**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное

учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Отчет по лабораторной работе №4 по дисциплине

**«Разработка программного обеспечения для ОС Android»**

Основы языка Kotlin

Выполнил:

Студент группы 8И7А \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Леонов

Проверил:

Ст. преп. ОИТ ИШИТР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Дорофеев

Томск 2020

# Задание

Разработайте программу-тренажёр для проверки математических навыков:

1. В верхней части экрана программа отображает случайно сгенерированное математическое уравнение, например, «5+7=» или «6\*4=».
2. В средней части экрана располагается список с вариантами ответа, только один из которых является верным.
3. В нижней части экрана находится статистическая информация: количество верных и ошибочных ответов.
4. При нажатии на элемент списка программа проверяет правильность выбора и увеличивает соответствующий показатель верных или ошибочных ответов, после чего генерирует новое задание.
5. Операция (сложение, умножение или вычитание) также должна генерироваться случайным образом.

Возможно усложнение заданий (увеличение разрядности чисел) если процент правильных ответов превышает определенный порог.

# Текст программы

<https://github.com/8ait/android-course>

**MainActivity.kt**

package com.leonovalexandr.lab4

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.widget.ArrayAdapter

import android.widget.ListView

import android.widget.TextView

import kotlin.random.Random

import kotlin.random.nextInt

class MainActivity : AppCompatActivity() {

private val answerCount = 4

private val ranges = mapOf(

Level.EASY to 0..5,

Level.MEDIUM to 5..20,

Level.HARD to 20..50

)

private val operationFuncs = listOf(

Operation("+") { n1: Int, n2: Int -> n1 + n2},

Operation("-") { n1: Int, n2: Int -> n1 - n2},

Operation("\*") { n1: Int, n2: Int -> n1 \* n2}

)

private var rightCount = 0

private var wrongCount = 0

private var equation: Equation? = null

private var answers = ArrayList<Int>()

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

startRound()

}

// Начать раунд.

private fun startRound() {

createEquation()

generateAnswers()

setScreen()

}

// Подготовить экран

private fun setScreen(){

val listView: ListView = findViewById(R.id.listView)

listView.choiceMode = ListView.CHOICE\_MODE\_SINGLE

// Подготовить пример

val equationView: TextView = findViewById(R.id.eq)

equationView.text = equation?.text

// Подготовить список

val adapter = ArrayAdapter<Int>(this,

android.R.layout.simple\_list\_item\_1, answers)

listView.adapter = adapter

listView.setOnItemClickListener { parent, view, position, id ->

var answer: Int = parent.getItemAtPosition(position) as Int

checkAnswer(answer)

}

// Подоговить иконку правильных ответов

val rightView: TextView = findViewById(R.id.right)

rightView.text = rightCount.toString()

// Подготовить иконку неправильных ответов

val wrongView: TextView = findViewById(R.id.wrong)

wrongView.text = wrongCount.toString()

}

// Проверить ответ.

private fun checkAnswer(number: Int){

if (number == equation?.value){

rightCount++

} else {

wrongCount++

}

startRound()

}

// Создать пример.

private fun createEquation() {

val operation = operationFuncs.random()

val range = ranges[getLevel()] ?: 0..0

equation = Equation(range, operation)

}

// Сгенерировать возможные ответы.

private fun generateAnswers() {

answers.clear()

val rightAnswer = equation?.value

if (rightAnswer != null){

answers.add(rightAnswer)

while(answers.count() != answerCount){

val nextAnswer = Random.nextInt(rightAnswer - 10, rightAnswer + 10)

if (answers.contains(nextAnswer))

continue

answers.add(nextAnswer)

}

}

}

// Получить сложность раунда.

private fun getLevel(): Level {

if (rightCount == 0)

return Level.EASY

if (rightCount / (wrongCount + rightCount).toFloat() > 0.75)

return Level.HARD

if (rightCount / (wrongCount + rightCount).toFloat() > 0.50)

return Level.MEDIUM

return Level.EASY

}

}

**Operation.kt**

package com.leonovalexandr.lab4

// Тип операции над числами

class Operation(name: String, calc: (n1: Int, n2: Int) -> Int) {

var name: String

var calc: (n1: Int, n2: Int) -> Int

init{

this.name = name

this.calc = calc

}

}

**Level.kt**

package com.leonovalexandr.lab4

enum class Level {

EASY,

MEDIUM,

HARD

}

**Equation.kt**

package com.leonovalexandr.lab4

import kotlin.random.Random

import kotlin.random.nextInt

// Пример

class Equation(range: IntRange, operation: Operation) {

// Строка примера

var text: String = ""

get() {return "$n1${operation.name}$n2="}

// Значение

var value: Int = 0

private set

// Ответы

var answers = ArrayList<Int>()

// Первое число

private var n1: Int = 0

// Второе число

private var n2: Int = 0

// Операция

private var operation: Operation = operation

init {

n1 = Random.nextInt(range)

n2 = Random.nextInt(range)

value = operation.calc(n1, n2)

}

}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<TextView

android:id="@+id/eq"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/equation"

android:textAlignment="center"

android:textSize="25dp"

android:textColor="@color/black"

android:textStyle="bold"

android:layout\_margin="20dp"

android:gravity="center\_horizontal" />

<ListView

android:id="@+id/listView"

android:layout\_below="@id/eq"

android:layout\_above="@id/statistic"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"/>

<LinearLayout

android:id="@+id/statistic"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="20dp"

android:layout\_alignParentBottom="true">

<TextView

android:id="@+id/right"

android:layout\_width="100dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:background="@drawable/tag\_green"

android:text="0"

android:textColor="@color/white"

android:textSize="25dp"

android:textStyle="bold"

android:textAlignment="center"

android:gravity="center\_horizontal" />

<TextView

android:id="@+id/wrong"

android:layout\_width="100dp"

android:layout\_weight="1"

android:layout\_height="wrap\_content"

android:background="@drawable/tag\_red"

android:text="0"

android:textColor="@color/white"

android:textSize="25dp"

android:textStyle="bold"

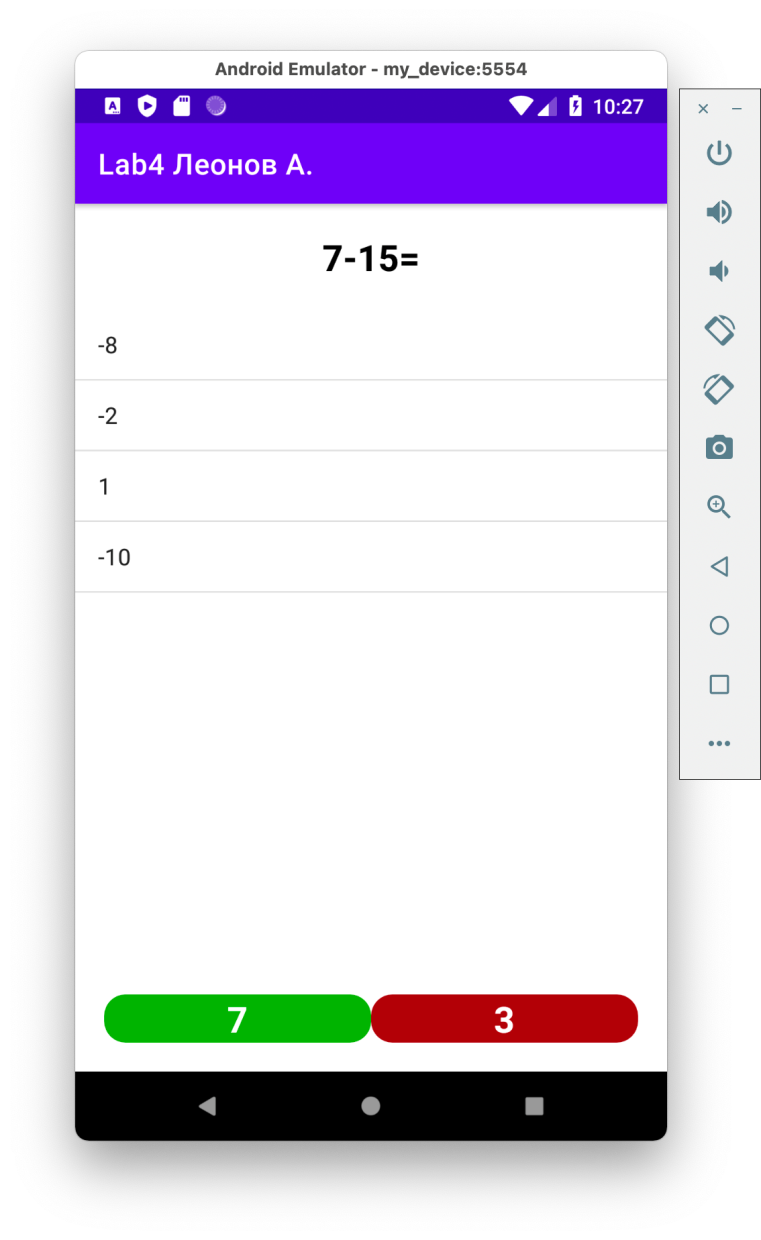
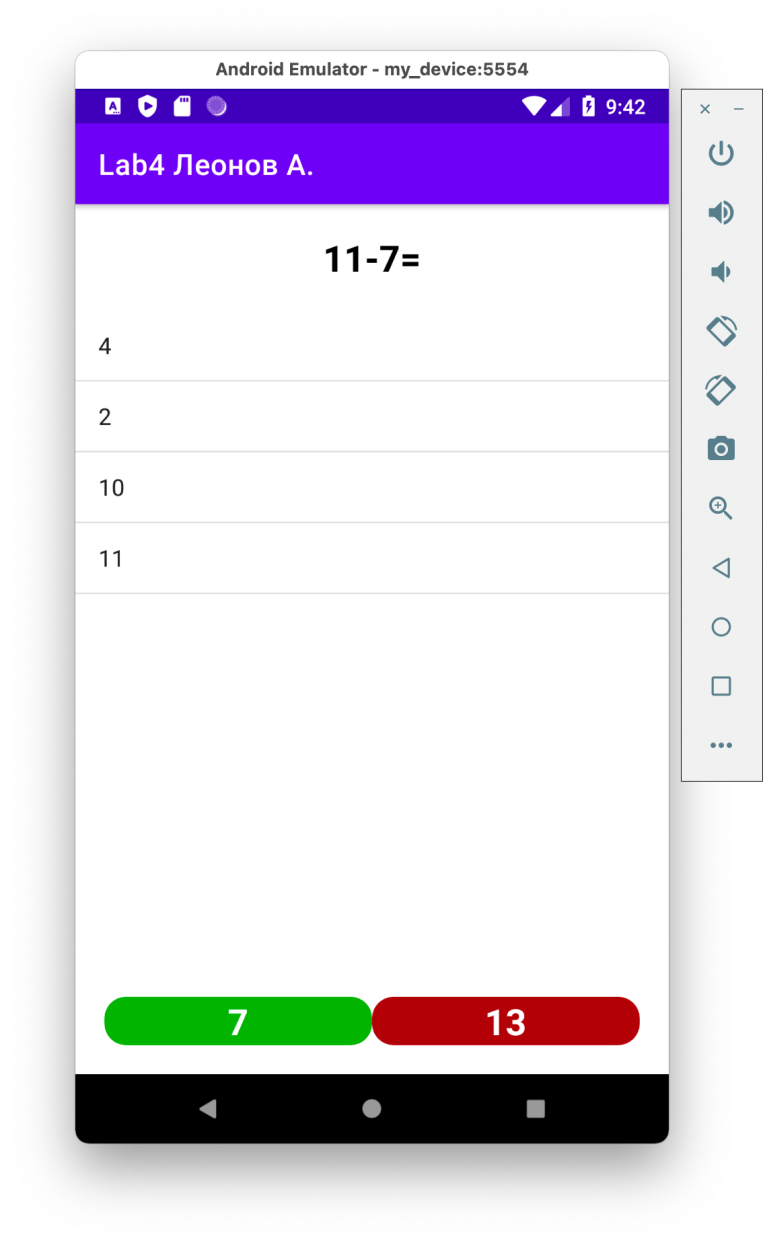
android:textAlignment="center"

android:gravity="center\_horizontal" />

</LinearLayout>

</RelativeLayout>

# Результаты работы



# Выводы

В результате выполнения лабораторной работы я познакомился со списками, использующимися для разработки приложений на Android.