

Министерство образования и науки Кыргызской Республики
Кыргызский государственный технический университет им.

И.Раззакова

Филиал им. академика Х.А. Рахматулина в г. Токмок

Кафедра «Программное обеспечение компьютерных систем»

Отчет по учебной практике

Выполнила: ст. гр. ИСЭ-24 Бекжанова Д.Ш

Руководитель Учебной практики:

старший преподаватель Забабурина М.В

Сроки прохождения с 02.06.2025 г. по 30.06. 2025 г.

Дата сдачи: 30.06.2025 г.

Оценка 90 (отм) 

(отметка о сдаче, роспись руководителя)

Токмок 2025

Задание на учебную практику

Направление: 710200 – Информационные системы и технологии.

Вариант № 2

Задание 1. Классификация языков программирования. Системы программирования

Изучить основные типы языков программирования: машинный, ассемблерный, языки высокого уровня..

Задание 2. Программное обеспечение ПК: базовое, системное, служебное, прикладное

Дать описание видам программного обеспечения.

Задание 3. Программа для расчёта по формулам

Написать программу на языке Python, которая рассчитывает значения по формулам:

$$z_1 = \frac{x^2 + 2x - 3 + (x + 1) \sqrt{x^2 - 9}}{x^2 - 2x - 3 + (x - 1) \sqrt{x^2 - 9}};$$
$$z_2 = \sqrt{\frac{x + 3}{x - 3}}.$$

Рисунок 1— Фрагмент задания по программированию

Задание 4. Пересчёт минут в часы и минуты

Составить программу, которая преобразует временной интервал в минутах в формат “часы и минуты”. Например, 130 минут → 2 часа 10 минут.

СОДЕРЖАНИЕ

Задание на учебную практику	2
ВВЕДЕНИЕ.....	4
1.2 Безопасность при работе за ПК	5
Задание 1. Классификация языков программирования. Системы программирования	6
1.1 Что такое язык программирования	6
1.3 Системы программирования.....	8
Задание 2. Программное обеспечение персонального компьютера (ПК) ..	9
2.1 Понятие программного обеспечения	9
2.2 Базовое программное обеспечение	9
2.4 Системное программное обеспечение	10
2.5 Служебное программное обеспечение.....	11
2.6 Прикладное программное обеспечение	11
2.7 Взаимодействие программных компонентов	12
Задание 3. Программа для расчёта по двум формулам	14
3.1 Постановка задачи	14
3.2 Алгоритм решения задачи.....	14
3.3 Листинг программы Python.....	15
3.4 Результат работы программы.....	16
Задание 4. Программа пересчёта минут в часы и минуты	17
4.1 Постановка задачи	17
4.2 Алгоритм решения задачи.....	17
4.4 Результат работы программы.....	18
Заключение	19
Список использованной литературы	20

ВВЕДЕНИЕ

Учебная практика по дисциплине «Информационные системы и технологии» направлена на комплексное закрепление теоретических знаний и развитие практических навыков, необходимых в современной ИТ-среде. В ходе практики студент знакомится с ключевыми аспектами информационных технологий:

- Изучение природы и классификации компьютерных вирусов — классификация по типам, методам заражения и целям.
- Принципы работы операционных систем — обзор функций ОС, их роли в защите информации и управлении ресурсами.
- Программирование и алгоритмизация — разработка базовых программ для математических расчётов и обработки данных, что закладывает фундамент навыков решения реальных задач.

Основная цель практики — формирование умений:

1. Понимать ИТ-понятия и анализировать информационные угрозы.
2. Разрабатывать простые программные решения.
3. Вырабатывать навыки критического мышления и структурированного подхода к задачам.

Задачи практики включают:

- изучение природы и классификации компьютерных вирусов;
- анализ различных видов и функций операционных систем;
- реализацию программ для математических расчётов и алгоритмов поиска/обработки чисел;

- закрепление навыков исследования ИТ-объектов и алгоритмизации процессов .

1.2 Безопасность при работе за ПК

Важным аспектом является соблюдение техники безопасности при работе:

- освещение рабочего места должно быть достаточным и не вызывать бликов;
- расстояние от глаз до экрана — не менее 500 мм; экран — на уровне глаз;
- кресло и стол — эргономичные, обеспечивающие комфортную осанку;
- перерывы каждые 40–50 минут, предпочтительно для гимнастики или смены активности;
- регулярные проверки техники: исправность заземления, кабелей; запрет касания корпуса мокрыми руками, соблюдение инструкций производителя.

Отдельное внимание в ходе практики уделяется вопросам техники безопасности при работе с компьютерной техникой. Студент учится правильно организовывать рабочее место, соблюдать режим труда и отдыха, а также следить за осанкой и состоянием зрения при работе за экраном.

Представленный отчёт содержит описание всех этапов выполнения практических заданий, сопровождаемых алгоритмами, кодом программ и анализом полученных результатов

Задание 1. Классификация языков программирования. Системы программирования

1.1 Что такое язык программирования

Язык программирования — это формализованная система знаков, предназначенная для описания алгоритмов, структур данных и управления поведением компьютера. Он служит посредником между человеком и машиной.

1.2 Классификация языков программирования

1. По уровню абстракции

- Языки низкого уровня — максимально приближены к машинному коду. Они обеспечивают полный контроль над аппаратной частью, но сложны в использовании.
- Пример: Ассемблер.
- Языки высокого уровня — позволяют писать программы, используя конструкции, близкие к естественному языку.
- Примеры: Python, Java, C++, Pascal.

2. По способу выполнения

- Компилируемые языки — переводятся целиком в машинный код до выполнения.
- Примеры: C, C++, Pascal.
- Интерпретируемые языки — исполняются построчно с помощью интерпретатора.

- Примеры: Python, JavaScript.
- Смешанные — имеют как компиляцию, так и интерпретацию (Java: компиляция в байт-код + интерпретация JVM).

3. По парадигме программирования

- Процедурные — акцент на последовательное выполнение команд (алгоритмы).
- Примеры: C, Pascal.
- Объектно-ориентированные (ООП) — моделируют систему через объекты и классы.
- Примеры: Java, C++, Python.
- Функциональные — основаны на использовании математических функций.
- Примеры: Haskell, Lisp.
- Логические — работают на основе логических выражений и правил.
- Пример: Prolog.

4. По назначению

- Универсальные — используются для разных целей (Python, Java).
- Специализированные — предназначены для определённой сферы.
- SQL — базы данных
- HTML — веб-разметка
- MATLAB — инженерные вычисления

1.3 Системы программирования

Система программирования — это программный комплекс, предназначенный для создания, тестирования и отладки программ.

Включает:

- Редактор исходного кода (например, Notepad++, VS Code)
- Компилятор/интерпретатор (gcc, python.exe)
- Отладчик (GDB, встроенные в IDE)
- Библиотеки (стандартные и внешние)
- Интегрированные среды разработки (IDE):
- Visual Studio
- PyCharm
- IntelliJ IDEA
- Code::Blocks

Задание 2. Программное обеспечение персонального компьютера (ПК)

2.1 Понятие программного обеспечения

Программное обеспечение (ПО) — это совокупность программ, используемых для управления аппаратными ресурсами компьютера и выполнения различных задач. Без программного обеспечения аппаратная часть компьютера (железо) остаётся неактивной и бесполезной.

Современное ПО делится на четыре основные категории:

- базовое
- системное
- служебное
- прикладное

2.2 Базовое программное обеспечение

Это минимальный набор программ, необходимый для запуска и функционирования компьютера. Оно хранится в энергонезависимой памяти (обычно ROM или BIOS/UEFI).

Компоненты:

- BIOS (Basic Input/Output System) — инициализирует оборудование при включении, обеспечивает первичную загрузку ОС.
- UEFI (современный аналог BIOS) — более безопасный и гибкий интерфейс для загрузки ОС.

- Загрузчик (Boot Loader) — программа, которая загружает ОС из постоянной памяти (жёсткий диск, SSD).

Функции базового ПО:

- Аппаратная инициализация
- Проверка оборудования (POST)
- Запуск операционной системы

2.4 Системное программное обеспечение

Это ПО, управляющее всеми аппаратными и программными компонентами компьютера. Оно обеспечивает взаимодействие пользователя и машины.

Главный компонент системного ПО — операционная система (ОС).

Функции ОС:

- Управление файлами и папками
- Управление памятью
- Управление процессами
- Управление устройствами ввода/вывода
- Обеспечение пользовательского интерфейса
- Обеспечение сетевого взаимодействия

Примеры операционных систем:

- Windows — универсальная ОС от Microsoft
- Linux — свободно распространяемая ОС с открытым кодом (Ubuntu, Mint)
- macOS — ОС Apple
- Android — мобильная ОС на базе ядра Linux

2.5 Служебное программное обеспечение

Служебные программы (утилиты) предназначены для диагностики, обслуживания и оптимизации системы.

Примеры служебных программ:

- Антивирусные программы (Kaspersky, Avast)
- Программы для дефрагментации диска
- Утилиты очистки системы (CCleaner)
- Мониторинг температуры, нагрузки, оперативной памяти

Функции:

- Поддержание стабильности ОС
- Повышение безопасности и производительности
- Обнаружение и устранение неисправностей

2.6 Прикладное программное обеспечение

Это основная категория программ, с которыми работает пользователь. Такие программы решают конкретные задачи.

Примеры прикладных программ:

- Текстовые редакторы: Microsoft Word, LibreOffice Writer
- Электронные таблицы: Excel, Google Sheets
- Графические редакторы: Adobe Photoshop, CorelDRAW
- Браузеры: Google Chrome, Mozilla Firefox
- Мультимедийные плееры: VLC Media Player, Windows Media
- Образовательные программы: Duolingo, Geogebra
- Программы для программирования: PyCharm, Visual Studio Code

2.7 Взаимодействие программных компонентов

ПО на ПК работает по уровням:

1. Базовое ПО → 2. Системное ПО → 3. Служебное ПО → 4.

Прикладное ПО

Это иерархия, где каждый уровень зависит от предыдущего.

Вывод:

Программное обеспечение ПК представляет собой сложную иерархическую структуру, включающую компоненты от базовой загрузки до пользовательских приложений. Грамотное понимание структуры ПО

позволяет эффективно использовать ресурсы компьютера, обеспечивать его безопасность и производительность.

Задание 3. Программа для расчёта по двум формулам

3.1 Постановка задачи

Цель задания — разработать программу, которая выполняет вычисление двух математических выражений по заданному значению переменной x .

Формулы:

$$z_1 = \frac{x^2 + 2x - 3 + (x + 1) \sqrt{x^2 - 9}}{x^2 - 2x - 3 + (x - 1) \sqrt{x^2 - 9}};$$
$$z_2 = \sqrt{\frac{x + 3}{x - 3}}.$$

Рисунок 3.1— Постановка задачи и формулы для вычислений

Входные данные:

- Одно действительное число x , вводимое пользователем.

Выходные данные:

- Значения выражений z_1 и z_2 .

Ограничения:

- Подкоренные выражения должны быть ≥ 0 , чтобы избежать ошибки извлечения корня.
- Знаменатель в формуле z_1 не должен быть равен нулю.
- $x \neq 3$ и $x \neq -3$ (во избежание деления на ноль).

3.2 Алгоритм решения задачи

Блок-схема к заданию 3 представлена ниже

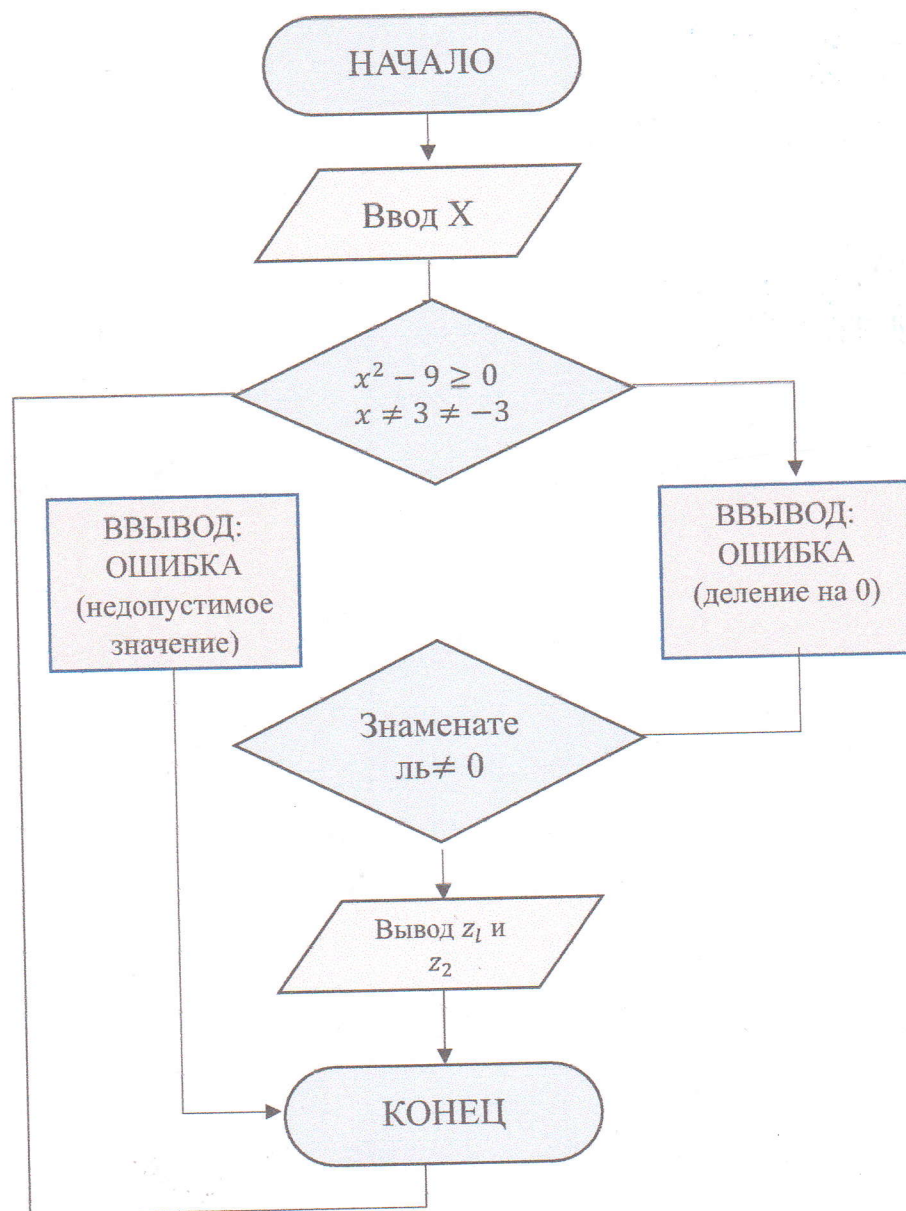


Рисунок 3.2 — Блок-схема алгоритма вычисления выражений z^1 и z^2

3.3 Листинг программы Python

```

import math

x = float(input("Введите значение x: "))

# Проверка корней и деления на ноль
if x**2 - 9 < 0 or x == 3 or x == -3:
    print("Ошибка: недопустимое значение x")
else:
    numerator = x*2 + 2*x - 3 + (x + 1) * math.sqrt(x*2 - 9)
    denominator = x*2 - 2*x - 3 + (x - 1) * math.sqrt(x*2 - 9)

    if denominator == 0:
        print("Ошибка: знаменатель равен нулю")
    else:
        z1 = numerator / denominator
        z2 = math.sqrt((x + 3) / (x - 3))
        print(f"Результат вычислений:")
        print(f"z1 = {z1}")
        print(f"z2 = {z2}")

```

Рисунок 3.3 — Программа на языке Python для расчёта z^1 и z^2

3.4 Результат работы программы

Пример 1:

```

Введите значение x: 5
Результат вычислений:
z1 = 23.0
z2 = 2.0

```

Рисунок 3.4 — Результат работы программы при корректном значении

Пример 2:

```

Введите значение x: 2
Ошибка: недопустимое значение x

```

Рисунок 3.5 — Результат работы программы при недопустимом значении

В ходе выполнения задания была разработана программа для вычисления выражений z_1 и z_2 . Реализованы все проверки условий: извлечение корня, деление на ноль. Алгоритм оформлен в виде блок-схемы. Программа работает корректно и выдаёт правильные результаты.

Задание 4. Программа пересчёта минут в часы и минуты

4.1 Постановка задачи

Требуется написать программу, которая преобразует временной интервал, заданный в минутах, в количество полных часов и оставшихся минут.

Входные данные:

Одно целое число мин, представляющее собой количество минут.

Выходные данные:

- Количество часов (часы)
- Количество оставшихся минут (остаток)

4.2 Алгоритм решения задачи

Блок-схема к заданию 4 представлена ниже

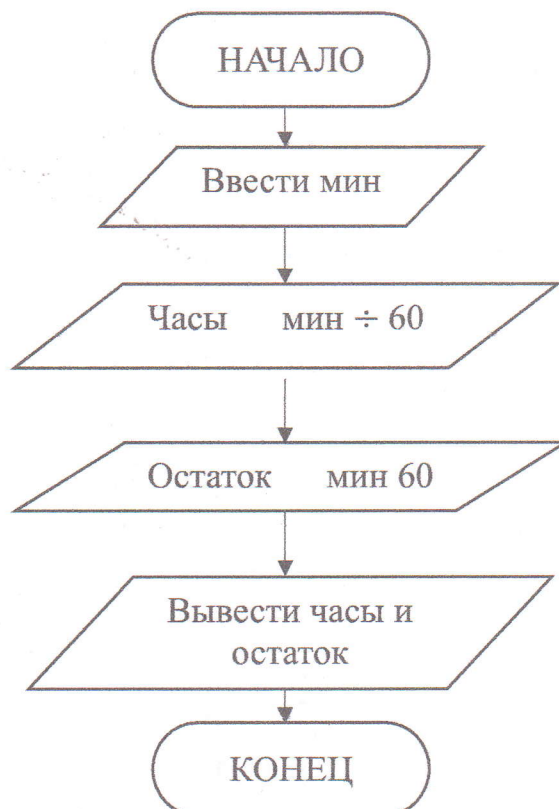


Рисунок 4.1 — Блок схема алгоритма пересчёта минут

4.3 Листинг программы Python

```
мин = int(input("Введите количество минут: "))  
часы = мин // 60  
остаток = мин % 60  
print(f"{мин} минут = {часы} ч. {остаток} мин.")
```

Рисунок 4.2 — Программа на языке Python для пересчёта минут в часы и минуты

4.4 Результат работы программы

Пример 1:

```
Введите количество минут: 139  
139 минут = 2 ч. 19 мин.
```

Рисунок 4.3 — Результат работы программы при вводе 139 мин

Пример 2:

```
Введите количество минут: 45  
45 минут = 0 ч. 45 мин.
```

Рисунок 4.4 — Результат работы программы при вводе 45 мин

Была написана простая и понятная программа для перевода минут в часы и минуты. Алгоритм оформлен в виде блок-схемы. Программа работает корректно и выдаёт верный результат при любом вводе.

Заключение

В ходе прохождения учебной практики по дисциплине «Информационные системы и технологии» были рассмотрены и выполнены как теоретические, так и практические задания. Практика позволила систематизировать и углубить знания в области программного обеспечения, языков программирования и основ разработки алгоритмов.

В теоретической части были изучены классификация языков программирования, их особенности и области применения. Также подробно рассмотрена структура программного обеспечения, его виды и функции. Полученные знания позволили лучше понять, как устроена работа персонального компьютера на уровне программной системы.

В практической части мною были разработаны две программы: одна для выполнения расчётов по заданным математическим формулам, вторая — для преобразования количества минут в часы и минуты. Каждое задание сопровождалось точной постановкой задачи, построением алгоритма, оформлением блок-схемы, написанием и отладкой программного кода на языке Python, а также анализом полученных результатов.

Проведённая работа позволила не только закрепить полученные знания, но и развить навыки логического мышления, внимательности и точности при составлении алгоритмов и написании программ. Все поставленные задачи были выполнены в полном объёме, цель практики достигнута.

Список использованной литературы

1. Глушаков И. В. Программное обеспечение ЭВМ. — М.: Форум, 2021.
2. Таненбаум Э., Бос Х. Современные операционные системы. — СПб.: Питер, 2020.
3. Метанит. Язык программирования Python. — <https://metanit.com/python/>
4. Хабр. Объяснение алгоритмов и блок-схем. — <https://habr.com/ru/articles/440436/>
5. Microsoft Docs — Операционные системы Windows. — <https://learn.microsoft.com/ru-ru/windows/>
6. Wikipedia. Программное обеспечение. — https://ru.wikipedia.org/wiki/Программное_обеспечение
7. Python. Официальная документация. — <https://docs.python.org/3/>