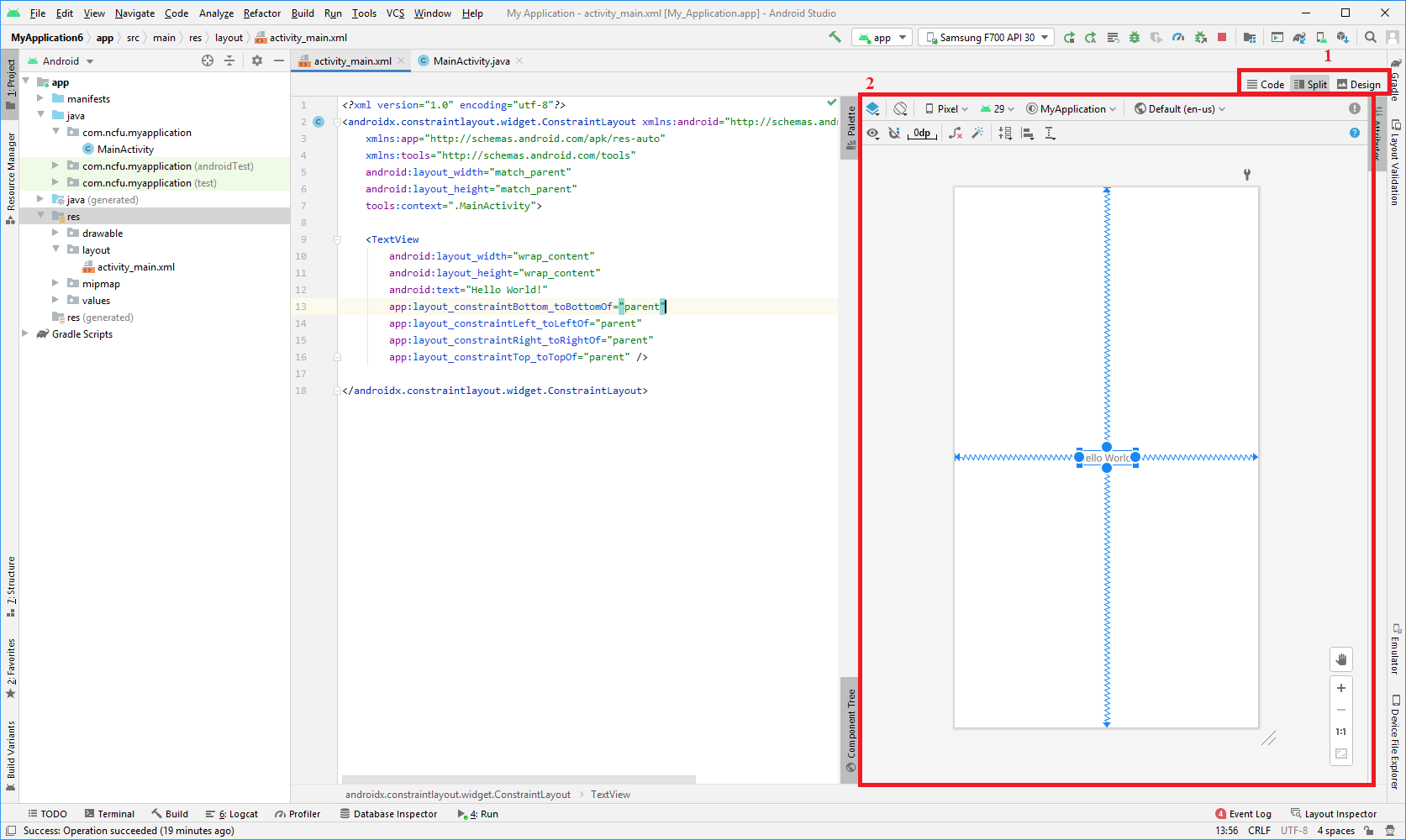


Рисунок 13 – Интерфейс xml

На рисунке 13 показан внешний вид студии во время работы с интерфейсом приложения.

1. Предпросмотр. Область, которая помогает увидеть расположение объектов, созданных с помощью языка xml. Здесь можно выбрать эмулятор, похожий на тот, который запускает приложение.
2. Кнопка, позволяющая выбрать, как будет выглядеть область предпросмотра.

Дизайн приложения разрабатывается с помощью языка XML. Процесс проектирования дизайна приложения отдаленно напоминает разработку web-страниц на языке HTML, однако об этом мы поговорим позже. Дизайн страниц (код на языке XML) хранится в файлах с расширением \*.xml. Так, в обозревателе решений(Project) дизайн главной страницы соответствует файлу activity\_main.xml. Однако о проектировании дизайна с помощью XML-разметки мы поговорим на следующих занятиях. Так как мы учимся в университете, нам главное добраться как можно ближе до истины. Поэтому мы попробуем понять, что собой страница представляет на более низком уровне.

В java/com.example.<nameuser>.<nameproject>/ отобразится еще один похожий по названию файл MainActivity. Этот файл тоже соответствует главной странице приложения, и в нем находится код на языке Java.

Обычное бизнес-приложение Android (как WindowsPhone, и IOS) состоит из определенного набора страниц с возможностью навигации между ними. Каждой странице при проектировании соответствует определенный класс. Программирование на всех платформах в основном и заключается в проектировании классов страниц и взаимодействия между ними.

Итак, рассмотрим содержимое файла MainActivity.

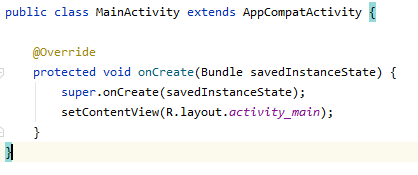


Рисунок 14 – Листинг главного окна

Класс MainActivity определяет функционал главной страницы приложения (эта страница отображается при запуске приложения).

Еще раз обратите внимание. Мы разрабатываем класс MainActivity. При запуске приложения инфраструктура ОС создает объект этого класса. И после этого отображает его на экране телефона. Что значит отобразить класс? Это значит, что объект класса MainActivity обладает свойствами-характеристиками отображения (например, цвет фона, цвет рамки, наличие дочерних элементов и др.), а инфраструктура приложения опрашивает объект об этих свойствах и отображает их на экране.

Изменить название приложения можно двумя способами:

1. Перейти в AndroidManifest.xml (рис.13).

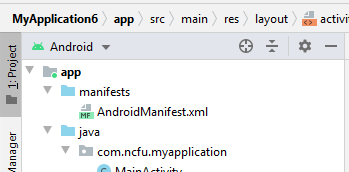


Рисунок 15 – Расположение AndroidManifest.xml в дереве решний

Нажмём на строку android:label="My application". Она изменится, как показано на рисунке.



Рисунок 16 –Содержимое файла AndroidManifest.xml

Держа ctrl нажмём на @string/app\_name. Нас перекинет на страницу с ресурсами типа данных «строка». Все переменные, которые созданы в приложении будут храниться именно в ресурсах. Изменим значение на необходимое.

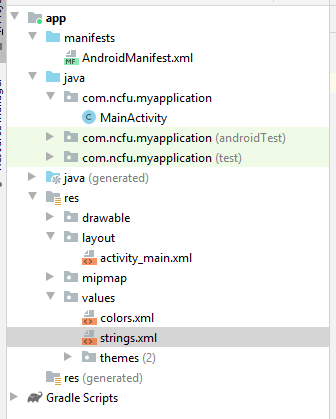


Рисунок 17 – Содержимое папки res (ресурсы)

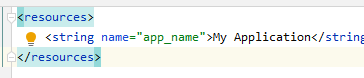


Рисунок 18 – Содержимое файла Strings

1. Внутри MainActivity.java с помощью метода setTitle(), где в параметре передаётся новое название приложения.

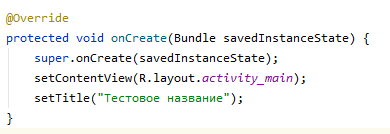


Рисунок 19 – Пример использования setTitle()

Следует отметить, что в языке программирования Java есть правила именования компонентов. Например, каждый класс должен всегда начинаться с большой буквы, в то время как все названия свойств и методов внутри в стиле CamelCase (первое слово с маленькое, а каждое последующее – с большой, например, setNameOfAnyObject()).

**Добавление элементов на страницу приложения.**

Страница приложения целиком и полностью состоит из элементов (текстовые блоки, кнопочки, галочки и многое другое). Каждый из этих блоков представляет собой также объект соответствующего класса.

При создании приложения, Android studio любезно разместила один из таких текстовых блоков (объектов) на странице без нашего ведома. Однако остается некая недосказанность – почему в описании класса эти элементы отсутствуют. Как нам их удалить и добавить те элементы, которые нам действительно нужны.

Попробуем создать новые элементы и разместить их на странице. Так как все элементы страницы являются объектами, для добавления новых элементов, нам нужно создавать новые объекты.

Поиграемся, для начала, с графическими примитивами. В android.graphics есть несколько графических классов, объекты которых мы будем использовать. Создаём класс DrawView, который наследуется от View. Конечно, каждый класс стоит создавать в новом файле. Но, так как для начала необходимо разобраться с простым рисованием, поместим новый класс в тот же файл, где расположена активность MainActivity. Более того, сделаем его вложенным классом.

Внутри определим свойства Paint (кисть) и Rect (прямоугольник).

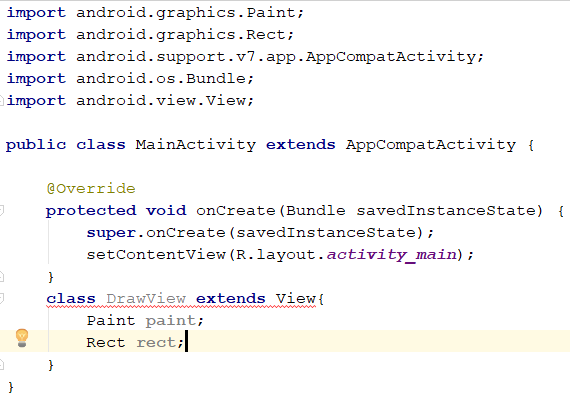


Рисунок 20 – Создание нового класса DrawView

На данном этапе выводит ошибку, которая будет исправлена после создания конструктора класса. Собственно, создадим конструктор.

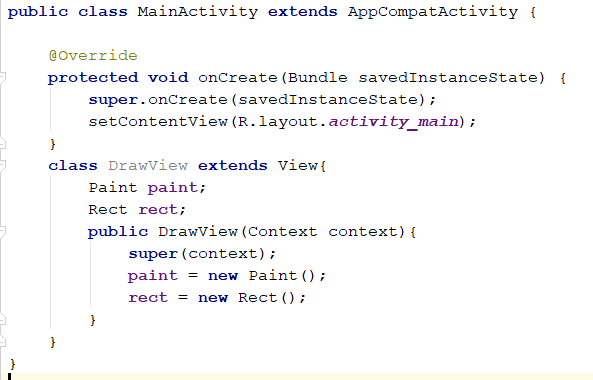


Рисунок 21 – Добавление конструктора

Теперь можно рисовать. Создадим метод, который будет «рисовать» за нас. Рисовать в воздухе не получится, поэтому необходимо создать «холст». Эту роль выполняет Canvas. Раскрасим наш холст и добавим его в контекст страницы. Класс DrawView наследуется от класса View (extends), в котором уже присутствует некоторое количество методов для «рисования». Чтобы не создавать заново всю инфраструктуру, воспользуемся уже существующими методами и переопределим их так, как нам удобно. Переопределяем метод OnDraw(), как показано на рисунке 20.

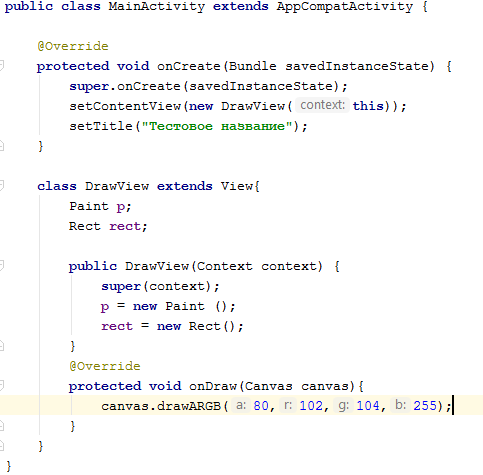


Рисунок 22 – Создание метода для "рисования"

Метод drawARGB() закрашивает цвет заднего фона холста, где a (alpha) – насыщенность цвета, а r, g, b – количество красного, зелёного и синего цвета соответственно. Запустите приложение. Сделайте выводы. Чтобы нарисовать прямоугольник необходимо просто разместить его на канвасе (холсте). Воспользуемся методом drawRect(), где в параметрах передаются координаты отрезка диагонали прямоугольника ( 200, 150 – координаты левого верхнего угла прямоугольника, а 400, 200 – нижнего правого), и кисть.

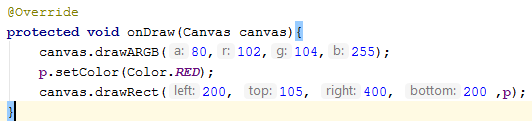


Рисунок 23 – Создание прямоугольника красного цвета

Чтобы нарисовать круг воспользуемся методом drawCircle(). Входные параметры: координаты центра (слева, сверху), радиус и кисть.

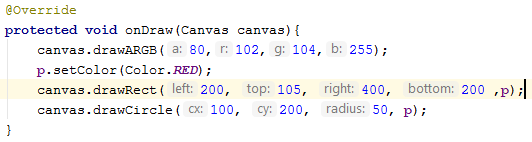


Рисунок 24 – Создание круга красного цвета

Присутствует также возможность рисовать прямую (-ые) с помощью метода drawLine(-s) (просто точка(-и) - drawPoint(-s)). Например, на рисунке 23 показано создание отрезка, где координаты его вершин передаются в качестве массива.

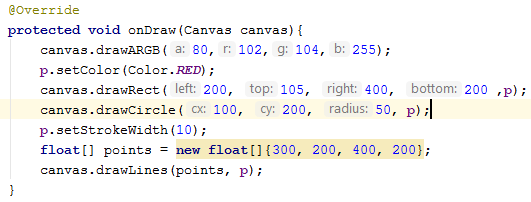


Рисунок 25 – Создание отрезка

setStrokeWidth – метод, позволяющий установить толщину кисти.

Необходимо также учесть, что массив должен содержать обязательно количество точек кратное 4, ибо отрезок – две точки, соединенные между собой, а точка определяется координатами x и y.

Для более сложных фигур используется соответственно более сложный метод рисования. Path – позволяет рисовать фигуры каким угодно способом: это может быть и банальный многоугольник, а может и использовать кривые. Например, чтобы нарисовать треугольник нужно следующее:

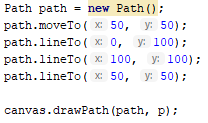


Рисунок 26 – Код треугольника

А для рисования кривых используется метод addArc(). В качестве параметров передаются координаты внутреннего прямоугольника, угол смещения и угол прорисовки. То есть, если нужно нарисовать полукруг с радиусом 50 px, выпуклый к низу, то воспользуемся следующим кодом:

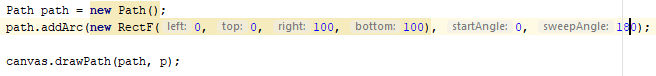


Рисунок 27 – Первый пример кривой

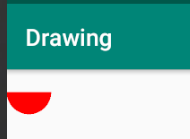


Рисунок 28 – Результат

Для вертикально-выпуклой кривой используем:

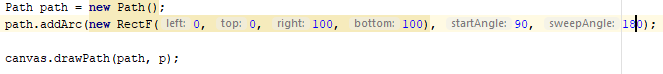


Рисунок 29 – Второй пример кривой



Рисунок 30 – Результат

**Оборудование и материалы:** для выполнения данной лабораторной работы необходим компьютер с установленной операционной системой Windows 7 (8,8.1,10) и программным продуктом Android Studio.

**Указания по технике безопасности:** к выполнению лабораторных работ допускаются студенты, ознакомившиеся с правилами работы в лаборатории, прошедшие инструктаж безопасности.

**Задания:**для выполнения лабораторной работы необходимо выполнить следующее:

1. Запустить и проверить, все ли работает правильно. При запуске сначала загрузится эмулятор мобильной платформы, вслед за чем на нем запустится ваше приложение. Это приложение является заготовкой и кроме как главную страницу ничего не покажет. Запомните, для того, чтобы остановить приложения не обязательно (более того не рекомендуется) закрывать эмулятор. При следующем запуске приложения, эмулятор уже будет загружен и запуск вашего приложения произойдет гораздо быстрее.
2. Указать в имени приложения авторство, а также на странице указать дату рождения, с помощью свойства страницы TextView.
3. Выполнить все описанные в лабораторной работе шаги. Запустите приложение. На экране должен отобразиться красный прямоугольник. Если этого не произошло, внимательно проверьте все выполненные шаги. Обратитесь к преподавателю за помощью. Выполните следующее задание согласно своему варианту. Если не указаны конкретные размеры, отталкиваться от того что ширина должны быть не меньше 150 пикселей. Площадь круга π\*r2
4. Создание фигур с помощью графических примитивов. Необходимо использовать минимум 3 цвета и ломаные.

**Варианты для задания 3:**

1. Зеленый прямоугольник, у которого высота в 2 раза больше ширины.
2. Синий круг, занимающий половину площади экрана.
3. Желтый прямоугольник, занимающий половину площади экрана.
4. Полупрозрачный зеленый овал, касающийся всех граней экрана.
5. Пурпурный квадрат, площадь которого в 2 раза больше периметра.
6. Розовый круг, площадь которого равна периметру.
7. Синий квадрат, периметр которого в 2 раза меньше периметра экрана.
8. Зеленый квадрат, площадь которого равна периметру.
9. Оранжевый квадрат, площадь которого в 2 раза меньше площади экрана.
10. Серый прямоугольник, с одинаковым расстоянием от граней экрана до граней прямоугольника.
11. Красный овал, с длиной вертикальной полуоси в 3 раза большей горизонтальной.
12. Белый круг, площадь которого в 2 раза больше периметра.
13. Желтый овал с одинаковым расстоянием от граней экрана до ближайшей точки на овале.
14. Розовый круг, площадь которого в 3 раза меньше площади экрана.

**Варианты для задания 4:**

1. Дом с дверью и окном.
2. Легковой автомобиль.
3. Грузовой автомобиль.
4. Логотип СКФУ.
5. Железная дорога, уходящая за горизонт.
6. Новогодняя елка.
7. Котик.
8. Велосипед.
9. Ножницы.
10. Куб с разноцветными гранями.
11. Солнце со смайлом.
12. Набор из 15 вложенных прямоугольников со случайными параметрами.
13. Набор из 15 вложенных кругов со случайными параметрами.
14. Кнопочный телефон.

**Содержание отчета:** отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям. Требования по форматированию: Шрифт TimesNewRoman, интервал – полуторный, поля левое – 3 см., правое – 1,5 см., верхнее и нижнее – 2 см. Абзацный отступ – 1,25. Текст должен быть выравнен по ширине.

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В конце отчета приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Каждый рисунок должен располагаться по центру страницы, иметь подпись (Рисунок 1 – Создание подсистемы) и ссылку на него в тексте.

**Контрольные вопросы:**

1. Понятие виджета.
2. Графические примитивы.
3. Переопределение метода.
4. Наследование.

**Список литературы, рекомендуемый к использованию по данной теме:**

1. Пархимович М. Н. , Липницкий А. А. , Некрасова В. А. Основы интернет-технологий: учебное пособие / Пархимович М. Н. , Липницкий А. А. , Некрасова В. А. – М.: Архангельск: ИПЦ САФУ, 2013 – 366 с.; То же [Электронный ресурс]. - URL:http://biblioclub.ru/index.php?page=book\_red&id=436379&sr=1
2. Соколова В. В. Разработка мобильных приложений: учебное пособие/ Соколова В. В. Томск: Издательство Томского политехнического университета, 2015 – 176с. То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book\_red&id=442808&sr=1