

**НПОУ «ЯКУТСКИЙ КОЛЛЕДЖ ИННОВАЦИОННЫХ ТЕХНОЛОГИЙ»**

(наименование образовательной организации)

**ОТДЕЛЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

(наименование структурного подразделения (кафедра / отделение))

**ОТЧЕТ**

**по учебной практике**

**ПМ.01.03 Разработка модулей программного обеспечения для компьютерных систем**

(наименование типа практики)

Выполнил:

обучающийся КИСП-23 группы  
Оконешников Родион Николаевич

(фамилия, имя, отчество (при наличии))

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 20\_\_\_ г.  
(личная подпись, дата)

Проверил руководитель практики  
от образовательной организации:  
преподаватель

(уч. степень, уч. звание, должность)

Цвикальский Андрей  
Константинович

(фамилия, имя, отчество (при наличии))

Оценка

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 20\_\_\_ г.  
(личная подпись, дата)

Рег. № \_\_\_\_\_ от «\_\_\_» \_\_\_\_\_ 20\_\_\_ г.

Якутск, 2025

---

**НПОУ «Якутский колледж инновационных технологий»**

---

(наименование образовательной организации)

---

**Отделение Информационных технологий**

---

(наименование структурного подразделения (кафедра / отделение))

УТВЕРЖДАЮ  
заведующий отделением

\_\_\_\_\_ / И.В. Пронин/  
« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

## **ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ**

### **Общие сведения**

ФИО обучающегося	Оконешников Родион Николаевич
Курс	2
Форма обучения	Очная
Направление подготовки / специальность / профессия	09.02.07 Информационные системы и программирование
Наименование структурного подразделения (кафедра / отделение)	Отделение информационных технологий
Группа	КИСП-23
Вид практики	Учебная
Тип практики	По модулю
Способ проведения практики	Стационарно
Форма проведения практики	Дискретно
Место прохождения практики	НПОУ «ЯКИТ»
Период прохождения практики	с 19.05.25 по 25.05.25
Реквизиты договора о прохождении практики (при проведении практики в профильной организации)	

### **Содержание индивидуального задания**

Разработать программу «Калькулятор» для телефона android со следующим функционалом:

- Основные арифметические операции (сложение, вычитание, умножение и деление);
- Операции с памятью (очистка, добавление, вычитание, выгрузка для операций);
- Квадратный корень;
- Процент от числа;
- Значение  $1/x$ ;
- Удаление последнего введенного символа с рабочего экрана;
- Очистка рабочего экрана.

Программу разработать средствами “Android Studio”, используя язык Java.

---

Задание на практику составил:  
руководитель практики от образовательной организации

<hr/>	<hr/>	<hr/>	<hr/>
(уч. степень, уч. звание, должность)	(подпись)	А.К. Цвикальский (И.О. Фамилия)	«19» мая 2025 г. (дата)
Задание на практику принял: обучающийся	<hr/>	<hr/>	<hr/>
	(подпись)	Р.Н. Оконешников (И.О. Фамилия)	«19» мая 2025 г. (дата)

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1. ТЕОРИТИЧЕСКАЯ ЧАСТЬ .....	7
1.1. Android Studio .....	7
1.2. Язык Java .....	7
1.3. Калькулятор .....	8
2. ПРАКТИЧЕСКАЯ ЧАСТЬ .....	9
2.1. Написание кода .....	9
2.2. Написание frond-end .....	19

## ВВЕДЕНИЕ

В процессе учебной практики была разработана Android-программа "Калькулятор". Это приложение предоставляет пользователям возможность выполнять базовые и расширенные математические операции, а также поддерживает работу с памятью и другие полезные функции.

### Актуальность проекта

Создание калькулятора является важным этапом в обучении разработке мобильных приложений. Это приложение не только демонстрирует базовые навыки программирования на языке Java, но и позволяет углубиться в такие аспекты, как работа с пользовательским интерфейсом (UI), обработка данных, сохранение и восстановление состояния приложения, а также обеспечение его совместимости с различными версиями операционной системы Android.

### Цели и задачи проекта

#### Основные цели работы:

- Освоение основ разработки Android-приложений:
- Изучение структуры Android-приложений.
- Понимание жизненного цикла Activity и Fragment.
- Работа с основными компонентами Android SDK.
- Изучение принципов работы с пользовательским интерфейсом:
- Создание и настройка пользовательских интерфейсов с использованием XML-разметки.
- Программирование интерактивных элементов, таких как кнопки, текстовые поля, меню и других.
- Разработка адаптивного интерфейса, который будет корректно отображаться на различных устройствах.

### Методология разработки

Проект был реализован с использованием следующих методологий и инструментов:

- Android Studio: Интегральная среда разработки (IDE) для создания Android-приложений.
- Java: Основной язык программирования для разработки Android-приложений.
- XML: Язык разметки для описания пользовательского интерфейса.
- Git: Система контроля версий для управления исходным кодом проекта.

## 1. ТЕОРИТИЧЕСКАЯ ЧАСТЬ

### 1.1. Android Studio

Это программа для создания приложений для смартфонов на Android. Она была впервые представлена в 2013 году. В этой программе можно создавать приложения для самых разных устройств, таких как умные часы и телевизоры на Android.

Android Studio можно скачать бесплатно. Сначала она была в стадии тестирования, но потом стала стабильной. В 2017 году в этой программе появилась возможность писать код на языке Kotlin, который теперь тоже официально используется для создания приложений на Android, вместе с языками Java и C++.

Что умеет Android Studio?

- Простота в использовании: программа предлагает шаблоны для быстрого создания стандартных элементов приложения.
- Удобный редактор: можно легко перетаскивать элементы интерфейса, видеть, как они будут выглядеть на разных экранах, и быстро исправлять ошибки в коде.
- Инструменты для сборки: Android Studio автоматически собирает и тестирует приложение.
- Поддержка Google: программа легко работает с сервисами Google, такими как отправка уведомлений и создание приложений для Google Cloud.
- Обновление для новых версий Android: с помощью Android Studio можно создавать приложения для последних версий Android.

Ограничения:

В Android Studio нельзя полностью отключить автосохранение файлов. Разработчики считают, что это помогает избежать потери важных данных.

Почему стоит использовать Android Studio?

С каждым обновлением программа становится лучше и предлагает новые возможности. Например, в последней версии можно использовать современные функции языка программирования Java и быстро обновлять приложения без полной их переделки.

Android Studio работает на компьютерах с операционными системами Windows, macOS и Linux. Она подходит как для начинающих, так и для опытных разработчиков.

## 1.2. Язык Java

Java — это язык программирования, который используется для создания различных приложений. Он был создан компанией Sun Microsystems, а сейчас его поддерживает компания Oracle.

Программы на Java сначала переводятся в специальный код, который может работать на любом компьютере, где есть специальная программа, называемая виртуальной машиной Java (JVM). Это позволяет запускать Java-программы на разных устройствах, даже если у них разное "железо".

У Java есть система безопасности, которая следит за тем, чтобы программы не делали ничего опасного, например, не пытались украсть данные или подключиться к чужим компьютерам.

Раньше Java-программы могли встраиваться в веб-страницы, но сейчас это используется реже. Вместо этого на Java пишут серверные приложения и клиентские программы.

Одной из особенностей Java является то, что она транслирует код в машинный код прямо во время работы программы. Это помогает ускорить её выполнение. Также в стандартных библиотеках Java много кода, который написан для работы с конкретным "железом", что тоже ускоряет работу программ.

Java очень популярен и занимает высокие места в рейтингах языков программирования.



## 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

### 2.1. Написание кода

```
private EditText display; 8 usages

private String vvod = ""; 70 usages
private double memoryValue = 0; 6 usages
private String currentOperator = ""; 11 usages
private double firstOperand = 0; 9 usages
private boolean isNewInput = true; 34 usages
```

Рисунок 1. Объявление переменных

display - текстовое поле для отображения вводимых чисел и результатов вычислений.

vvod - строка, которая накапливает вводимые пользователем цифры и символы перед отображением на экране.

memoryValue - числовая переменная для хранения значения в памяти калькулятора между операциями.

currentOperator - строка, сохраняющая последнюю выбранную математическую операцию (сложение, вычитание, умножение или деление).

firstOperand - числовая переменная для хранения первого операнда при выполнении операций между двумя числами.

isNewInput - логический флаг, определяющий, начинается ли новый ввод (true) или продолжается текущий (false).

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

    display = findViewById(R.id.display);
}
```

Рисунок 2. Метод onCreate() - инициализация приложения

Функции:

Загрузка макета activity\_main.xml

Получение ссылки на поле вывода

## Инициализация обработчиков кнопок

```
Button button0 = findViewById(R.id.btn_0);
Button button1 = findViewById(R.id.btn_1);
Button button2 = findViewById(R.id.btn_2);
Button button3 = findViewById(R.id.btn_3);
Button button4 = findViewById(R.id.btn_4);
Button button5 = findViewById(R.id.btn_5);
Button button6 = findViewById(R.id.btn_6);
Button button7 = findViewById(R.id.btn_7);
Button button8 = findViewById(R.id.btn_8);
Button button9 = findViewById(R.id.btn_9);
```

Рисунок 3. Определение кнопок

```
button0.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isNewInput == true) {
            vvod = "0";
            isNewInput = false;
        } else {
            vvod = vvod + "0";
        }
        updateDisplay();
    }
});
```

Рисунок 4. Обработчики цифровых кнопок

Что происходит, когда ты нажимаешь кнопку "0":

1. Проверка состояния ввода:

- isNewInput — это переменная, которая показывает, начинаем ли мы вводить новое число.

- Если true (например, после нажатия оператора или очистки), значит, нужно начать ввод нового числа.

2. Обработка нового ввода:

vvod = "0"; // Заменяем всё, что уже было на экране, на "0"

isNewInput = false; // Теперь, когда ты будешь вводить цифры, они будут добавляться к числу, а не начинать новое число

Это помогает избежать ситуаций, когда ты начинаешь ввод с нуля, например, "05".

3. Продолжение ввода:

vvod += "0"; // Добавляем ноль к тому числу, которое уже есть на экране

Например, если было "15", станет "150"

4. Обновление экрана:

updateDisplay(); // Выводим текущее значение vvod на экран

Аналогично происходит, когда ты нажимаешь другие цифры.

```
Button addButton = findViewById(R.id.btn_add);
addButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (vvod.equals("") == false) {
            firstOperand = Double.parseDouble(vvod);
            currentOperator = "+";
            isNewInput = true;
        }
    }
});
```

Рисунок 5. Обработчики арифметических операций

Когда ты нажимаешь на кнопку "+":

Сначала программа проверяет, есть ли что-то на экране.

Здесь — это то, что сейчас видно на экране калькулятора. означает "если экран не пустой".

Если на экране что-то есть, программа запоминает это число:

Например, если на экране "5", программа превращает это в настоящее число 5.0 и сохраняет его в переменную (первое число для операции).

Затем программа говорит калькулятору, что теперь мы будем складывать:

Теперь калькулятор знает, что, когда мы нажмём кнопку "=", нужно будет сложить два числа.

И наконец, программа готовится к вводу второго числа. Она говорит: "следующая цифра, которую ты введёшь, должна начать новое число":

Это нужно, чтобы если после "+", ты нажмёшь "2", получилось "2", а не "52".

```
Button equalsButton = findViewById(R.id.btn_equals);
equalsButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (currentOperator.equals("") == false && isNewInput == false) {
            double secondNumber = Double.parseDouble(vvod);
            double result = 0;

            if (currentOperator.equals("+")) {
                result = firstOperand + secondNumber;
            } else if (currentOperator.equals("-")) {
                result = firstOperand - secondNumber;
            } else if (currentOperator.equals("*")) {
                result = firstOperand * secondNumber;
            } else if (currentOperator.equals("/")) {
                if (secondNumber == 0) {
                    display.setText("Ошибка");
                    return;
                } else {
                    result = firstOperand / secondNumber;
                }
            }

            vvod = Double.toString(result);
            currentOperator = "";
            isNewInput = true;
            updateDisplay();
        }
    }
});
```

Рисунок 6. Обработчик кнопки "=" (равно)

## 1. Вход в функцию

Когда пользователь нажимает на кнопку "=", выполняется этот код.

## 2. Проверка условий

Код проверяет два важных условия:

Оператор должен быть выбран (например, +, -, \*, /).

Должно быть введено второе число.

Если хотя бы одно из этих условий не выполнено, кнопка "=" ничего не делает.

## 3. Получение второго числа

Код преобразует текущее значение на экране в число.

## 4. Выполнение операции

Код выбирает операцию в зависимости от выбранного оператора:

Сложение (+)

Вычитание (-)

Умножение (\*)

Деление (/) с проверкой деления на ноль. Если второе число равно нулю, на экране появляется сообщение "Ошибка".

## 5. Обработка результата

Код преобразует результат вычислений обратно в строку для отображения на экране.

## 6. Сброс состояния

Код сбрасывает:

Выбранный оператор

Устанавливает флаг нового ввода, чтобы следующая цифра начала новое число.

Обновление дисплея

```

Button clearButton = findViewById(R.id.btn_clear);
clearButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        vvod = "";
        firstOperand = 0;
        currentOperator = "";
        isNewInput = true;
        display.setText("0");
    }
});

```

Рисунок 7. Кнопка очистки

```

Button backspaceButton = findViewById(R.id.btn_backspace);
backspaceButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (vvod.length() > 0) {
            vvod = vvod.substring(0, vvod.length() - 1);
            if (vvod.length() == 0) {
                vvod = "0";
                isNewInput = true;
            }
            updateDisplay();
        }
    }
});

```

Рисунок 8. Удаление символа

```

Button decimalButton = findViewById(R.id.btn_decimal);
decimalButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isNewInput == true) {
            vvod = "0.";
            isNewInput = false;
        } else if (vvod.contains(".") == false) {
            vvod = vvod + ".";
        }
        updateDisplay();
    }
});

```

Рисунок 9. Десятичные числа

#### Основная логика работы

##### 1. Два возможных сценария при нажатии:

Если начинается новый ввод (после операции или очистки) → добавляет "0."

Если продолжается текущий ввод → добавляет точку, если её ещё нет

##### 2. Защитные механизмы:

Предотвращает появление нескольких точек в одном числе

Автоматически добавляет ноль перед точкой, если она нажата первой

##### 3. Ключевые переменные:

vvod - строка текущего ввода (то, что отображается на экране)

isNewInput - флаг, указывающий, начинаем ли мы новое число

```

Button mcButton = findViewById(R.id.btn_memory_clear);
mcButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        memoryValue = 0;
    }
});

Button mrButton = findViewById(R.id.btn_memory_recall);
mrButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        vvod = Double.toString(memoryValue);
        isNewInput = true;
        updateDisplay();
    }
});

Button mPlusButton = findViewById(R.id.btn_memory_add);
mPlusButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (vvod.length() > 0) {
            memoryValue = memoryValue + Double.parseDouble(vvod);
        }
    }
});

Button mMinusButton = findViewById(R.id.btn_memory_subtract);
mMinusButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (vvod.length() > 0) {
            memoryValue = memoryValue - Double.parseDouble(vvod);
        }
    }
});

```

Рисунок 10. Работа с памятью

## Основные компоненты

Переменная `memoryValue` - хранит числовое значение в памяти между операциями



Переменная `vvod` - содержит текущий ввод пользователя в виде строки

Четыре кнопки управления памятью

Функционал кнопок памяти

1. MC (Memory Clear) - Очистка памяти

Просто обнуляет сохраненное значение (`memoryValue = 0`)

2. MR (Memory Recall) - Вызов из памяти

Берет сохраненное значение из памяти

Отображает его на экране калькулятора (записывает в `vvod`)

Устанавливает флаг нового ввода (`isNewInput = true`)

Обновляет дисплей

3. M+ (Memory Add) - Добавить к памяти

Берет текущее значение на экране (из `vvod`)

Преобразует его в число

Прибавляет к сохраненному в памяти значению

Результат сохраняется обратно в `memoryValue`

4. M- (Memory Subtract) - Вычесть из памяти

Берет текущее значение на экране (из `vvod`)

Преобразует его в число

Вычитает из сохраненного в памяти значения

Результат сохраняется обратно в `memoryValue`

```

Button sqrtButton = findViewById(R.id.btn_sqrt);
sqrtButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (vvod.length() > 0) {
            double num = Double.parseDouble(vvod);
            if (num >= 0) {
                vvod = Double.toString(Math.sqrt(num));
                isNewInput = true;
                updateDisplay();
            } else {
                display.setText("Ошибка");
            }
        }
    }
});

```

Рисунок 11. Квадратный корень

```

Button percentButton = findViewById(R.id.btn_percent);
percentButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (vvod.length() > 0) {
            double num = Double.parseDouble(vvod);
            vvod = Double.toString(num / 100);
            isNewInput = true;
            updateDisplay();
        }
    }
});

```

Рисунок 12. Процент

```

Button reciprocalButton = findViewById(R.id.btn_reciprocal);
reciprocalButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (vvod.length() > 0) {
            double num = Double.parseDouble(vvod);
            if (num != 0) {
                vvod = Double.toString(1 / num);
                isNewInput = true;
                updateDisplay();
            } else {
                display.setText("Ошибка");
            }
        }
    }
});

```

Рисунок 13. Обратное число

```

private void updateDisplay() { 17 usages
    if (vvod.length() == 0) {
        display.setText("0");
    } else {
        if (vvod.endsWith(".0")) {
            display.setText(vvod.substring(0, vvod.length() - 2));
        } else {
            display.setText(vvod);
        }
    }
}

```

Рисунок 14. Метод updateDisplay() - обновление экрана

## 2.2. Написание frond-end

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

```

Рисунок 15. Главная форма

```

<EditText
    android:id="@+id/display"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="32sp"
    android:inputType="none"
    android:gravity="end"
    android:padding="8dp"
    android:background="@android:color/transparent"
    android:text="" />

```

Рисунок 16. Поле ввода

```

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

    <Button
        android:id="@+id/btn_clear"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="C" />

    <Button
        android:id="@+id/btn_backspace"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="⌫" />

    <Button
        android:id="@+id/btn_memory_clear"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="MC" />

    <Button
        android:id="@+id/btn_memory_recall"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="MR" />

</LinearLayout>

```

Рисунок 17. Управление и память

```

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

    <Button
        android:id="@+id/btn_memory_add"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="M+"/>

    <Button
        android:id="@+id/btn_memory_subtract"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="M-"/>

    <Button
        android:id="@+id/btn_sqrt"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="√"/>

    <Button
        android:id="@+id/btn_percent"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="%"/>
</LinearLayout>

```

Рисунок 18. Доп. функции

M+/M- - операции с памятью

√ - квадратный корень

% - проценты

```

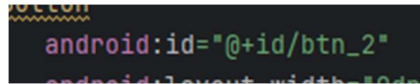
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"

```

Рисунок 19. Система вёрстки

layout\_weight="1" - все кнопки в строке получают равную ширину

$0dp + weight$  - стандартный способ равномерного распределения



```
android:id="@+id/btn_2"
```

Рисунок 20. Связь с кодом

Присваиваем объекту уникальный идентификатор для его распознавания и отслеживания.

## ЗАКЛЮЧЕНИЕ

Прохождение учебной практики является важным элементом учебного процесса по подготовке специалиста в области компьютерных наук. Во время её прохождения будущий программист применяет полученные в процессе обучения знания, умения и навыки на практике.

В данной практике было изучены, как создавать проект на android studio, создавать форму для проекта, его код и взаимодействие frond-end с back-end, работа с git

Поставленная задача и цель выполнены успешно.

**НПОУ «Якутский колледж инновационных технологий»**

(наименование образовательной организации)

**Отделение Информационных технологий**

(наименование структурного подразделения (кафедра / отделение))

УТВЕРЖДАЮ  
заведующий отделением\_\_\_\_\_  
/ И.В. Пронин/  
« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.**РАБОЧИЙ ГРАФИК (ПЛАН) ПРОВЕДЕНИЯ ПРАКТИКИ****Общие сведения**

ФИО обучающегося	Оконешников Родион Николаевич
Курс	2
Форма обучения	Очная
Направление подготовки / специальность / профессия	09.02.07 Информационные системы и программирование
Наименование структурного подразделения (кафедра / отделение)	Отделение информационных технологий
Группа	КИСП-23
Вид практики	Учебная
Тип практики	По модулю
Способ проведения практики	Стационарно
Форма проведения практики	Дискретно
Место прохождения практики	НПОУ «ЯКИТ»
Период прохождения практики	с 19.05.25 по 25.05.25
Реквизиты договора о прохождении практики (при проведении практики в профильной организации)	

**Планируемые работы**

№ п/п	Содержание работы	Срок выполнения	Отметка о выполнении
1.	Оформление документов по прохождению практики	до начала практики	
2.	Проведение медицинских осмотров (обследований) в случае выполнения обучающимся работ, при выполнении которых проводятся обязательные предварительные и периодические медицинские осмотры (обследования) в соответствии с законодательством РФ	до начала практики	
3.	Вводный инструктаж по правилам охраны труда, технике безопасности, пожарной безопасности, оформление временных пропусков для прохода в профильную организацию (при необходимости).	в первый день практики	
4.	Выполнение индивидуального задания практики	в период практики	



5.	Консультации руководителя(-ей) практики о ходе выполнения заданий, оформлении и содержании отчета, по производственным вопросам	в период практики	
6.	Подготовка отчета по практике	за два дня до промежуточной аттестации	
7.	Проверка отчета по практике, оформление характеристики руководителя(-ей) практики	за два дня до промежуточной аттестации	
8.	Промежуточная аттестация по практике	в последний день практики	

Рабочий график (план) составил:  
руководитель практики от образовательной организации

_____	_____	А.К. Цвикальский	«19» мая 2025 г.
(уч. степень, уч. звание, должность)	(подпись)	(И.О. Фамилия)	(дата)

С рабочим графиком (планом) ознакомлен:  
обучающийся

_____	Р.Н. Оконешников	«19» мая 2025 г.
(подпись)	(И.О. Фамилия)	(дата)

# НПОУ «Якутский колледж инновационных технологий»

(наименование образовательной организации)

## Отделение Информационных технологий

(наименование структурного подразделения (кафедра / отделение))

### ДНЕВНИК ПРАКТИКИ

#### Общие сведения

ФИО обучающегося	Оконешников Родион Николаевич
Курс	2
Форма обучения	Очная
Направление подготовки / специальность / профессия	09.02.07 Информационные системы и программирование
Наименование структурного подразделения (кафедра / отделение)	Отделение информационных технологий
Группа	КИСП-23
Вид практики	Учебная
Тип практики	По модулю
Способ проведения практики	Стационарно
Форма проведения практики	Дискретно
Место прохождения практики	НПОУ «ЯКИТ»
Период прохождения практики	с 19.05.25 по 25.05.25
Реквизиты договора о прохождении практики (при проведении практики в профильной организации)	

#### Учет выполняемой работы

№ п/п	Содержание работы	Дата выполнения	Отметка о выполнении
1.	Прошел инструктаж по технике безопасности	19. 05.2025	
2.	Ознакомление с заданием	20. 05.2025	
3.	Подготовка среды для разработки	21. 05.2025	
4.	Реализация задания	22. 05.2025	
5.	Работа над front-end	23. 05.2025	
6.	Написание отчета	24. 05.2025	
7.	Сдача отчета	25.05.2025	

Дневник заполнил:  
обучающийся

(подпись)

Р.Н. Оконешников  
(И.О. Фамилия)

«25» мая 2025 г  
(дата)

Дневник проверил:  
руководитель практики от образовательной организации

(уч. степень, уч. звание,  
должность)

(подпись)

А.К. Цвикальский  
(И.О. Фамилия)

«25» мая 2025 г  
(дата)