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№1**

ТЕМА: Интерфейс интегрированной среды разработки Android Studio. Создание мобильного приложения для простых математических вычислений

Вариант № 10

Преподаватель Д. Н. Петров

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 | Входные данные | Результат |
| 10 | Вычисление времени нагрева жидкости, с от 24 °С до температуры кипения | Нагреваемая жидкость (5 шт. на выбор), объем нагреваемой жидкости, мощность нагревателя | Теплоемкость и температура кипения выбранной жидкости, время нагрева, с. до температуры кипения |

**Этапы выполнения**

Задачи:

1) Создание проекта мобильного приложения:

- выбор шаблона;

- указание наименования приложения и выбор версии SDK;

- выбор языка программирования.

2) Ознакомление со структурой проекта, инструментами разработчика и элементами интерфейса Android Studio.

3) Настройка проекта

4) В соответствии с вариантом задания:

- создание графического интерфейса;

- создание управляющей логики;

- тестирование мобильного приложения.

**Описание выполнения лабораторной работы**

1. **Создание проекта мобильного приложения**

В среде разработки мобильных приложений Android Studio Giraffe 17.0.6 для Android был создан проект Lab1. На рисунках 1-4 представлен процесс создания проекта.

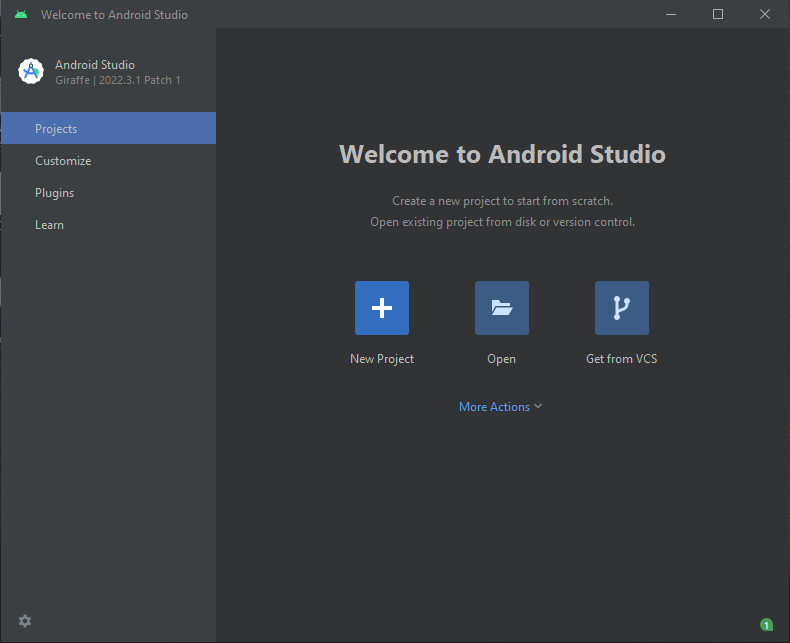


Рисунок 1 – Создание нового проекта

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

Изображение выглядит как текст, Шрифт, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

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).

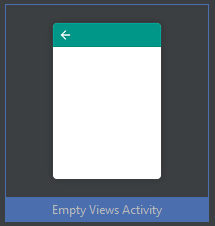
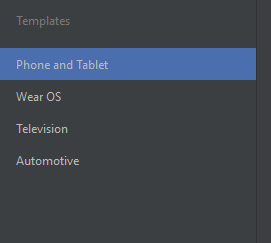


Рисунок 3 – Выбор типа целевого устройства и шаблона приложения

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

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 4 – Настройка нового приложения

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

1. **Ознакомление со структурой проекта, инструментами разработчика и элементами интерфейса Android Studio**

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.lab1 | MainActivity.java | Управляющий код выполнения вычисления |
| res/drawable | icon.xml | Логотип приложения |
| res/layout | activity\_main.xml | Главная активность приложения для вычисления времени нагрева, на основе введенных данных |
| res/values | colors.xml | Используемые цвета в приложении |
| colors.xml (night) | Используемые цвета в приложении для темной темы |
| 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 |

В таблице 3 представлено описание разделов «Project Structure» (рисунок 5)

Таблица 3 – Описание разделов Project Structure

|  |  |
| --- | --- |
| Раздел | Описание |
| Project | Информация о версии плагинов Gradle и Android Gradle |
| Modules | Информация о конфигурации сборки для данного модуля: версия SDK и инструментов сборки, используемые при компиляции, целевой и минимальный SDK, сертификат, который будет использоваться для подписи приложения, и зависимости библиотек |
| Dependencies | Список зависимостей библиотек, файлов и модулей для данного модуля |

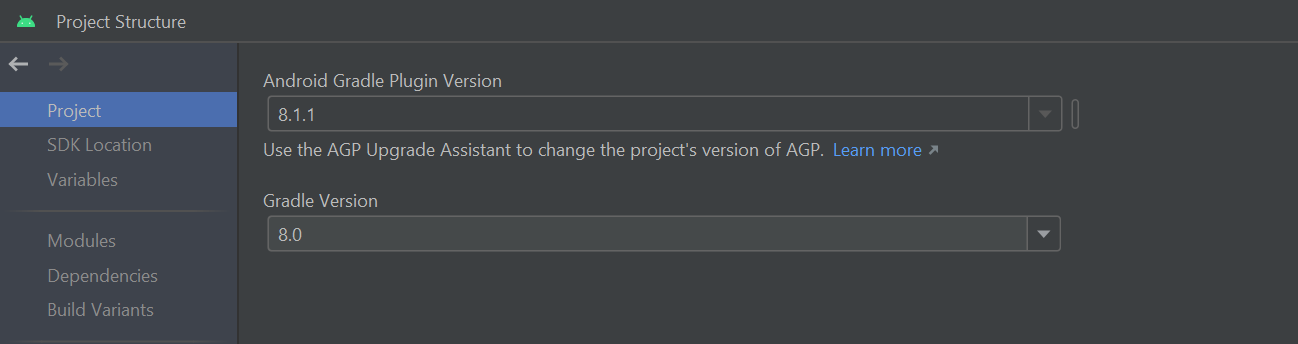


Рисунок 5 – Внешний вид раздела «Project Structure»

В процессе разработки могут быть использованы режимы Split, Code и Design, позволяющие на усмотрение разработчика вести работу только с кодом, только с интерфейсом или одновременно с кодом и интерфейсом приложения (рисунки 6-8).

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

Рисунок 6 – Внешний вид окна в режиме разработки «Split»

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 7 – Внешний вид окна в режиме разработки «Code»

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 8 – Внешний вид окна в режиме разработки «Design»

В разделе «Device Manager» (рисунок 9) находится информация о доступных физических и виртуальных устройствах, которые могут быть использованы для запуска и отладки приложения. У разработчика есть возможность добавления виртуальных устройств с необходимыми характеристиками.

В разделе «SDK Manager» (рисунок 10) находится информация об установленных SDK и инструментах, используемых средой разработки, а также URL ресурсов для обновления версий.

Изображение выглядит как текст, Мультимедийное программное обеспечение, программное обеспечение, линия

Автоматически созданное описание

Рисунок 9 – Внешний вид раздела «Device Manager»

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 10 – Внешний вид раздела «SDK Manager»

Для отладки использованы инструменты «LogCat», «Problems», и «Terminal».

Окно Logcat помогает производить отладку приложения, отображая журнал активности устройства в режиме реального времени – например, сообщения, которые пользователь добавил в приложение с помощью класса Log, сообщения от служб, запущенных на устройстве, или системные сообщения, например, при выполнении сборки мусора. Когда приложение генерирует исключение, Logcat показывает сообщение, за которым следует соответствующая трассировка стека, содержащая ссылки на строку кода.

Окно Problems содержит список всех ошибок и предупреждений, относящихся к данному проекту.

Окно Terminal предназначено для запуска команд к устройству.

1. **Настройка проекта**

При разработке приложения для запуска и отладки было использовано физическое устройство – мобильный телефон Xiamomi Redmi Note 9 Pro. Минимальная версия OC Android – 7, целевая – 13. Минимальная версия SDK – 24, целевая – 31.

1. **Выполнение задания в соответствии с вариантом**

**Создание графического интерфейса**

Листинг файла манифеста проекта 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">  
 <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="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MyApplication"  
 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>  
 </application>  
  
</manifest>

Листинг файла строковых констант strings.xml:

<resources>  
 <string name="app\_name">Лабораторная работа №1</string>  
 <string name="dialog\_title">Ошибка</string>  
 <string name="dialog\_message">Неверный формат данных</string>  
 <string name="water">Вода</string>  
 <string name="ethyl">Этиловый спирт</string>  
 <string name="oil">Нефть</string>  
 <string name="glycerol">Глицерин</string>  
 <string name="heatCapacity">Теплоемкость</string>  
 <string name="boilingTemperature">Температура кипения</string>  
 <string name="density">Плотность</string>  
 <string name="heatingTime">Время нагрева</string>  
 <string name="result">Результат</string>  
 <string name="benzene">Бензол</string>  
 <string name="Error\_Title">Ошибка</string>  
 <string name="Error\_Empty">Введите значения</string>  
 <string name="Error\_Empty\_Volume">Введите значение объема</string>  
 <string name="Error\_Empty\_Wattage">Введите значение мощности нагревательного элемента</string>  
 <string name="Label\_Volume">Объем жидкости, л</string>  
 <string name="Label\_HeaterWattage">Мощность\nнагревательного элемента, Вт</string>  
 <string name="Label\_ComputeTimeToHeat">Расчет времени нагрева жидкости</string>  
 <string name="Label\_ChooseLiquid">Выберите жидкость</string>  
 <string name="Button\_Calculate">Рассчитать</string>  
 <string name="TextInput\_Volume">Л</string>  
 <string name="TextInput\_PowerWattage">Вт</string>  
 <string name="Unit\_J\_K\_KG">Дж/(К\*кг)</string>  
 <string name="Unit\_0C">°C</string>  
 <string name="Unit\_S">с</string>  
</resources>

Листинг файла цветовых констант colors.xml:

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="textColorPrimary">#000000</color>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
 <color name="purple">#9334e6</color>  
 <color name="creamy">#F7E7CE</color>  
 <color name="grey">#1a1a1a</color>  
</resources>

Листинг файла цветовых констант colors.xml (night):

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="textColorPrimary">#FFFFFF</color>  
</resources>

Используемые изображения:

Логотипы приложения и оформления кнопок размещены в res/drawable (таблица 4)

Таблица 4 – 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="M15,13L15,5c0,-1.66 -1.34,-3 -3,-3S9,3.34 9,5v8c-1.21,0.91 -2,2.37 -2,4 0,2.76 2.24,5 5,5s5,-2.24 5,-5c0,- |  |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Код разметки | Графический вид | | |
| icon.xml | |
| 1.63 -0.79,-3.09 -2,-4zM11,5c0,-0.55 0.45,-1 1,-1s1,0.45 1,1h-1v1h1v2h-1v1h1v2h-2L11,5z"/> </vector> |  |

**Темы оформления:**

Листинг файла светлой темы themes.xml:

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

Листинг файла темной темы themes.xml (night):

<resources xmlns:tools="http://schemas.android.com/tools">

<!-- Base application theme. -->

<style name="Base.Theme.MyApplication" parent="Theme.Material3.DayNight.NoActionBar">

<item name="android:background">@color/grey</item>

<item name="android:textColor">@color/textColorPrimary</item>

</style>

</resources>

**Разметка макетов графических интерфейсов**

На рисунке 11 представлена разработанная структура макета активности приложения в режиме дизайна.

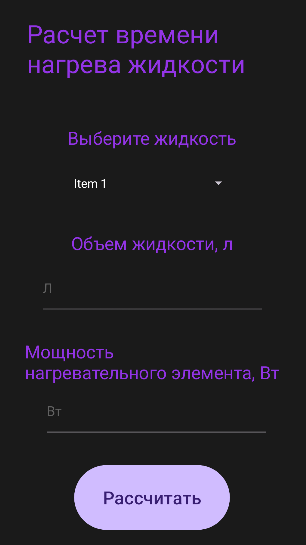


Рисунок 11 – Структура макета activity\_main

На макет графического интерфейса были добавлены следующие графические компоненты:

* ConstraintLayout – представляет контейнер, который позволяет создавать гибкие и масштабируемые визуальные интерфейсы (ConstraintLayout);
* textView – предназначен для отображения текста без возможности редактирования его пользователем (textViewHeader, textViewLqdTtl, textViewVTtl, textViewPWTtl);
* TextInput – предназначен для ввода текста пользователем (vTextInput, pWTextInput);
* Spinner – выпадающий список, позволяющий выбрать одно значение (spinnerLqdLst);
* Button – кнопка (calcBtn).

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"  
 tools:context=".MainActivity"  
 tools:layout\_editor\_absoluteX="0dp"  
 tools:layout\_editor\_absoluteY="-116dp">  
  
 <TextView  
 android:id="@+id/textViewHeader"  
 android:layout\_width="335dp"  
 android:layout\_height="82dp"  
 android:text="@string/Label\_ComputeTimeToHeat"  
 android:textColor="@color/purple"  
 android:textSize="34sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.501"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.04"  
 tools:ignore="TextSizeCheck" />  
  
 <TextView  
 android:id="@+id/textViewLqdTtl"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="64dp"  
 android:text="@string/Label\_ChooseLiquid"  
 android:textColor="@color/purple"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewHeader" />  
  
 <Spinner  
 android:id="@+id/spinnerLqdLst"  
 android:layout\_width="225dp"  
 android:layout\_height="72dp"  
 android:layout\_marginTop="8dp"  
 android:entries="@android:array/emailAddressTypes"  
 android:textColor="@color/creamy"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.501"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewLqdTtl" />  
  
 <TextView  
 android:id="@+id/textViewVTtl"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="28dp"  
 android:text="@string/Label\_Volume"  
 android:textColor="@color/purple"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/spinnerLqdLst" />  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/vTextInput"  
 android:layout\_width="300dp"  
 android:layout\_height="72dp"  
 android:layout\_marginTop="8dp"  
 android:hint="@string/TextInput\_Volume"  
 android:inputType="number|numberDecimal|numberSigned"  
 android:maxLines="1"  
 android:minHeight="48dp"  
 android:textColorHint="#616161"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.502"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewVTtl" />  
  
 <TextView  
 android:id="@+id/textViewPWTtl"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:text="@string/Label\_HeaterWattage"  
 android:textColor="@color/purple"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/vTextInput" />  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/pWTextInput"  
 android:layout\_width="300dp"  
 android:layout\_height="72dp"  
 android:layout\_marginStart="16dp"  
 android:hint="@string/TextInput\_PowerWattage"  
 android:inputType="number|numberDecimal|numberSigned"  
 android:maxLines="1"  
 android:minHeight="48dp"  
 android:textColorHint="#616161"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.477"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewPWTtl" />  
  
 <Button  
 android:id="@+id/calcBtn"  
 android:layout\_width="208dp"  
 android:layout\_height="94dp"  
 android:layout\_marginTop="32dp"  
 android:text="@string/Button\_Calculate"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/pWTextInput" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Создание управляющей логики**

Листинг кода MainActivity.java:

package com.example.myapplication;  
  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Spinner;  
import android.widget.TextView;  
  
import java.text.DecimalFormat;  
  
class Liquid {  
 double heatCapacity; //Дж/(кг\*K)  
 double boilingTemperature;  
 double density = 1;  
  
 Liquid(double heatCapacity, double boilingTemperature) {  
 this.heatCapacity = heatCapacity;  
 this.boilingTemperature = boilingTemperature;  
 }  
  
 Liquid(double heatCapacity, double boilingTemperature, double density) {  
 this.heatCapacity = heatCapacity;  
 this.boilingTemperature = boilingTemperature;  
 this.density = density;  
 }  
  
 private static double calculateTimeToBoilingPointTemplate(Liquid liquid, double volume, double startTemp, double endTemp, double heaterWattage) {  
 return ((liquid.heatCapacity \* volume \* liquid.density \* (endTemp - startTemp)) / (heaterWattage));  
 }  
  
 static double calculateTimeToBoilingPoint(Liquid liquid, double volume, double startTemp, double heaterWattage) {  
 return *calculateTimeToBoilingPointTemplate*(liquid, volume, startTemp, liquid.boilingTemperature, heaterWattage);  
 }  
  
 static double calculateTimeToBoilingPoint(Liquid liquid, double volume, double startTemp, double endTemp, double heaterWattage) {  
 return *calculateTimeToBoilingPointTemplate*(liquid, volume, startTemp, endTemp, heaterWattage);  
 }  
  
 static double calculateTimeToBoilingPoint(Liquid liquid, double volume, double heaterWattage) {  
 double startTemp = 24;  
 return *calculateTimeToBoilingPointTemplate*(liquid, volume, startTemp, liquid.boilingTemperature, heaterWattage);  
 }  
  
 double calculateTimeToBoilingPoint(double volume, double startTemp, double heaterWattage) {  
 return *calculateTimeToBoilingPointTemplate*(this, volume, startTemp, boilingTemperature, heaterWattage);  
 }  
  
  
}  
  
  
  
public class MainActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 // The TextView to show your Text  
 TextView dialogTextView = new TextView(this);  
 dialogTextView.setLineSpacing(55, 0);  
 dialogTextView.setPadding(70, 20, 70, 40);  
 dialogTextView.setTextIsSelectable(true);  
 AlertDialog.Builder builderDialog = new AlertDialog.Builder(this);  
 // Build the Dialog  
 AlertDialog dialog = builderDialog  
 .setView(dialogTextView)  
 .setCancelable(true)  
 .create();  
  
 AlertDialog dialog3 = new AlertDialog.Builder(this).create();  
 dialog3.setTitle(R.string.*Error\_Title*);  
  
  
 //создаем пять различных жидкостей  
 Liquid water = new Liquid(4186, 100);  
 Liquid ethyl = new Liquid(2470, 78.37, 0.789);  
 Liquid oil = new Liquid(2100, 140, 0.850);  
 Liquid glycerol = new Liquid(2430, 290, 1.252);  
 Liquid benzene = new Liquid(1050, 80.1, 0.876);  
  
 final Liquid[] liquid = new Liquid[1];  
 final double[] volume = {0};  
// double startTemp = 24;//C  
 final double[] heaterWattage = new double[1];  
  
 //добавляем надписи для спинера  
 String[] arraySpinner = new String[]{  
 getString(R.string.*water*),  
 getString(R.string.*ethyl*),  
 getString(R.string.*oil*),  
 getString(R.string.*glycerol*),  
 getString(R.string.*benzene*)  
 };  
  
 Spinner s = (Spinner) findViewById(R.id.*spinnerLqdLst*);  
 ArrayAdapter<String> adapter = new ArrayAdapter<>(this, R.layout.*color\_spinner\_layout*, arraySpinner);  
 adapter.setDropDownViewResource(androidx.constraintlayout.widget.R.layout.*support\_simple\_spinner\_dropdown\_item*);  
 s.setAdapter(adapter);  
  
  
 //button and calculation  
 Button calculateButton = (Button) findViewById(R.id.*calcBtn*);  
 calculateButton.setOnClickListener(v -> {  
 volume[0] = parseDoubleOrNull(((EditText) findViewById(R.id.*vTextInput*)).getText().toString());  
 heaterWattage[0] = parseDoubleOrNull(((EditText) findViewById(R.id.*pWTextInput*)).getText().toString());  
 dialog.setTitle(R.string.*Error\_Title*);  
 if (volume[0] <= 0 && heaterWattage[0] <= 0) {  
 dialogTextView.setText(R.string.*Error\_Empty*);  
 dialog.show();  
 return;  
 } else if (volume[0] <= 0) {  
 dialogTextView.setText(R.string.*Error\_Empty\_Volume*);  
 dialog.show();  
 return;  
 } else if (heaterWattage[0] <= 0) {  
 dialogTextView.setText(R.string.*Error\_Empty\_Wattage*);  
 dialog.show();  
 return;  
 }  
  
  
 if (s.getSelectedItem().toString().equals(getString(R.string.*water*)))  
 liquid[0] = water;  
 else if (s.getSelectedItem().toString().equals(getString(R.string.*ethyl*)))  
 liquid[0] = ethyl;  
 else if (s.getSelectedItem().toString().equals(getString(R.string.*oil*)))  
 liquid[0] = oil;  
 else if (s.getSelectedItem().toString().equals(getString(R.string.*glycerol*)))  
 liquid[0] = glycerol;  
 else if (s.getSelectedItem().toString().equals(getString(R.string.*benzene*)))  
 liquid[0] = benzene;  
  
  
 double timeToHeat = Liquid.*calculateTimeToBoilingPoint*(liquid[0], volume[0], 24, heaterWattage[0]);  
 String resultMsg = getString(R.string.*heatCapacity*) + ", " + getString(R.string.*Unit\_J\_K\_KG*) + ": " +  
 new DecimalFormat("0.00").format(liquid[0].heatCapacity) + "\n" +  
 getString(R.string.*boilingTemperature*) + ", " + getString(R.string.*Unit\_0C*) + ": " +  
 new DecimalFormat("0.00").format(liquid[0].boilingTemperature) + "\n" +  
 getString(R.string.*heatingTime*) + ", " + getString(R.string.*Unit\_S*) + ": " +  
 new DecimalFormat("0.00").format(timeToHeat);  
  
 dialogTextView.setText(resultMsg);  
 dialog.setTitle(R.string.*result*);  
 dialog.show();  
 });  
 }  
  
 public double parseDoubleOrNull(String value) {  
 try {  
 return Double.*parseDouble*(value);  
 } catch (NumberFormatException e) {  
 return 0;  
 }  
 }  
}

**Тестирование мобильного приложения**

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

Результат работы мобильного приложения представлен на рисунках 12-18.

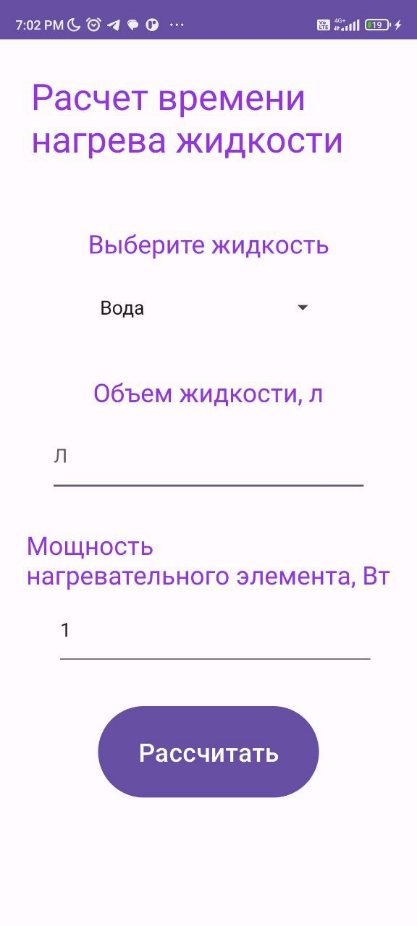
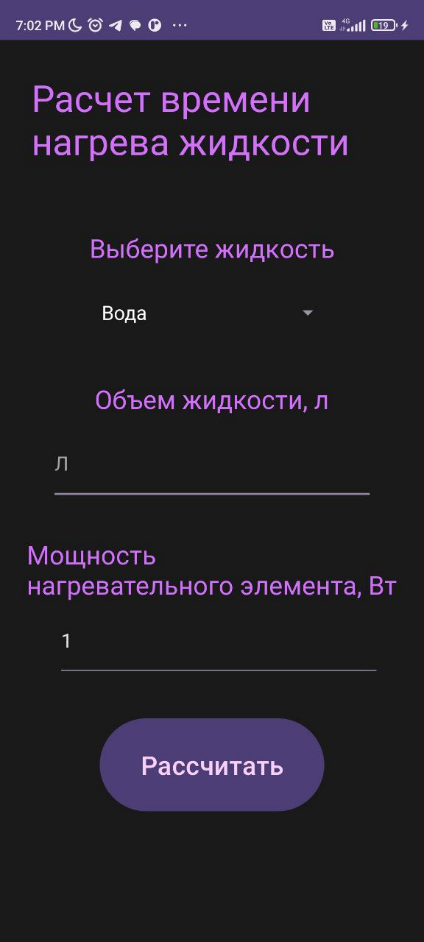


Рисунок 12 – Результат работы главной активности мобильного приложения на светлой теме, портретная ориентация

Рисунок 13 – Результат работы главной активности мобильного приложения на темной теме, портретная ориентация

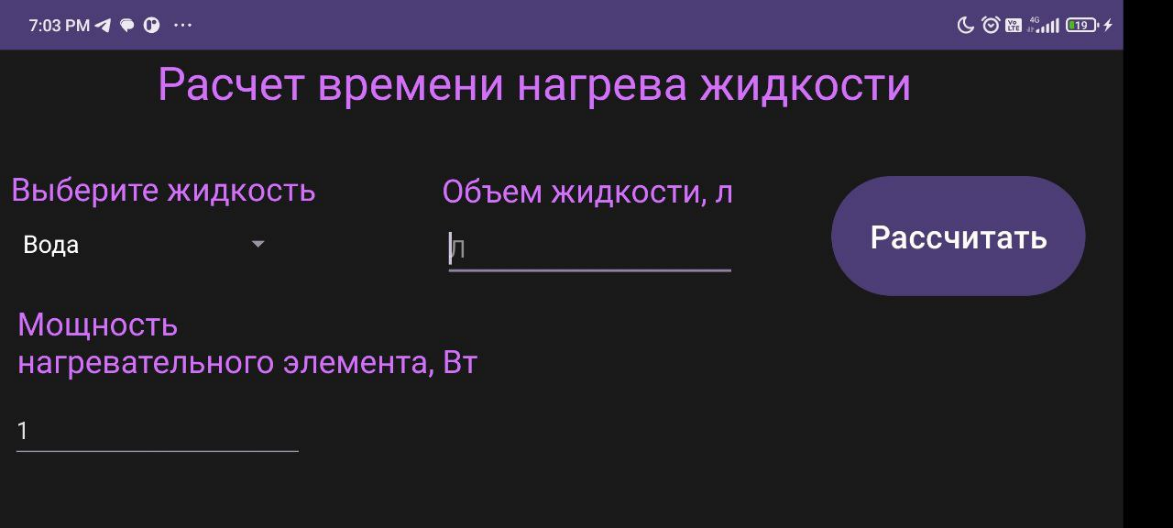


Рисунок 14 – Результат работы главной активности мобильного приложения на темной теме, альбомная ориентация

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 15 – Диалоговое окно о незаполненных полях ввода мобильного приложения на темной теме, портретная ориентация

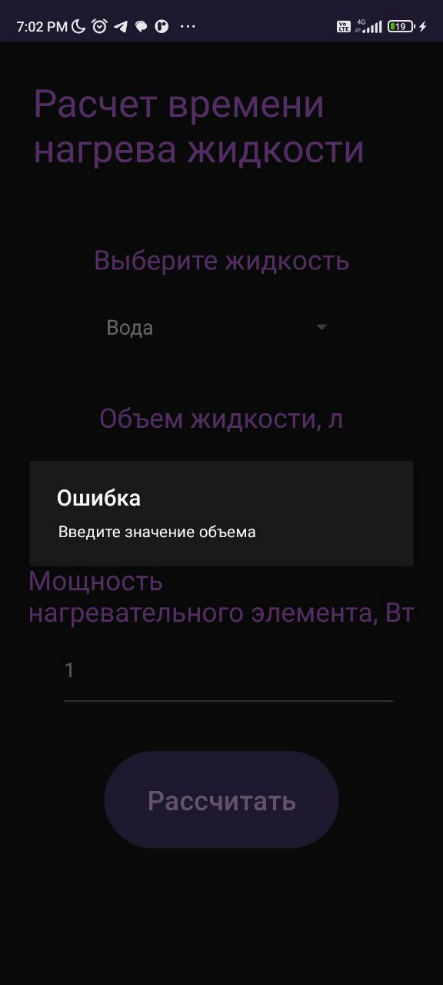


Рисунок 16 – Диалоговое окно о незаполненном поле ввода значения объема мобильного приложения на темной теме, портретная ориентация

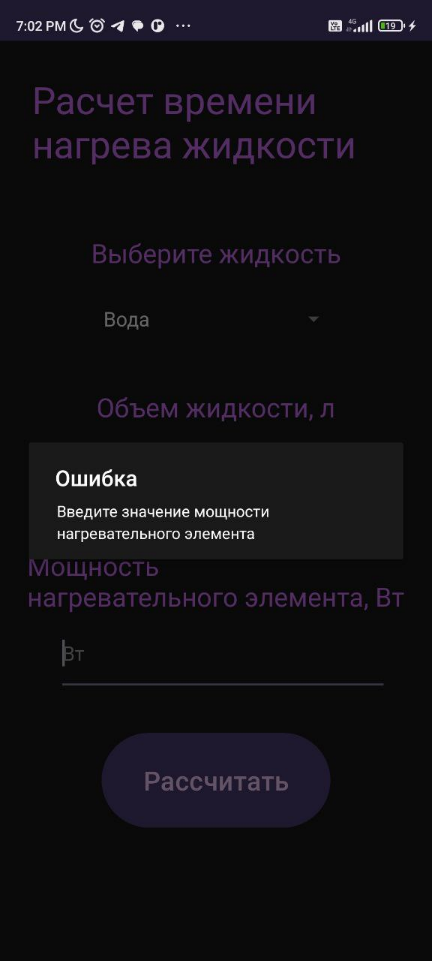


Рисунок 17 – Диалоговое окно о незаполненном поле ввода значения мощности нагревательного элемента мобильного приложения на темной теме, портретная ориентация

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 18 – Результат вычисления времени нагрева жидкости

**Выводы**

В ходе выполнения лабораторной работы разработано мобильное приложение для Android для вычисления времени нагрева жидкости, с от 24 °С до температуры кипения. Для достижения поставленной цели были выполнены следующие задачи:

1) Создан проект мобильного приложения для Android в среде Android Studio Giraffe 2022.3.1:

- выбран шаблон;

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

- выбран язык программирования – Java.

2) Изучена структура проекта, инструменты разработчика и элементы интерфейса Android Studio.

3) Настроен проект

4) В ходе разработки мобильного приложения для расчета константы скорости химической реакции:

- создан графический интерфейс;

- создана управляющая логика приложения;

- проведено тестирование мобильного приложения. Тестирование показало работоспособность мобильного приложения в портретной и альбомной ориентации, а также в светлой и темной темах.

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-A – USB-C |
| Операционная система ЭВМ | Windows 10 |
| Версия Android Studio | Android Studio Giraffe 2022.3.1 |