

МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖНОЙ ПОЛИТИКИ НИЖЕГОРОДСКОЙ ОБЛАСТИ  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
«НИЖЕГОРОДСКИЙ РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

- ПМ. 01 Разработка модулей программного обеспечения для компьютерных систем

Обучающегося

Мамонов Антон Дмитриевич

(Фамилия Имя Отчество)

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Специалист по тестированию в области информационных технологий

Курс: 3

Группа: ЗИСиП-19-1

Руководитель практики от ГБПОУ «НРТК» Преподаватель  
(Должность)

Алексеева Зинаида Петровна  
(Фамилия Имя Отчество)

Нижний Новгород  
2022 г.

## Содержание

ВВЕДЕНИЕ.....	3
ПМ.01 Разработка модулей программного обеспечения для компьютерных систем.....	4
Выполнение индивидуального задания .....	5
1. Краткая история создания предприятия, основные сведения о предприятии, выпускаемой продукции (оказании услуг), организация рабочего места и условий труда. ....	5
2. Описание технологии меню входа в Android Studio на Java. ....	5
3. Разработка мобильного приложения «Умный дом — управление вытяжкой». ....	6
Главная страница приложения .....	7
Страница управления кулерами отдельно .....	10
ЗАКЛЮЧЕНИЕ.....	12

## **ВВЕДЕНИЕ**

По окончании производственной практики у меня должны быть сформированы следующие профессиональные компетенции:

- Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием;
- Разрабатывать программные модули в соответствии с техническим заданием.
- Выполнять отладку программных модулей с использованием специализированных программных средств;
- Выполнять тестирование программных модулей;
- Осуществлять рефакторинг и оптимизацию программного кода;
- Разрабатывать модули программного обеспечения для мобильных платформ.

А также, следующие общие компетенции:

- Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.;
- Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;
- Планирование и реализация собственного профессионального и личностного развития;
- Работа в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами;
- Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста;
- Проявление гражданско-патриотической позиции, умение демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей;
- Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях;
- Оказание содействия сохранению окружающей среды, ресурсосбережению, умение эффективно действовать в чрезвычайных ситуациях;
- Использование информационных технологий в профессиональной деятельности;
- Пользоваться профессиональной документацией на государственном и иностранном языке;
- Планирование предпринимательской деятельности в профессиональной сфере.

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
«НИЖЕГОРОДСКИЙ РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

**Согласовано**

Заведующий практикой

\_\_\_\_\_ Л.Д. Бабушкина

«01» июня 2022 г.

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

Обучающийся Мамонов Антон Дмитриевич

(Фамилия Имя Отчество)

Группа ЗИСИП-19-1

на период производственной практики

- ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

Вопросы:

1. Краткая история создания предприятия, основные сведения о предприятии, выпускаемой продукции (оказании услуг), организация рабочего места и условий труда.
2. Описание технологии меню входа в Android Studio на Java.
3. Разработка мобильного приложения «Умный дом — управление вытяжкой».
  - 3.1 Создать макет.
  - 3.2 Реализовать работу с bluetooth.
  - 3.3 Возможность управления куллерами.

Дата выдачи «01» июня 2022 г.

Срок исполнения «28» июня 2022 г.

Руководитель практики от ГБПОУ «НРТК»

Алексеева З.П.

(Фамилия Имя Отчество)

Руководитель практики от предприятия

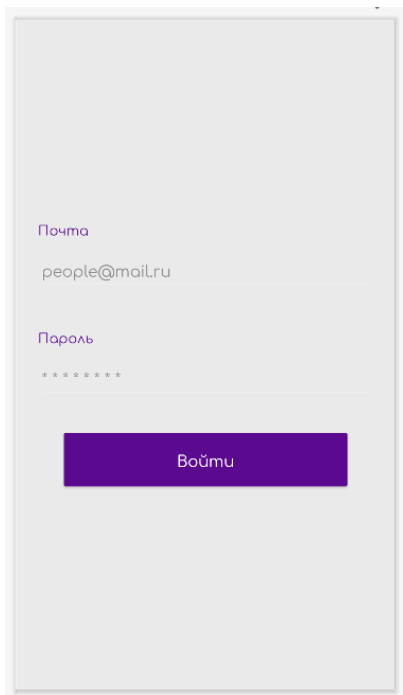
\_\_\_\_\_  
(Фамилия Имя Отчество)

## Выполнение индивидуального задания

### 1. Краткая история создания предприятия, основные сведения о предприятии, выпускаемой продукции (оказании услуг), организация рабочего места и условий труда.

Государственное бюджетное профессиональное образовательное учреждение «Нижегородский радиотехнический колледж» (ГБПОУ «НРТК»). Учреждение образовано в соответствии с постановлением Президиума Нижегородского Крайисполкома от 16 июня 1930 года № 33 как Нижегородский электротехникум. Выпускники колледжа востребованы на рынке труда, что обусловлено тесным сотрудничеством с ведущими предприятиями региона. В рамках стратегического партнерства в колледже функционирует Координационный совет.

### 2. Описание технологии меню входа в Android Studio на Java.



Для реализации страницы авторизации необходимо объявить два поля для ввода и кнопку «Войти».

В поле «Почта» нужно сделать проверку на корректность введенных данных, пароль должен быть скрыт.

При нажатии на кнопку «Войти» на сервер отправляется POST запрос с данными для авторизации, в зависимости от ответа сервера будет выведено сообщение «Не удалось авторизовать» или же вход будет успешно осуществлен.

#### Алгоритм отправки запроса на сервер и получение ответа

```
public void tryLogin() {
    String mail = email.getText().toString();
    String pass = password.getText().toString();

    JSONObject jsonObject = new JSONObject();

    try {
        jsonObject.put( name: "email", mail);
        jsonObject.put( name: "password", pass);
    } catch (JSONException e) {
        e.printStackTrace();
    }

    String url = "https://food.madskill.ru/auth/login";
    RequestQueue requestQueue = Volley.newRequestQueue( context: this);

    JSONObjectRequest request = new JSONObjectRequest(Request.Method.POST, url, jsonObject,
        response -> {
            Intent intent = new Intent( packageContext: this, list.class);
            LoginActivity.this.startActivity(intent);
            LoginActivity.this.finish();
        },
        error -> createDialog( activity: this, msg: "Неверная почта или пароль"));
    requestQueue.add(request);
}
```

### 3. Разработка мобильного приложения «Умный дом — управление вытяжкой».

Приложение разрабатывалось для умного дома в условиях комнаты. Первостепенная задача – взаимодействие с самодельной вентиляцией в окне.

Вентиляция включает в себя следующие части:

- 3 кулера
- Управляющая плата
- Bluetooth модуль
- Датчик температуры и влажности

Требуемый функционал:

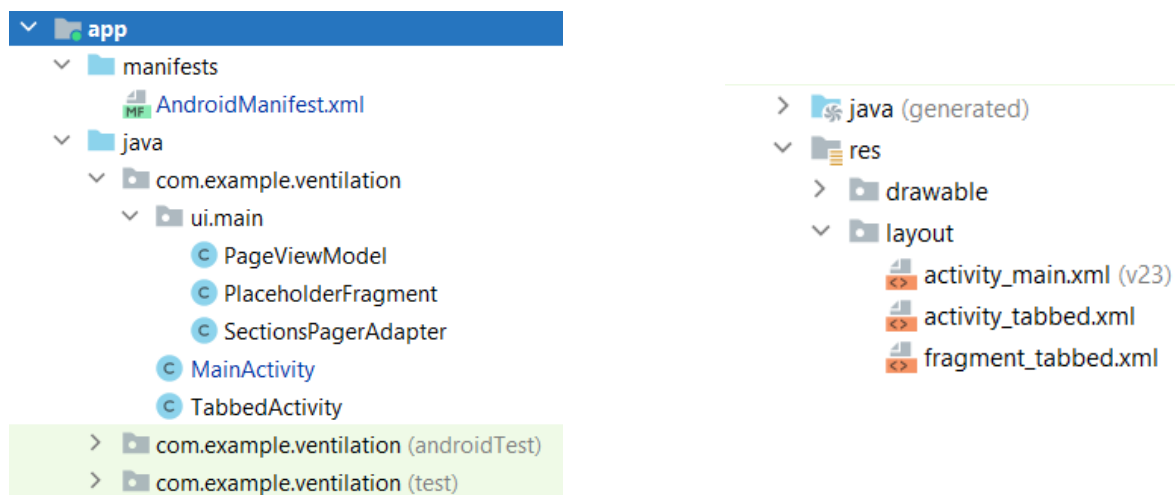
- Работа с Bluetooth – подключение / отключение устройства
- Управление кулерам – всеми / по отдельности
- Считывание значений температуры / влажности

Для работы приложению требуется выдать права на работу с модулем Bluetooth

```
<uses-permission android:name="android.permission.BLUETOOTH"
    android:maxSdkVersion="30" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"
    android:maxSdkVersion="30" />

<uses-permission android:name="android.permission.BLUETOOTH_SCAN"
    android:usesPermissionFlags="neverForLocation" />
<uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE" />
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
```

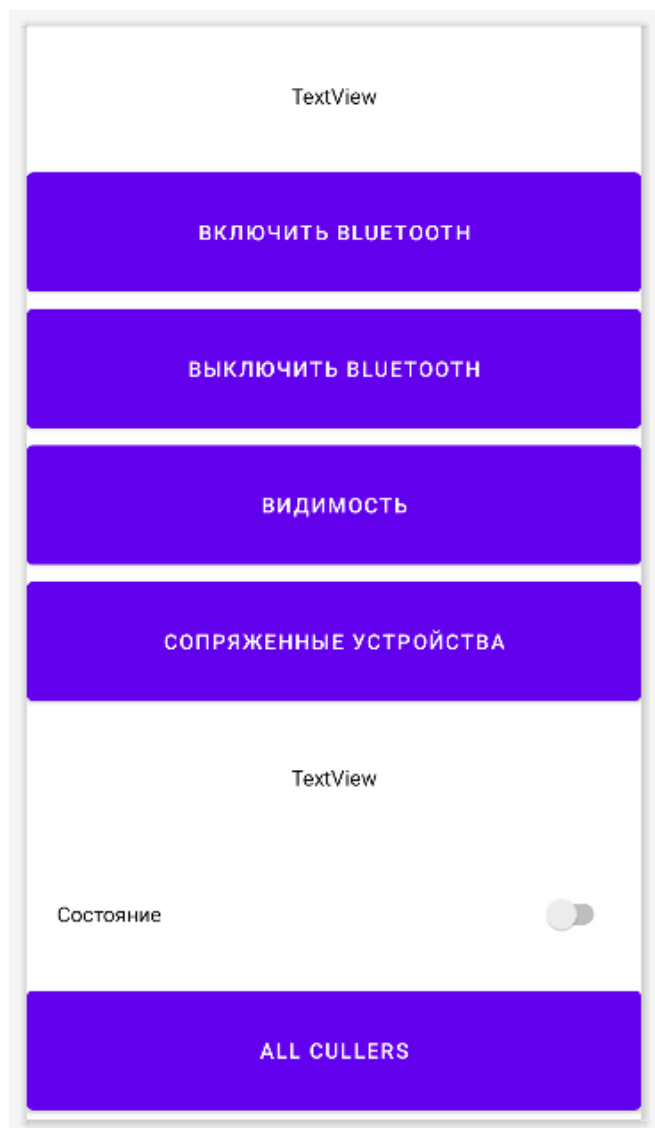
Структура проекта



## Главная страница приложения

Элементы главного экрана сверху вниз:

- Отображение температуры / влажности воздуха – текстовое поле
- Включение Bluetooth - кнопка
- Выключение Bluetooth - кнопка
- Видимость устройства - кнопка
- Список устройств - кнопка
- Текущее подключенное устройство – текстовое поле
- Состояние – вкл / выкл все кулеры
- Управление кулерами по-отдельности - кнопка



```
<Button
    android:id="@+id/onBtn"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="Включить bluetooth" />
```

```
<Button
    android:id="@+id/offBtn"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="выключить bluetooth" />
```

```
<Button
    android:id="@+id/discoverableBtn"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="видимость" />
```

```
<Button
    android:id="@+id/pairedBtn"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="сопряженные устройства" />
```

При создании приложения сперва объявляются все элементы главного экрана и идет проверка на подключение Bluetooth, получение данных температуры и влажности

```
@SuppressWarnings("SetTextI18n")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    @SuppressWarnings("UseSwitchCompatOrMaterialCode") Switch switchMsg = findViewById(R.id.switch2);
    statusTemperatureDevice = findViewById(R.id.statusBluetooth);
    pairedDevice = findViewById(R.id.pairedDevice);
    onBtn = findViewById(R.id.onBtn);
    offBtn = findViewById(R.id.offBtn);
    discoverBtn = findViewById(R.id.discoverableBtn);
    pairedBtn = findViewById(R.id.pairedBtn);

    bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

    if (bluetoothAdapter == null) {
        statusTemperatureDevice.setText("Температура");
    } else {
        statusTemperatureDevice.setText("Влажность");
    }
}
```

При нажатии на кнопку включения Bluetooth

```
onBtn.setOnClickListener(new View.OnClickListener() {
    @SuppressWarnings("MissingPermission")
    @Override
    public void onClick(View view) {
        if (!bluetoothAdapter.isEnabled()) {
            showToast( msg: "Включение bluetooth");
            Intent intent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(intent, REQUEST_ENABLE_BT);
        } else {
            showToast( msg: "Bluetooth уже включен");
        }
    }
});
```

При нажатии на кнопку выключения Bluetooth

```
offBtn.setOnClickListener(new View.OnClickListener() {
    @SuppressWarnings("MissingPermission")
    @Override
    public void onClick(View view) {
        if (bluetoothAdapter.isEnabled()) {
            bluetoothAdapter.disable();
            showToast( msg: "Выключаем Bluetooth");
        } else {
            showToast( msg: "Bluetooth уже выключен");
        }
    }
});
```



## Включение видимости устройства

```
discoverBtn.setOnClickListener(new View.OnClickListener() {
    @SuppressWarnings("MissingPermission")
    @Override
    public void onClick(View view) {
        if (!bluetoothAdapter.isDiscovering()) {
            showToast( msg: "Делаем устройство видимым");
            Intent intent = new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
            startActivityForResult(intent, REQUEST_DISCOVER_BT);
        }
    }
});
```

## Просмотр списка устройств

```
pairedBtn.setOnClickListener(new View.OnClickListener() {
    @SuppressWarnings("MissingPermission")
    @Override
    public void onClick(View view) {
        if (bluetoothAdapter.isEnabled()) {
            pairedDevice.setText("Подключенные устройства");
            Set<BluetoothDevice> devices = bluetoothAdapter.getBondedDevices();
            for (BluetoothDevice device : devices) {
                pairedDevice.append("\nУстройство " + device.getName() + ", " + device);
            }
        } else {
            showToast( msg: "Сперва нужно включить bluetooth");
        }
    }
});
```

## Включение / выключение всех кулеров

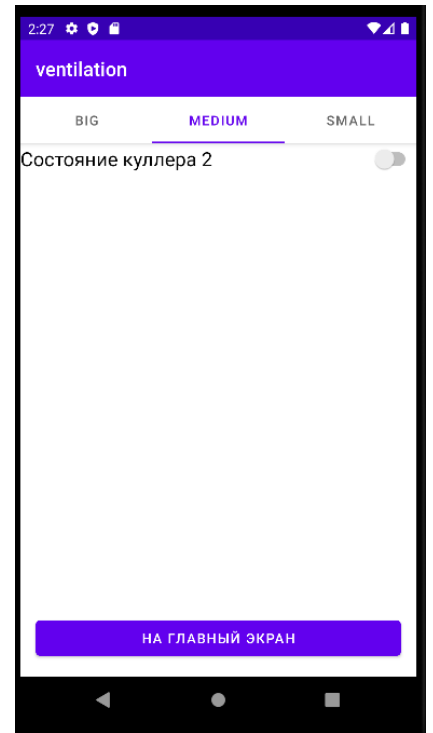
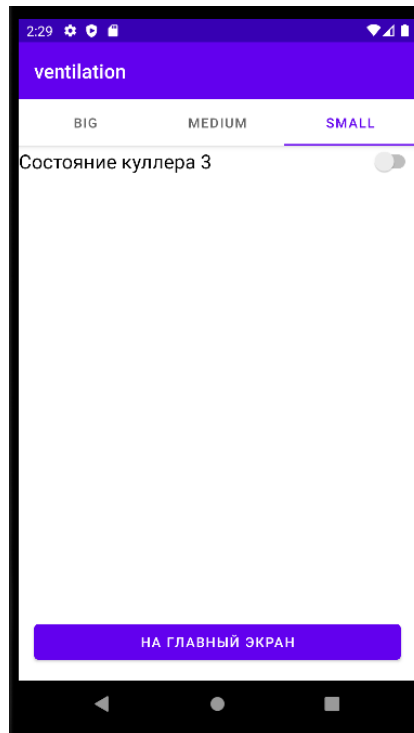
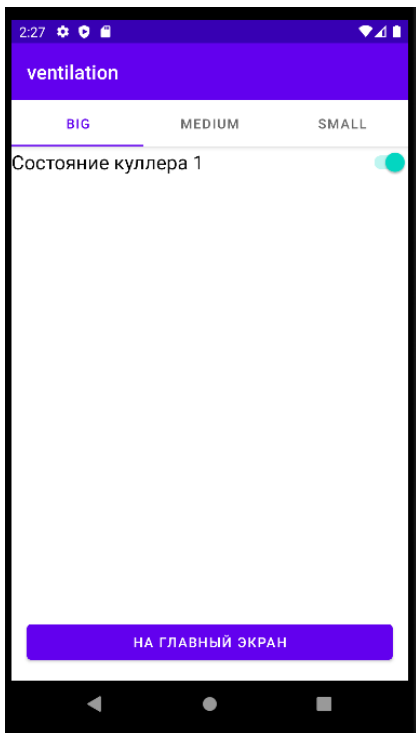
```
class MyOnCheckedChangeListener implements CompoundButton.OnCheckedChangeListener {
    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
        String str = "";
        if (b) {
            str = "Включено";
        } else {
            str = "Выключено";
        }
        Toast.makeText( context: MainActivity.this, str, Toast.LENGTH_SHORT).show();
    }
}
```

## Переход на страницу для управления кулерами отдельно

```
public void all(View view) {
    Intent intent = new Intent( packageContext: this, TabbedActivity.class);
    MainActivity.this.startActivity(intent);
    MainActivity.this.finish();
}
```

## Страница управления кулерами отдельно

На экране смартфона данная страница имеет три вкладки – три кулера – большой, средний и маленький. Каждый из них мы можем включить и выключить.

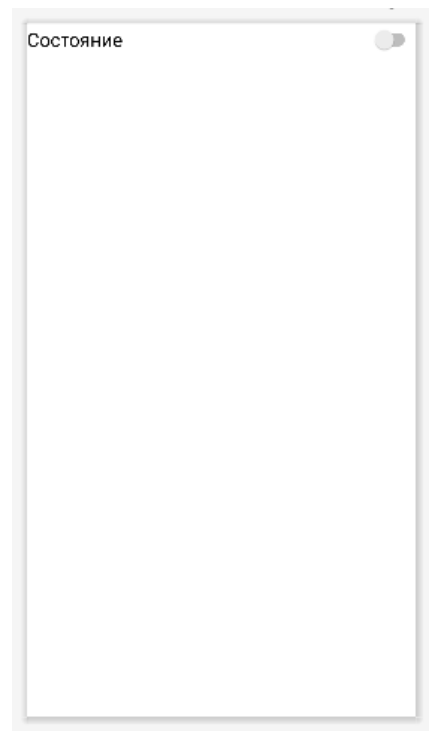


Данная страница формируется из двух макетов

Первый – постоянные элементы, которые видны постоянно на странице



Второй – элементы для каждого кулера, в данном случае индикатор состояния



## Обработка изменения состояния кулера

```
public class PageViewModel extends ViewModel {  
  
    private MutableLiveData<Integer> mIndex = new MutableLiveData<>();  
    private LiveData<String> mText = Transformations.map(mIndex, new Function<Integer, String>() {  
        @Override  
        public String apply(Integer input) { return "Состояние куллера " + input; }  
    });  
  
    public void setIndex(int index) { mIndex.setValue(index); }  
  
    public LiveData<String> getText() { return mText; }  
}
```

## Задание количества кулеров – в данном случае 3 кулера (Big, Medium, Small)

```
@StringRes  
private static final int[] TAB_TITLES = new int[]{"Big", "Medium", "Small"};  
private final Context mContext;  
  
public SectionsPagerAdapter(Context context, FragmentManager fm) {  
    super(fm);  
    mContext = context;  
}
```

## Создание страницы с кулерами, с нужным количеством элементов (3)

```
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    pageViewModel = new ViewModelProvider( owner: this).get(PageViewModel.class);  
    int index = 1;  
    if (getArguments() != null) {  
        index = getArguments().getInt(ARG_SECTION_NUMBER);  
    }  
    pageViewModel.setIndex(index);  
}  
  
@Override  
public View onCreateView(  
    @NonNull LayoutInflater inflater, ViewGroup container,  
    Bundle savedInstanceState) {  
  
    binding = FragmentTabbedBinding.inflate(inflater, container, attachToParent: false);  
    View root = binding.getRoot();  
  
    final TextView textView = binding.switch1;  
    pageViewModel.getText().observe(getViewLifecycleOwner(), new Observer<String>() {  
        @Override  
        public void onChanged(@Nullable String s) { textView.setText(s); }  
    });  
    return root;  
}
```

## ЗАКЛЮЧЕНИЕ

За время практики я повысил качество своих знаний. Ознакомился с предприятием, рабочим местом, охраной труда и инструктажем по технике безопасности, выполнением норм и правилами безопасности, а также изучением предметной области для создания базы данных предприятия.

Научился:

- формировать алгоритмы разработки программных модулей в соответствии с техническим заданием;
- разрабатывать программные модули в соответствии с техническим заданием;
- выполнять отладку программных модулей с использованием специализированных программных средств;
- выполнять тестирование программных модулей;
- осуществлять рефакторинг и оптимизацию программного кода;
- разрабатывать модули программного обеспечения для мобильных платформ.

Данная практика – это опыт, который обязательно должен потребоваться в моей дальнейшей профессиональной деятельности.

Обучающийся \_\_\_\_\_

за производственную практику заслуживает оценку \_\_\_\_\_

Руководитель практики от ГБПОУ «НРТК» \_\_\_\_\_

(Подпись)

(Фамилия Имя Отчество)

«\_\_» \_\_\_\_\_ 2022 г.