УДК 004

Создание приложения калькулятор на языке Kotlin

Семченко Регина Викторовна
Приамурский государственный университет имени Шолом-Алейхема
студент

Еровлев Павел Андреевич
Приамурский государственный университет имени Шолом-Алейхема
студент

Аннотация
В данной статье описан процесс создания приложения калькулятор на android устройства с помощью языка Kotlin. Практическим результатом имеется рабочее приложение калькулятор.
Ключевые слова: Kotlin, приложение, андроид, счетчик

Creating a calculator application in Kotlin

Semchenko Regina Viktorovna
Sholom-Aleichem Priamursky State University
student

Erovlev Pavel Andreevich
Sholom-Aleichem Priamursky State University
student

Abstract
This article describes the process of creating a calculator application on an android device using the Kotlin language. The bottom line is a working calculator application.
Keywords: Kotlin, application, android, counter

Kotlin был открыт как один из языков программирования для Android, его можно применять для создания Android приложений, вместо JavaScript. Google открыл Kotlin как один из инструментов для разработчиков Android, представили его 17 мая 2017 года.

Цель данной статьи создать Android приложение калькулятор на языке Kotlin.

С.В.Виноградов и Р.Н.Воротняк рассмотрели новый язык Kotlin, как хорошая замена языку Java. Представляя, что у этого языка много возможностей и что он легче для понимания и обучения[1]. В.И.Макаров провел в своей статье анализ способой для создания пользовательского интефейса при разработке приложений, которые применяются в среде

разработке AndroidStudio [2]. Так же А.И.Долженко совместно с С.А.Глушенко провели анализ целесообразности разработки мобильного приложения для android устройств, а так же разработали собственное приложение на android [3]. В статье Р.В.Мальчева и С.В.Кривошеева был выполнен анализ архитектурной системы ARM как опаратной основы для создания симуляторов т/с [4]. Е.Н.Колмакова рассмотрела се преимущества языка Kotlin перед всеми остальными, вывела как плюсы, так и минусы, а так же провела основные возможности, которые отсутствуют в Java[5].

Для начала необходимо переконвертировать «gadle» файл с Java в Kotlin, для этого необходимо быстрым набором вызвать функцию «win+shift+a», либо если операционная система macOS, то командой «command+shift+a». После чего «gadle» файл поменяет свой язык(рис.1).

```
buildscript {
    ext.kotlin_version = '1.1.2-4'
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.3.1'
        classpath "org.jetbrains.kotlin:kotlin-gradle-
plugin:$kotlin_version"


they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

Рисунок 1 – файл «gradle»

Дальше следует изменить файл, дополнив версию андройда, установить зависимости, и исключить некоторые Java файлы, для стабильной работы скрипта (рис.2).

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.3"
    defaultConfig {
        applicationId "com.example.irwancannady.aritmatika"
        minSdkVersion 15
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-
android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-
core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-
annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    testCompile 'junit:junit:4.12'
    compile "org.jetbrains.kotlin:kotlin-stdlib-jre7:$kotlin_version"
}
repositories {
    mavenCentral()
}
```

Рисунок 2 – Добавление операций

Теперь можно заниматься стилистикой приложения, для этого добавим код в «activity_main»(рис.3-4).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_margin="16dp"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:context="com.example.irwancannady.aritmatika.MainActivity">

    <EditText
        android:hint="0"
        android:id="@+id/edt_satu"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <EditText
        android:hint="0"
        android:id="@+id/edt_dua"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/edt_satu"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <Button
            android:id="@+id/btn_tambah"
            android:layout_width="50dp"
            android:layout_height="wrap_content"
            android:layout_below="@+id/edt_dua"
            android:text="+" />

        <Button
            android:id="@+id/btn_kurang"
            android:layout_width="50dp"
            android:layout_height="wrap_content"
            android:text="-" />

        <Button
            android:id="@+id/btn_kali"
            android:layout_width="50dp"
            android:layout_height="wrap_content"
            android:text="*" />
```

Рисунок 3 – код интерфейса

```
        <Button
            android:id="@+id/btn_bagi"
            android:layout_width="50dp"
            android:layout_height="wrap_content"
            android:text="/" />
    </LinearLayout>

    <LinearLayout
        android:layout_marginTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hasilnya : " />

        <TextView
            android:id="@+id/tv_result"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="0" />

    </LinearLayout>


</LinearLayout>
```

Рисунок 4 – код интерфейса

Интерфейс готов, теперь дело осталось за функциональной частью, для этого в «MainActivity» добавим основные функции, это сложение, вычитание, умножение, деление(рис.5-6).

```
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast




class MainActivity : AppCompatActivity() {


    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val edt1 = findViewById(R.id.edt_satu) as EditText
        val edt2 = findViewById(R.id.edt_dua) as EditText
        val result = findViewById(R.id.tv_result) as TextView
        val btnTambah = findViewById(R.id.btn_tambah) as Button
        val btnKurang = findViewById(R.id.btn_kurang) as Button
        val btnKali = findViewById(R.id.btn_kali) as Button
        val btnBagi = findViewById(R.id.btn_bagi) as Button

        btnTambah.setOnClickListener {

            val a: String = edt1.text.toString()
            val b: String = edt2.text.toString()

            if(edt1.text.isEmpty() && edt2.text.isEmpty()){
                Toast.makeText(applicationContext,"silahkan isi",
Toast.LENGTH_SHORT).show()
            } else {
                val c = a.toInt()
                val d = b.toInt()
                val e = c + d
                result.setText(e.toString())
            }
        }

        btnKurang.setOnClickListener {
            v ->  kurang()
        }

        btnKali.setOnClickListener {
            v -> result.setText(kali(edt1.text.toString().toInt(),
edt2.text.toString().toInt()).toString())
        }

        btnBagi.setOnClickListener{
            v -> result.text = (getEdt1() / getEdt2()).toString()
        }
    }
```

Рисунок 5 – функции приложения

```
        val edt1 = findViewById(R.id.edt_satu) as EditText
        val edt2 = findViewById(R.id.edt_dua) as EditText
        val result = findViewById(R.id.tv_result) as TextView
        if(edt1.text.isEmpty() || edt2.text.isEmpty()){
           Toast.makeText(applicationContext,"silahkan isi",
Toast.LENGTH_SHORT).show()
        } else {
          val a: Int = edt1.text.toString().toInt()
          val b: Int = edt2.text.toString().toInt()
          val c: Int = a - b
          result.setText(c.toString())
       }
    }



    fun kali(a: Int, b: Int): Int {
        return a * b
    }



    fun getEdt1(): Int {
        val edt1 = findViewById(R.id.edt_satu) as EditText
        val a: String = edt1.text.toString()
        return a.toInt()
    }

    fun getEdt2(): Int {
        val edt2 = findViewById(R.id.edt_dua) as EditText
        val b: String = edt2.text.toString()
        return b.toInt()
    }
}
```

Рисунок 6 – код приложения

Код готов, а значит и приложение готово.

Язык Kotlin не сложный и довольно новый язык и именно поэтому сейчас стоит начать его изучение.

Практическим результатом данной статьи является рабочее мобильно приложение для андройд устройств на языке Kotlin.

Библиографический список

1. Виноградов С.В., Воротняк Р.Н. Язык программирования "Kotlin"// Вестник научного общества студентов, аспирантов и молодых ученых. 2017. № 4 . С. 8-10.
2. Макаров В.И. Особенности разработки пользовательского интерфейса для android-приложений в среде разработки android studio// Современные научные исследования и инновации. 2017. № 7-5 (43). С. 47-55.
3. Долженко А.И., Глушенко С.А. Разработка мобильного приложения для тсж на платформе android // Труды Международного симпозиума

«Надежность и качество». 2014. №5. С. 14-20.
4. Мальчева Р.В., Кривошеева С.В. Разработка симуляторов транспортных средств с использованием операционной системы android // Автоматика. Вычислительная техника. 2012. №1. С. 24-30.
5. Колмакова Е.Н. Обзор особенностей языка программирования kotlin // Теория и практика современной науки. 2016. №3 (9). С. 208-211.