Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра «МОП ЭВМ»

РАСЧЁТНО-ГРАФИЧЕСКАЯ РАБОТА

по дисциплине «Программирование мобильных устройств»

Студент группы 7ИСб-1

Преподаватель В.А. Тихомиров

2021

**Задания**

Разработать измеритель координат (Координаты.exe).

**Содержание**

[Введение 4](#_Toc35433989)

[Ход выполнения работы 5](#_Toc35433991)

[Заключение 12](#_Toc35433992)

[Список использованных источников 13](#_Toc35433993)

Введение

Android Studio - это [интегрированная среда разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) (IDE) для работы с платформой [Android](https://ru.wikipedia.org/wiki/Android" \o "Android). [1]

Android Studio, основанная на программном обеспечении [IntelliJ IDEA](https://ru.wikipedia.org/wiki/IntelliJ_IDEA" \o "IntelliJ IDEA) от компании [JetBrains](https://ru.wikipedia.org/wiki/JetBrains" \o "JetBrains), - официальное средство разработки Android приложений. Данная среда разработки доступна для [Windows](https://ru.wikipedia.org/wiki/Windows" \o "), [OS X](https://ru.wikipedia.org/wiki/OS_X) и Linux.

17 мая 2017 на ежегодной конференции Google I/O, [Google](https://ru.wikipedia.org/wiki/Google_(%D0%BA%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F)" \o "Google (компания)) анонсировал поддержку языка [Kotlin](https://ru.wikipedia.org/wiki/Kotlin" \o "Kotlin), используемого в Android Studio, как официального языка программирования для платформы Android в добавление к [Java](https://ru.wikipedia.org/wiki/Java" \o "Java) и [С++](https://ru.wikipedia.org/wiki/C%2B%2B).

Новые функции появляются с каждой новой версией Android Studio. На данный момент доступны следующие функции:

* Расширенный редактор макетов: [WYSIWYG](https://ru.wikipedia.org/wiki/WYSIWYG), способность работать с [UI](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) компонентами при помощи [Drag-and-Drop](https://ru.wikipedia.org/wiki/Drag-and-drop" \o "Drag-and-drop), функция предпросмотра макета на нескольких конфигурациях экрана.
* Сборка приложений, основанная на [Gradle](https://ru.wikipedia.org/wiki/Gradle" \o "Gradle).
* Различные виды сборок и генерация нескольких [.apk](https://ru.wikipedia.org/wiki/.APK) файлов
* [Рефакторинг](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3) кода
* [Статический анализатор кода](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80_%D0%BA%D0%BE%D0%B4%D0%B0&action=edit&redlink=1) (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое.
* Встроенный ProGuard и утилита для подписывания приложений.
* Шаблоны основных макетов и компонентов Android.
* Встроенная поддержка [Google Cloud Platform](https://ru.wikipedia.org/wiki/Google_Cloud_Platform" \o "Google Cloud Platform), которая включает в себя интеграцию с сервисами [Google Cloud Messaging](https://ru.wikipedia.org/wiki/Google_Cloud_Messaging" \o "Google Cloud Messaging) и [App Engine](https://ru.wikipedia.org/w/index.php?title=App_Engine&action=edit&redlink=1" \o "App Engine (страница отсутствует)).
* Новая версия Android Studio 2.1 способна работать с обновленным компилятором Jack, а также получила улучшенную поддержку Java 8 и усовершенствованную функцию Instant Run.
* В Android Studio 3.0 будут по стандарту включены инструменты языка [Kotlin](https://ru.wikipedia.org/wiki/Kotlin" \o "Kotlin) основанные на [JetBrains](https://ru.wikipedia.org/wiki/JetBrains" \o "JetBrains) [IDE](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8). [2]

Ход выполнения работы

При разработке приложения были использованы 5 файлов в Android Studio. Содержимое файла AndroidManifest.xml приведено в листинге 1.

Листинг 1 - AndroidManifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.rgr"**>  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.RGR"**>  
 <**activity android:name=".MainActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 </**application**>  
  
</**manifest**>

Содержимое файла main\_activity.xml приведено в листинге 3.

Листинг 2 - main\_activity.xml

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context="com.example.rgr.MainActivity"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="0.5"  
 android:orientation="vertical"**>  
  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Радианы:"** />  
  
 <**EditText  
 android:id="@+id/editTextNumber"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="number"  
 android:text="0"** />  
  
 <**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Градусы:"** />  
  
 <**EditText  
 android:id="@+id/editTextNumber2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="number"  
 android:text="0"** />  
  
 <**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="X:"** />  
  
 <**EditText  
 android:id="@+id/editTextNumber3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="number"  
 android:text="0"** />  
  
 <**TextView  
 android:id="@+id/textView4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Y:"** />  
  
 <**EditText  
 android:id="@+id/editTextNumber4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="number"  
 android:text="0"** />  
  
 <**TextView  
 android:id="@+id/textView5"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Дистанция:"** />  
  
 <**EditText  
 android:id="@+id/editTextNumber5"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="number"  
 android:text="0"** />  
  
 <**Button  
 android:id="@+id/button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="В центр"  
 android:onClick="setToCenter"**/>  
  
 </**LinearLayout**>  
  
  
 <**RelativeLayout  
 android:id="@+id/move"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="0.5"  
 android:background="@drawable/img"  
 android:orientation="vertical"**>  
  
 <**ImageView  
 android:id="@+id/ImageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:src="@mipmap/ic\_launcher"** />  
  
 </**RelativeLayout**>  
</**LinearLayout**>

Содержимое файла MainActivity.java приведено в листинге 3.

Листинг 3 - MainActivity.java

**package** com.example.rgr;  
  
**import** android.app.Activity;  
**import** android.os.Bundle;  
**import** android.view.MotionEvent;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.EditText;  
**import** android.widget.ImageView;  
**import** android.widget.RelativeLayout;  
  
**public class** MainActivity **extends** Activity **implements** View.OnTouchListener {  
  
 ImageView **mImageView**;  
 ViewGroup **mMoveLayout**;  
 **int mX**;  
 **int mY**;  
 *// координаты X и Y от центра* **double x** = 0;  
 **double y** = 0;  
 EditText **radianInput**, **distInput**, **grInput**, **xInput**, **yInput**;  
 ImageView **imageView**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **radianInput** = (EditText) findViewById(R.id.***editTextNumber***);  
 **distInput** = (EditText) findViewById(R.id.***editTextNumber5***);  
 **grInput** = (EditText) findViewById(R.id.***editTextNumber2***);  
 **xInput** = (EditText) findViewById(R.id.***editTextNumber3***);  
 **yInput** = (EditText) findViewById(R.id.***editTextNumber4***);  
 **imageView** = (ImageView) findViewById(R.id.***ImageView***);  
  
 *//Связываемся с нашими объектами, определяя изображение через заданный ViewGroup:* **mMoveLayout** = (ViewGroup) findViewById(R.id.***move***);  
 **mImageView** = (ImageView) **mMoveLayout**.findViewById(R.id.***ImageView***);  
  
 *//Создаем программно RelativeLayout с параметрами 100\*100:* RelativeLayout.LayoutParams layoutParams = **new** RelativeLayout.LayoutParams(50, 50);  
  
 *//Применяем эти параметры к нашему изображению:* **mImageView**.setLayoutParams(layoutParams);  
 *//И настраиваем ему слушателя (обработчик) прикосновений:* **mImageView**.setOnTouchListener(**this**);  
 }  
  
 *//Обрабатываем прикосновения к объекту:* **public boolean** onTouch(View view, MotionEvent event) {  
  
 *//Определение координат через getRawX() и getRawY() дает  
 //координаты по отношению к размерам экрана устройства:* **final int** X = (**int**) event.getRawX();  
 **final int** Y = (**int**) event.getRawY();  
  
 **switch** (event.getAction() & MotionEvent.***ACTION\_MASK***) {  
  
 *//ACTION\_DOWN срабатывает при прикосновении к экрану,  
 //здесь определяется начальное стартовое положение объекта:* **case** MotionEvent.***ACTION\_DOWN***:  
 RelativeLayout.LayoutParams lParams = (RelativeLayout.LayoutParams) view.getLayoutParams();  
 **mX** = X - lParams.**leftMargin**;  
 **mY** = Y - lParams.**topMargin**;  
  
 **x** = X - **mMoveLayout**.getWidth() / 2;  
 **y** = -Y + **mMoveLayout**.getHeight() + **mMoveLayout**.getHeight() /2 + 70;  
 reset();  
  
 **break**;  
  
 *//ACTION\_MOVE обрабатывает случившиеся в процессе прикосновения изменения, здесь  
 //содержится информация о последней точке, где находится объект после окончания действия прикосновения ACTION\_DOWN:* **case** MotionEvent.***ACTION\_MOVE***:  
 RelativeLayout.LayoutParams layoutParams = (RelativeLayout.LayoutParams) view.getLayoutParams();  
 layoutParams.**leftMargin** = X - **mX**;  
 layoutParams.**topMargin** = Y - **mY**;  
 layoutParams.**rightMargin** = -250;  
 layoutParams.**bottomMargin** = -250;  
 view.setLayoutParams(layoutParams);  
  
 **x** = X - **mMoveLayout**.getWidth() / 2;  
 **y** = -Y + **mMoveLayout**.getHeight() + **mMoveLayout**.getHeight() /2 + 70;  
 reset();  
  
 **break**;  
 }  
 **return true**;  
 }  
  
 **public void** reset() {  
 **xInput**.setText(String.*valueOf*(**x**));  
 **yInput**.setText(String.*valueOf*(**y**));  
 *// Установка градусов* **double** angle;  
 **if** (**x** < 0) {  
 angle = 270 - (Math.*atan*(**y** / -**x**) \* 180 / Math.***PI***);  
 } **else** {  
 angle = 90 + (Math.*atan*(**y** / **x**) \* 180 / Math.***PI***);  
 }  
 **grInput**.setText(String.*valueOf*(angle));  
 *// Установка радиан* **radianInput**.setText(String.*valueOf*(Math.*toRadians*(angle)));  
 *// Установка дистанции* **distInput**.setText(String.*valueOf*(Math.*sqrt*(**x**\***x** + **y**\***y**)));  
  
 }  
  
 **public void** setToCenter(View view) {  
 **x** = 0;  
 **y** = 0;  
 **mImageView**.setX(**mMoveLayout**.getPivotX() - **mImageView**.getWidth() / 2);  
 **mImageView**.setY(**mMoveLayout**.getPivotY() - **mImageView**.getHeight() / 2);  
 reset();  
 }  
}

Результат работы приложения показан на рисунках 1-5.

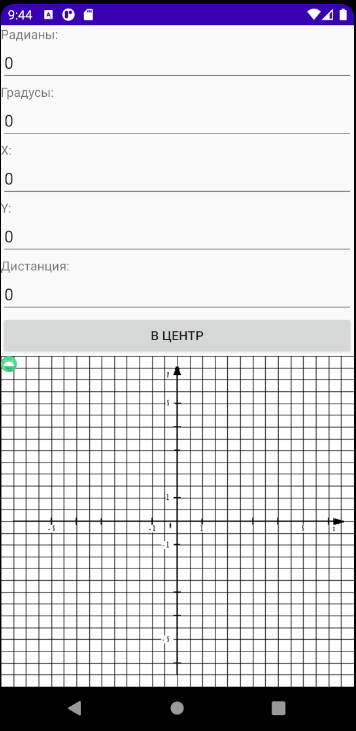


Рисунок 1 – Стартовый экран

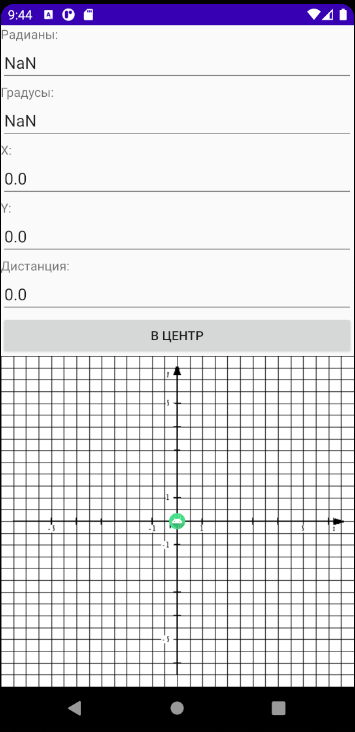


Рисунок 2 – Центрирование обьекта

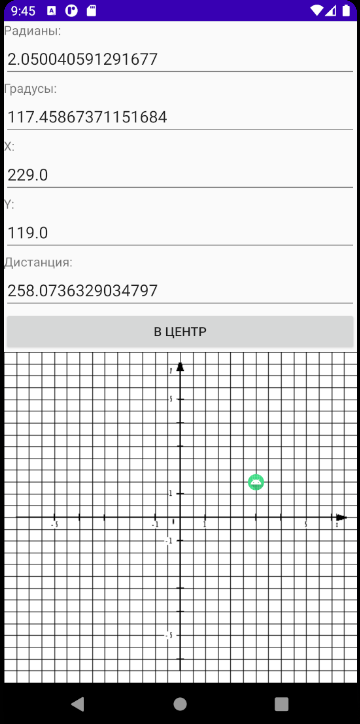


Рисунок 3 – Перемещение объекта

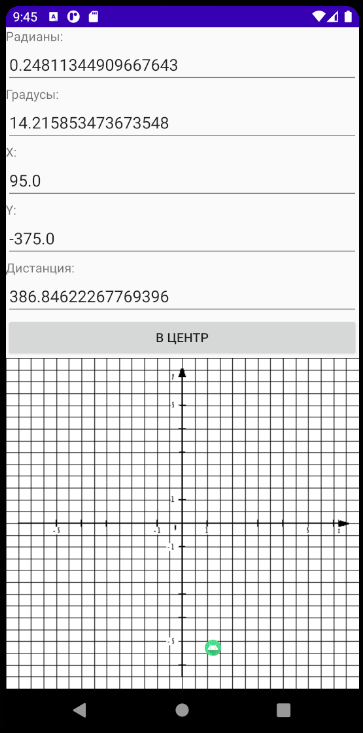


Рисунок 4 – Перемещение объекта 2

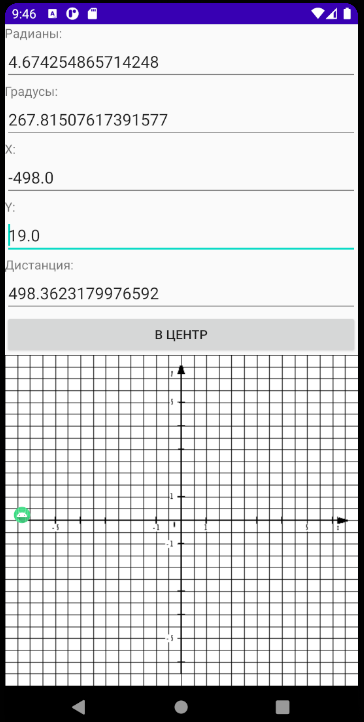


Рисунок 5 – Перемещение объекта 3

Заключение

В ходе выполнения расчётно-графической работы были изучены и рассмотрены возможности [интегрированной среды разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) «Android Studio», в которой было разработано приложение с выпадающим меню.

Полученные навыки и знания будут использоваться в дальнейшем.

Список использованных источников

1. Дейтел, П. Android для разработчиков. 3-е изд. / П. Дейтел, Х. Дейтел, А. Уолд. — СПб.: Питер, 2016. — 512 с.
2. Гриффитс. Д. Head First. Программирование для Android / Дон Гриффитс, Дэвид Гриффитс — СПб.: Питер, 2016. — 704 с.