Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт-Петербургский государственный технологический институт

(технический университет)»

Факультет: информационных технологий и управления

Кафедра: систем автоматизированного проектирования и управления

Направление подготовки:09.03.03 Прикладная информатика в химии

Форма обучения: очная

Уровень подготовки: бакалавр

Учебная дисциплина: разработка программного обеспечения для мобильных систем

Группа: 405

**ОТЧЕТ**

**О ЛАБОРАТОРНОЙ РАБОТЕ№2**

ТЕМА: Аппаратное обеспечение мобильного устройства. Создание приложения для управления периферией и сбора сведений от внутренних аппаратных компонентов.

Вариант № 10

Преподаватели Д.Н. Петров

К.А. Захаренко

Исполнители М. А. Шахов

К. И. Емельянова

Санкт-Петербург

2023

**Задание:**

Задание на лабораторную работу представлено в таблице 1.

Таблица 1 – Задание на лабораторную работу

|  |  |  |
| --- | --- | --- |
| № | Наименование отслеживаемого параметра | Наименование устройства |
| 10 | Абсолютное ускорение | Accelerometer |

**Этапы выполнения**

1) Изучение и описание аппаратного модуля (в соответствии с вариантом)

2) Разработка мобильного приложения:

* создание и настройка нового проекта приложения;
* разметка макета графического интерфейса;
* создание управляющей логики;
* тестирование мобильного приложения.

**Описание выполнения лабораторной работы**

**1) Изучение и описание аппаратного модуля (в соответствии с вариантом)**

В Android для работы с датчиками, включая акселерометр, предоставляется API датчиков, который позволяет разработчикам получать данные с физических датчиков устройства, таких как акселерометр, гироскоп, датчик освещённости и другие. Датчик акселерометра особенно важен, поскольку он измеряет ускорение устройства, включая силу гравитации, в м/с² по трём осям: X, Y и Z.

В Android доступ к датчикам осуществляется через класс **SensorManager**, а данные самих датчиков представлены через класс **Sensor**. Для акселерометра используется тип датчика **Sensor.TYPE\_ACCELEROMETER**.

**Основные свойства и методы для работы с акселерометром:**

* **SensorManager sensorManager**: Класс для работы с датчиками устройства. Через **SensorManager** вы можете получить экземпляр акселерометра.
* **Sensor accelerometer**: Представляет датчик акселерометра. С его помощью можно получить данные об ускорении устройства в пространстве.
* **sensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER)**: Метод для получения экземпляра датчика акселерометра.
* **SensorEvent.values[]**: Массив значений, получаемых от датчика. Для акселерометра **values[0]** представляет ускорение по оси X, **values[1]** - по оси Y, **values[2]** - по оси Z.
* **sensorManager.registerListener()**: Метод для регистрации слушателя датчика, который будет получать обновления данных с датчика акселерометра.
* **sensorManager.unregisterListener()**: Метод для отмены регистрации слушателя датчика.

Также в лабораторной работе необходимо вывести все доступные на устройстве датчики.

Датчики, следящие за физическими свойствами и состоянием окружающей среды, предоставляют инновационные способы для улучшения мобильных приложений. Наличие в современных телефонах электронных компасов, датчиков равновесия, яркости и близости открывает целый ряд новых возможностей для взаимодействия с устройством, таких как дополненная реальность и ввод данных, основанный на перемещениях в пространстве.

Датчики в Android делятся на несколько категорий: движения, положения и окружающей среды. Ниже перечислены некоторые виды популярных датчиков:

* Акселерометр (TYPE\_ACCELEROMETER);
* гироскоп (TYPE\_GYROSCOPE);
* датчик освещения (TYPE\_LIGHT);
* датчик расстояния (TYPE\_PROXIMITY);
* датчик магнитных полей (TYPE\_MAGNETIC\_FIELD);
* барометр (TYPE\_PRESSURE);
* датчик температуры окружающей среды (TYPE\_AMBIENT\_TEMPERATURE);
* измеритель относительной влажности (TYPE\_RELATIVE\_HUMIDITY).

В каждом телефоне может быть свой набор датчиков. В большинстве аппаратов есть акселерометр и гироскоп.

За работу с сенсорами отвечает класс **SensorManager**, содержащий несколько констант, которые характеризуют различные аспекты системы датчиков Android, в том числе:

* **Тип датчика.** Ориентация, акселерометр, свет, магнитное поле, близость, температура и т.д.
* **Частота измерений.** Максимальная, для игр, обычная, для пользовательского интерфейса. Когда приложение запрашивает конкретное значение частоты отсчётов, с точки зрения сенсорной подсистемы это лишь рекомендация. Никакой гарантии, что измерения будут производиться с указанной частотой, нет.
* **Точность.** Высокая, низкая, средняя, ненадёжные данные.

Чтобы получить доступ к сенсорам, нужно вызвать метод **getSystemService()**.

**Структура проекта**

В среде разработки мобильных приложений Android Studio Giraffe 17.0.6 для Android был создан проект lab2. На рисунках 1-4 представлен процесс создания проекта.

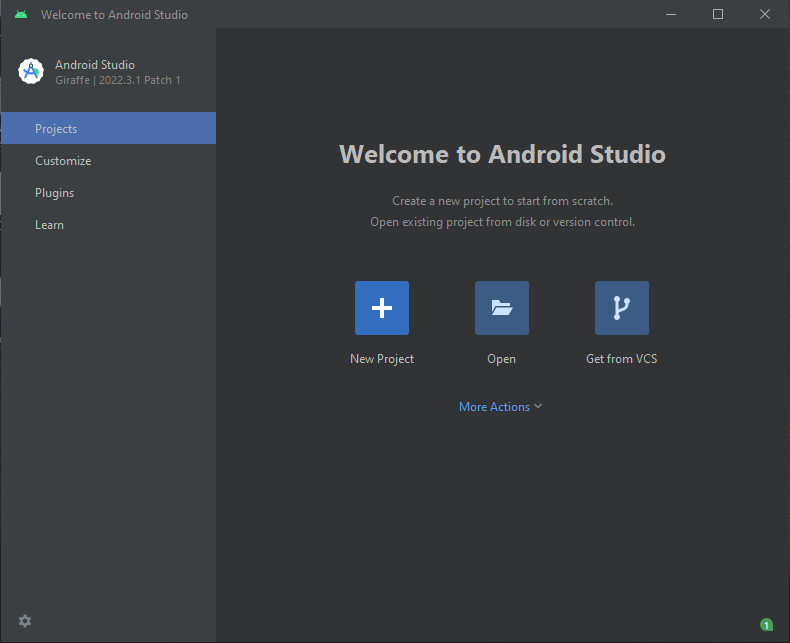


Рисунок 1 – Создание нового проекта

Создать новый проект также возможно из рабочего интерфейса по команде File->New->New Project (рисунок 2)

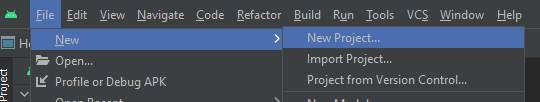


Рисунок 2 – Создание нового проекта

На появившейся форме выбран тип целевого мобильного устройства (Phone and Tablet) и шаблон приложения (Empty Activity до версии Android Studio Giraffe, а начиная с Giraffe, Empty Views Activity, т.к. требуется поддержка языка Java) (рисунок 3).

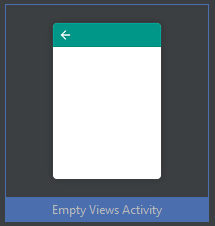
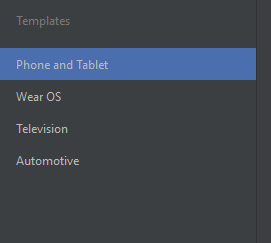


Рисунок 3 – Выбор типа целевого устройства и шаблона приложения

Далее указано наименование приложения, пакета, выбрана директория для сохранения файлов проекта и выбрана минимальную версию SDK, от которой зависит поддержка приложения операционными системами (рисунок 4).

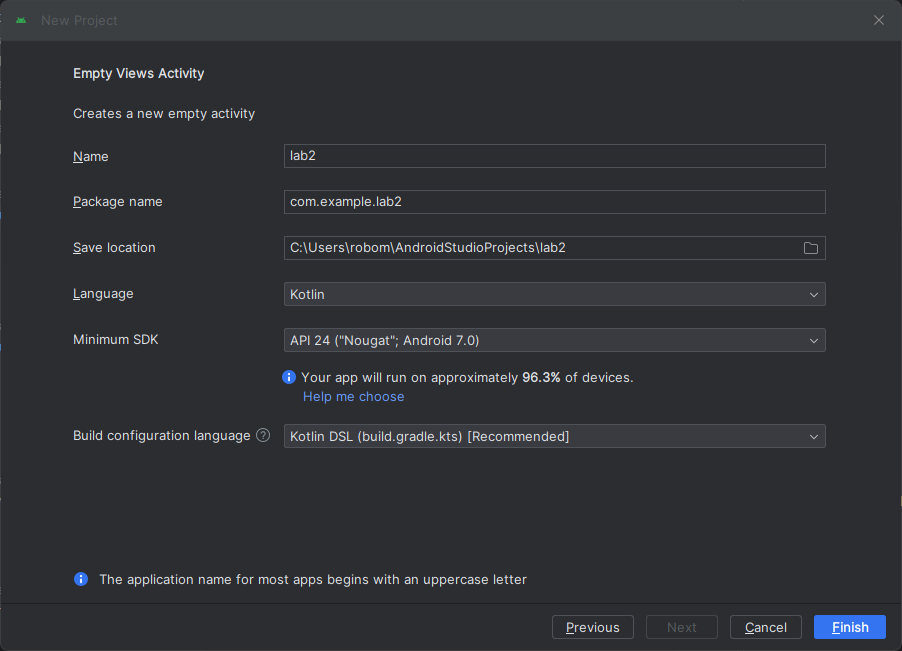


Рисунок 4 – Настройка нового приложения

Минимальная версия SDK – 24, минимальная версия ОС Android – 7.

В таблице 2 представлена структура мобильного приложения.

Таблица 2 – Структура проекта мобильного приложения

|  |  |  |
| --- | --- | --- |
| Каталог | Файлы | Описание |
| manifests | AndroidManifest.xml | Основная информация о программе |
| java/com.example.lab2 | Activity.java | Управляющий код получения информации о датчиках |
| DialogWindow.java | Диалоговое окно с информацией о проекте |
| MainActivity.java | Управляющий код вывода данных с датчика акселерометра |
| java/AccelPackage | AccelerometerSensorListener.java | Код получения и обработки данных с датчика акселерометра |
| res/drawable | icon.xml | Логотип приложения |
| baseline\_keyboard\_backspace\_24 | Логотип кнопки возврата |
| res/layout | activity\_main.xml | Главная активность приложения с информацией об устройстве |
| Activity\_sensors\_list.xml | Дополнительная активность приложения с информацией о датчиках |
| res/values | colors.xml | Используемые цвета в приложении |
| strings.xml | Строковые константы приложения |
| res/values/themes | themes.xml | Файл светлой темы приложения |
| themes.xml (night) | Файл темной темы приложения |
| res/xml | backup\_rules.xml | Настройка резервной копии приложения |
| data\_extraction\_rules.xml | Настройка извлечения данных |

При разработке приложения для запуска и отладки было использовано физическое устройство – мобильный телефон Samsung Galaxy S21 FE. Минимальная версия OC Android – 7, целевая – 13. Минимальная версия SDK – 24, целевая – 31.

Листинг файла манифеста проекта (файл AndroidManifest.xml):

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
 <uses-permission android:name="android.permission.READ\_PHONE\_STATE" />  
 <uses-permission android:name="android.permission.BODY\_SENSORS" />  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@drawable/icon"  
 android:label="@string/app\_name"  
 android:roundIcon="@drawable/icon"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Lab2"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity android:name="com.example.lab2.activity" >  
 </activity>  
 </application>  
</manifest>

Листинг файла строковых констант strings.xml:

<resources>  
 <string name="app\_name">Лабораторная работа №2</string>  
 <string name="sensors">Датчики</string>  
 <string name="infoPhone">Информация об устройстве</string>  
 <string name="info">Информация о проекте</string>  
 <string name="getSensors">Получить датчики</string>  
 <string name="about">Работу выполнили студенты 405 группы СПбГТИ(ТУ) Шахов Михаил и Емельянова Ксения</string>  
 <string name="index">индекс типа: </string>  
 <string name="manufacture">производитель: </string>  
 <string name="ver">версия: </string>  
 <string name="power">мощность: </string>  
 <string name="delay">задержка: [</string>  
 <string name="max">макс.значение: </string>  
 <string name="eps">точность: </string>  
 <string name="m"> мВт</string>  
 <string name="mk">] мкс</string>  
 <string name="minAbsAcc">Минимальное абсолютное ускорение, м/с²:</string>  
 <string name="maxAbsAcc">Максимальное абсолютное ускорение, м/с²:</string>  
 <string name="absAcc">Абсолютное ускорение, м/с²:</string>  
 <string name="ok">OK</string>  
 <string name="ClParanth">)</string>  
 <string name="comma">,</string>  
</resources>

Листинг файла цветовых констант colors.xml:

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
 <color name="backgroundActivity">#C5A9F6</color>  
 <color name="purple">#FF6D3EC1</color>  
</resources>

Используемые изображения:

Логотипы приложения и оформления кнопок размещены в res/drawable (таблица 3)

Таблица 3 – Логотипы для оформления кнопок в приложении

|  |  |
| --- | --- |
| Код разметки | Графический вид |
| icon.xml | |
| <vector android:height="48dp" android:tint="@color/purple" android:viewportHeight="24" android:viewportWidth="24" android:width="48dp" xmlns:android="http://schemas.android.com/apk/res/android"> <path  android:fillColor="@color/purple"  android:pathData="M3,7h2v7H3V7zM0,10h2v7H0V10zM22,7h2v7h-2V7zM19,10h2v7h-2V10zM16,2.01L8,2C6.9,2 6,2.9 6,4v16c0,1.1 0.9,2 2,2h8c1.1,0 2,-0.9 2,-2V4C18,2.9 17.1,2.01 16,2.01zM16,17H8V7h8V17z"/> </vector> |  |
| baseline\_keyboard\_backspace\_24 | |
| <vector android:autoMirrored="true" android:height="24dp"  android:tint="#FFFFFF" android:viewportHeight="24"  android:viewportWidth="28" android:width="28dp" xmlns:android="http://schemas.android.com/apk/res/android">  <path android:fillColor="@android:color/white" android:pathData="M21,11H6.83l3.58,-3.59L9,6l-6,6 6,6 1.41,-1.41L6.83,13H21z"/></vector> |  |

Темы оформления:

Листинг файла светлой темы (файл themes.xml):

<resources xmlns:tools="http://schemas.android.com/tools">  
 <!-- Base application theme. -->  
 <style name="Base.Theme.Lab2" parent="Theme.Material3.DayNight.NoActionBar">  
 <!-- Customize your light theme here. -->  
 </style>  
 <style name="Theme.Lab2" parent="Base.Theme.Lab2" />  
</resources>

Листинг файла темной темы (themes.xml(night)):

<resources xmlns:tools="http://schemas.android.com/tools">  
 <!-- Base application theme. -->  
 <style name="Base.Theme.Lab2" parent="Theme.Material3.DayNight.NoActionBar">  
 <!-- Customize your dark theme here. -->  
 </style>  
</resources>

**Разметка макетов графических интерфейсов**

На рисунках 5-6 представлены разработанные структуры макетов активностей приложения в режиме дизайна.

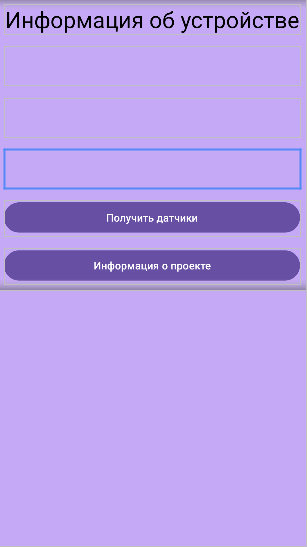


Рисунок 5 – Структура макета activity\_main



Рисунок 6 – Структура макета activity2

На макеты графического интерфейса были добавлены следующие графические компоненты:

* scrollView – вертикальная полоса прокрутки (scrollView3);
* toolbar – панель инструментов (toolbarSnsrs);
* linearLayout – выравнивает все дочерние объекты в одном направлении – вертикально или горизонтально (LinearSnsrs);
* textView – предназначен для отображения текста без возможности редактирования его пользователем (textViewHeader, textView\_Accel, textView\_AccelMin, textView\_AccelMax, textViewSnrsTtl, textViewSnsr);
* imageView – предназначен для отображения изображений (imageViewBck);
* button – кнопка (buttonGetSnrs, buttonAbout).

Листинг файла разметки макета экрана главной активности (файл activity\_main.xml):

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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"  
 android:background="@color/backgroundActivity"  
 tools:context=".MainActivity">  
  
 <ScrollView  
 android:id="@+id/scrollView3"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:background="@color/backgroundActivity"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="1.0">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:visibility="visible">  
  
 <TextView  
 android:id="@+id/textViewHeader"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="@string/infoPhone"  
 android:textAlignment="center"  
 android:textColor="@color/black"  
 android:textSize="30sp" />  
  
 <TextView  
 android:id="@+id/textView\_Accel"  
 android:layout\_width="match\_parent"  
 android:layout\_height="52dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/textView\_AccelMin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="52dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/textView\_AccelMax"  
 android:layout\_width="match\_parent"  
 android:layout\_height="52dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
 <Button  
 android:id="@+id/buttonGetSnrs"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="@string/getSensors" />  
  
 <Button  
 android:id="@+id/buttonAbout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:onClick="showDialog"  
 android:text="@string/info" />  
 </LinearLayout>  
 </ScrollView>  
</androidx.constraintlayout.widget.ConstraintLayout>

Листинг файла разметки макета экрана активности для вывода информации о датчиках, портретная ориентация (файл activity\_sensors\_list.xml):

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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:id="@+id/constraint"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/backgroundActivity"  
 tools:context=".MainActivity">  
  
 <androidx.appcompat.widget.Toolbar  
 android:id="@+id/toolbarSnsrs"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:background="@color/purple"  
 android:minHeight="?attr/actionBarSize"  
 android:theme="?attr/actionBarTheme"  
 app:layout\_constraintBottom\_toTopOf="@+id/scrollView3"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/textViewSnrsTtl"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="sans-serif"  
 android:text="@string/sensors"  
 android:textAlignment="center"  
 android:textColor="@color/white"  
 android:textSize="30sp"  
 app:layout\_constraintBottom\_toBottomOf="@+id/toolbarSnsrs"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="@+id/toolbarSnsrs"  
 app:layout\_constraintTop\_toTopOf="@+id/toolbarSnsrs" />  
  
 <ImageView  
 android:id="@+id/imageViewBck"  
 android:layout\_width="53dp"  
 android:layout\_height="45dp"  
 android:onClick="clickButton"  
 app:layout\_constraintBottom\_toBottomOf="@+id/toolbarSnsrs"  
 app:layout\_constraintEnd\_toStartOf="@+id/textViewSnrsTtl"  
 app:layout\_constraintStart\_toStartOf="@+id/toolbarSnsrs"  
 app:layout\_constraintTop\_toTopOf="@+id/toolbarSnsrs"  
 app:srcCompat="@drawable/baseline\_keyboard\_backspace\_24"  
 tools:ignore="ContentDescription,SpeakableTextPresentCheck,TouchTargetSizeCheck" />  
  
 <ScrollView  
 android:id="@+id/scrollView3"  
 android:layout\_width="368dp"  
 android:layout\_height="666dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"  
 android:background="@color/backgroundActivity"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/toolbarSnsrs">  
  
 <LinearLayout  
 android:id="@+id/LinearSnsrs"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:visibility="visible">  
  
 <TextView  
 android:id="@+id/textViewSnsr"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
 </LinearLayout>  
 </ScrollView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Создание управляющего кода**

Листинг кода activity.java:

package com.example.lab2;  
  
import android.os.Bundle;  
import android.view.View;  
import java.util.ArrayList;  
import java.util.List;  
import android.hardware.Sensor;  
import android.hardware.SensorManager;  
import android.content.Context;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class activity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_sensors\_list*);  
  
 SensorManager sensorManager = (SensorManager) getSystemService(Context.*SENSOR\_SERVICE*);  
 List<Sensor> deviceSensors = sensorManager.getSensorList(Sensor.*TYPE\_ALL*);  
 TextView textView = (TextView)findViewById(R.id.*textViewSnsr*);  
 List<String> listSensorType = new ArrayList<>();  
  
 for (int i = 0; i < deviceSensors.size(); i++) {  
 listSensorType.add(i + 1 + getString(R.string.*ClParanth*) + " " + deviceSensors.get(i).getName() + "\n"+  
 getString(R.string.*index*) + " " + deviceSensors.get(i).getType() + "\n" +  
 getString(R.string.*manufacture*) + " " + deviceSensors.get(i).getVendor() + "\n" +  
 getString(R.string.*ver*) + " " + deviceSensors.get(i).getVersion() + "\n" +  
 getString(R.string.*power*) + " " + deviceSensors.get(i).getPower() + " " + getString(R.string.*m*) + "\n"+  
 getString(R.string.*delay*) + deviceSensors.get(i).getMinDelay() + getString(R.string.*comma*) + deviceSensors.get(i).getMaxDelay() + getString(R.string.*mk*) + "\n"+  
 getString(R.string.*max*) + " " + deviceSensors.get(i).getMaximumRange() +"\n"+  
 getString(R.string.*eps*) + " " + deviceSensors.get(i).getResolution() + "\n"  
 );  
 }  
  
 for (int i = 0; i < deviceSensors.size(); i++) {  
 textView.append(listSensorType.get(i) + "\n" );  
 }  
  
 }  
 public void clickButton(View imageView) {  
 finish();  
 }  
}

Листинг кода DialogWindow.java:

package com.example.lab2;  
  
import android.app.AlertDialog;  
import android.app.Dialog;  
import android.os.Bundle;  
import androidx.fragment.app.DialogFragment;  
import androidx.annotation.NonNull;  
  
public class DialogWindow extends DialogFragment {  
 @NonNull  
 public Dialog onCreateDialog(Bundle savedInstanceState) {  
 AlertDialog.Builder builder=new AlertDialog.Builder(getActivity());  
 return builder.setTitle(getString(R.string.*info*)).setMessage(getString(R.string.*about*)).setPositiveButton(getString(R.string.*ok*),null).create();  
 }  
}

Листинг кода MainActivity.java:

package com.example.lab2;  
  
import static com.example.lab2.R.id.*buttonGetSnrs*;  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
import AccelPackage.AccelerometerSensorListener;  
  
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
 private AccelerometerSensorListener sensorListener;  
 private TextView textViewAccel;  
 private TextView textViewAccelMin;  
 private TextView textViewAccelMax;  
 private float accelMax = 0;  
 private float accelMin = 299792458; //скорость света в вакууме  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Button sensor = findViewById(*buttonGetSnrs*);  
 sensor.setOnClickListener(this);  
  
 textViewAccel = findViewById(R.id.*textView\_Accel*);  
 textViewAccelMin = findViewById(R.id.*textView\_AccelMin*);  
 textViewAccelMax = findViewById(R.id.*textView\_AccelMax*);  
 sensorListener = new AccelerometerSensorListener(this, absoluteAcceleration -> runOnUiThread(() -> {  
 if (accelMin > absoluteAcceleration) {  
 accelMin = absoluteAcceleration;  
 textViewAccelMin.setText(String.*format*(getString(R.string.*minAbsAcc*) + " %.2f", accelMin));  
 }  
 if (accelMax < absoluteAcceleration) {  
 accelMax = absoluteAcceleration;  
 textViewAccelMax.setText(String.*format*(getString(R.string.*maxAbsAcc*) + " %.2f", accelMax));  
 }  
 textViewAccel.setText(String.*format*(getString(R.string.*absAcc*) + " %.2f", absoluteAcceleration));  
 }));  
 }  
  
  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 sensorListener.register();  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 sensorListener.unregister();  
 }  
 public void onClick(View v) {  
 Intent intent = new Intent(this, activity.class);  
 startActivity(intent);  
  
 }  
 public void showDialog(View button2) {  
 DialogWindow dialog = new DialogWindow();  
 dialog.show(getSupportFragmentManager(), "custom");  
 }  
}

Листинг кода AccelerometerSensorListener:

package AccelPackage;  
  
import android.hardware.Sensor;  
import android.hardware.SensorEvent;  
import android.hardware.SensorEventListener;  
import android.hardware.SensorManager;  
import android.content.Context;  
  
public class AccelerometerSensorListener implements SensorEventListener {  
 private final SensorManager sensorManager;  
 private final Sensor accelerometer;  
 private final OnAccelerationChangedListener listener;  
  
 public interface OnAccelerationChangedListener {  
 void onAccelerationChanged(float absoluteAcceleration);  
 }  
  
 public AccelerometerSensorListener(Context context, OnAccelerationChangedListener listener) {  
 this.listener = listener;  
 sensorManager = (SensorManager) context.getSystemService(Context.*SENSOR\_SERVICE*);  
 accelerometer = sensorManager.getDefaultSensor(Sensor.*TYPE\_ACCELEROMETER*);  
 }  
  
 public void register() {  
 sensorManager.registerListener(this, accelerometer, SensorManager.*SENSOR\_DELAY\_NORMAL*);  
 }  
  
 public void unregister() {  
 sensorManager.unregisterListener(this);  
 }  
  
 @Override  
 public void onSensorChanged(SensorEvent event) {  
 if (event.sensor.getType() == Sensor.*TYPE\_ACCELEROMETER*) {  
 float absoluteAcceleration = (float) Math.*sqrt*(Math.*pow*(event.values[0], 2) + Math.*pow*(event.values[1] - 9.81, 2) + Math.*pow*(event.values[2], 2));  
  
 if (listener != null) {  
 listener.onAccelerationChanged(absoluteAcceleration);  
 }  
 }  
 }  
  
 @Override  
 public void onAccuracyChanged(Sensor sensor, int accuracy) {  
 }  
}

**Тестирование мобильного приложения**

Тестирование выполнялось на физическом устройстве (мобильный телефон Xiaomi Redmi Note 9 Pro).

Результат работы мобильного приложения представлен на рисунках 7-15.

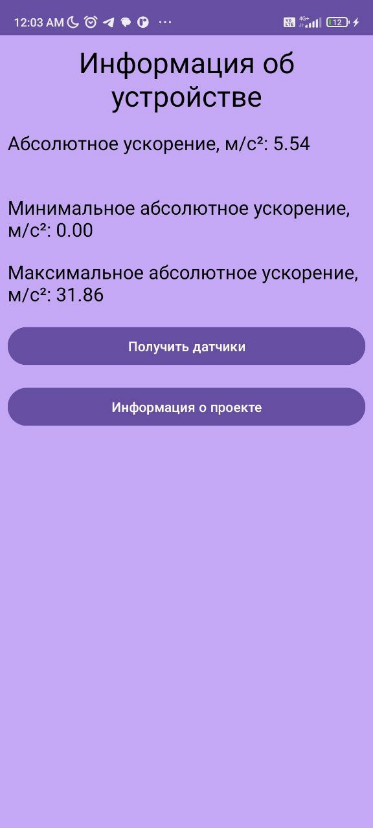


Рисунок 7 – Результат работы главной активности мобильного приложения на светлой теме, портретная ориентация

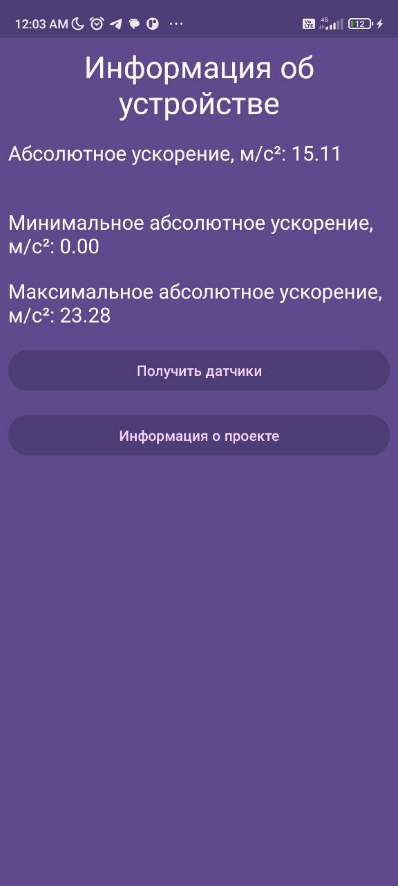


Рисунок 8 – Результат работы главной активности мобильного приложения на темной теме, портретная ориентация

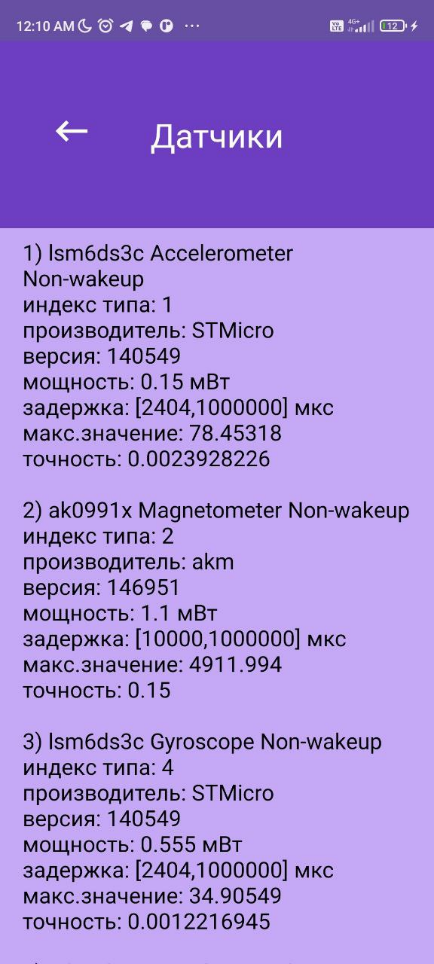


Рисунок 9 – Результат работы дополнительной активности мобильного приложения на светлой теме, портретная ориентация

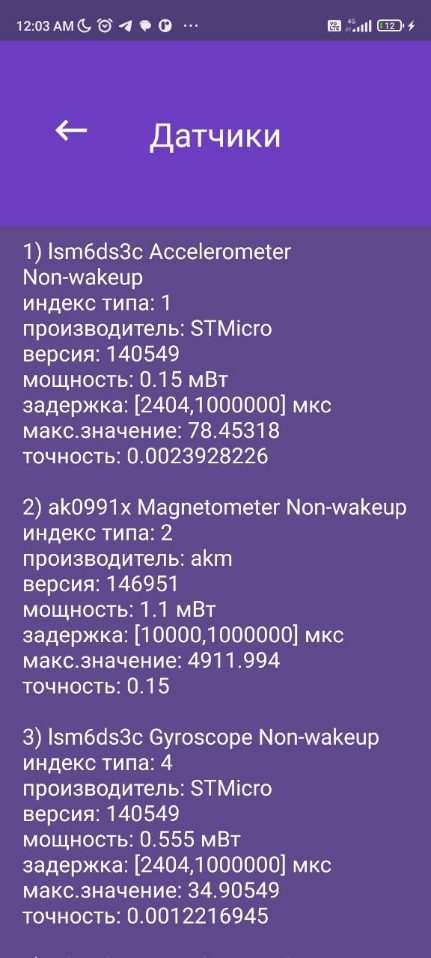


Рисунок 10 – Результат работы дополнительной активности мобильного приложения на темной теме, портретная ориентация

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 11 – Результат работы главной активности мобильного приложения на светлой теме, альбомная ориентация

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 12 – Результат работы главной активности мобильного приложения на темной теме, альбомная ориентация

Изображение выглядит как текст, снимок экрана, Фиолетовый, фиолетовый

Автоматически созданное описание

Рисунок 13 – Результат работы дополнительной активности мобильного приложения на светлой теме, альбомная ориентация

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 14 – Результат работы дополнительной активности мобильного приложения на темной теме, альбомная ориентация

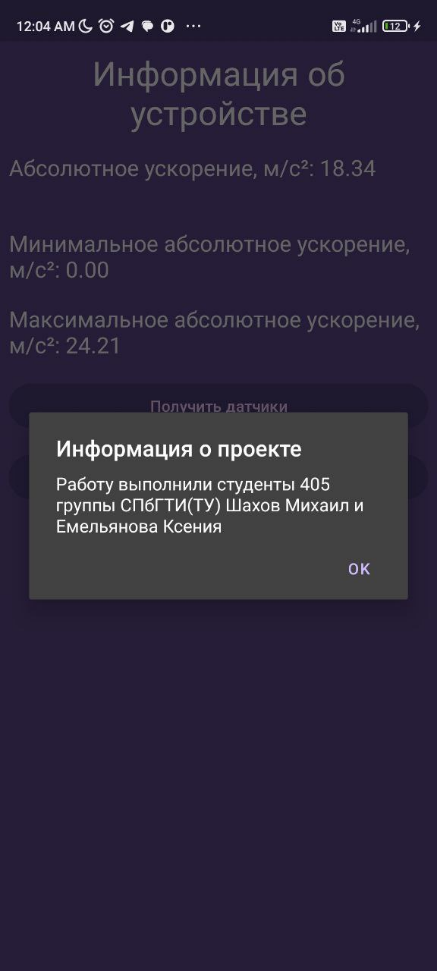


Рисунок 15 – Диалоговое окно мобильного приложения на светлой теме, портретная ориентация

**Выводы**

В ходе выполнения лабораторной работы было разработано мобильное приложения для Android для вывода абсолютного ускорения, с использованием датчика акселерометра. Для достижения поставленной цели были выполнены следующие задачи:

1) Изучен и описан аппаратный модуль (информация о датчиках, считывание данных с датчика акселерометра).

2) Разработано мобильное приложение для Android в среде Android Studio Giraffe 2022.3.1:

- создан и настроен новый проект приложения;

- указано наименование приложения и выбраны версии SDK: минимальная – 24, целевая – 31, а также версии ОС Android: минимальная – 7, целевая – 13;

- выбран язык программирования – Java.

- создана разметка макета графического интерфейса;

- создана управляющая логика приложения;

- проведено тестирование мобильного приложения. Тестирование показало работоспособность мобильного приложения в портретной и альбомной ориентации, а также с ветлой и темной теме.

Характеристики аппаратного обеспечения разработчика мобильного приложения приведены в таблице 4.

Таблица 4 – Характеристики аппаратного обеспечения

|  |  |
| --- | --- |
| Показатель | Значение |
| Тип ЭВМ | Ноутбук |
| Тип и тактовая частота (ν) процессора | Intel Core i7, ν = 2,71 ГГц |
| Состав и технические характеристики периферийных устройств ЭВМ | Мобильный телефон Xiaomi Redmi Note 9 Pro  Кабель USB - TYPE-C |
| Операционная система ЭВМ | Windows 10 |
| Версия Android Studio | Android Studio Giraffe 2022.3.1 |