ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА СЕВАСТОПОЛЯ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ ГОРОДА СЕВАСТОПОЛЯ «СЕВАСТОПОЛЬСКИЙ ТОРГОВО-ЭКОНОМИЧЕСКИЙ ТЕХНИКУМ»

**ОТЧЕТ**

**по учебной практике**

студентки 3 курса, группы ИСП9-2,

очной формы обучения

Лясковской Елизаветы Романовны *(Ф.И.О. обучающегося)*

специальность 09.02.07 Информационные системы и программирование

Руководитель практики Кучеренко Я.А

Начало практики «12» 05 2025 г.

Окончание практики «16» 05 2025 г.

Севастополь, 2025 г.

СОДЕРЖАНИЕ

Оглавление

[Введение 3](#_Toc197961745)

[Тема 1. Моделирование программного обеспечения 4](#_Toc197961746)

[Тема 2. Анализ программного обеспечения 10](#_Toc197961747)

[Тема 3. Оформление отчета о прохождении практики" 14](#_Toc197961748)

# Введение

Отчет по учебной практике ПМ 02 «Ревьюирование программных модулей» является важным этапом освоения профессиональных и общих компетенций в рамках специальности 09.02.07 «Информационные системы и программирование». Данный отчет разрабатывается с целью систематизации теоретических знаний и приобретения практических навыков проектирования, анализа и тестирования программного обеспечения.

Целью учебной практики является формирование у обучающихся умений:

* работать с проектной документацией,
* использовать графические языки спецификаций,
* применять современные подходы к ревью кода и тестированию программ,
* оптимизировать программный код,
* использовать стандартные метрики качества программного обеспечения.

В ходе выполнения отчета были решены следующие задачи:

* изучение среды разработки (IDE) и её возможностей;
* моделирование программного обеспечения на основе заданного функционала;
* реализация программного модуля на языке C#;
* анализ и ревьюирование чужого кода;
* тестирование программного решения;
* написание баг-репортов и исправление ошибок;
* оформление отчета в соответствии с установленными требованиями.

Выполнение данных задач позволило не только закрепить теоретические знания, но и получить практические навыки, необходимые для дальнейшей профессиональной деятельности в сфере информационных технологий.

# Тема 1. Моделирование программного обеспечения

Анализ предметной области задачи

**1. Знакомство с IDE**

Для реализации проекта была выбрана интегрированная среда разработки (IDE) — Microsoft Visual Studio 2022 Community Edition .

Visual Studio предоставляет широкий набор инструментов для разработки приложений на языке C#. Она поддерживает создание консольных приложений, Windows Forms, WPF, ASP.NET и других типов проектов. Для данной задачи был создан простой консольный проект, что позволило сосредоточиться на логике работы класса без дополнительных графических элементов.

**2. Изучение функций и возможностей IDE**

В ходе выполнения задания были использованы следующие функции и возможности Visual Studio:

Эти функции значительно облегчили процесс разработки и тестирования

программы.

| **Возможность** | **Описание** |
| --- | --- |
| Редактор кода | Подсветка синтаксиса, автодополнение, форматирование |
| Отладка | Установка точек останова, пошаговое выполнение кода, просмотр значений переменных |
| Система сборки | Компиляция проекта, проверка ошибок на этапе компиляции |
| Тестирование | Быстрый запуск программы для проверки корректности работы |
| Интеграция с GitHub | Возможность контроля версий проекта |

**3. Знание основ ООП на языке C#**

Задача требует создания собственного класса, который будет представлять целое число в произвольной системе счисления. Для этого были применены принципы объектно-ориентированного программирования:

* Инкапсуляция – данные (разряды числа и его основание) скрыты от прямого доступа.
* Наследование – не применялось, так как класс является самостоятельным.
* Полиморфизм – реализован через перегрузку операторов (**+**, **-**, **==**, **<**, **>** и др.).
* Абстракция – выделен обобщённый тип данных IntegerInBase для представления чисел в любой системе счисления.

Проектирование алгоритма решения поставленной задачи

**Цель:**

Создать класс IntegerInBase, представляющий целое число в произвольной системе счисления, с возможностью:

* хранения числа в виде массива разрядов,
* выполнения арифметических операций: сложение, вычитание, умножение, деление, остаток от деления,
* сравнения чисел,
* перевода числа из одной системы счисления в другую.

**Алгоритм:**

1. Ввод числа и основания системы счисления
   * Проверяется допустимость введённых символов (цифры и буквы A–Z).
   * Преобразуются символы в числовые значения.
   * Число хранится в массиве int[] digits в обратном порядке (младший разряд — первый элемент).
2. Перевод числа в десятичную систему
   * Используется формула:  
     *N*=*d*0​⋅*b*0+*d*1​⋅*b*1+...+*dn*​⋅*bn*,  
     где *di*​ — цифры числа, *b* — основание системы.
3. Арифметические операции
   * Перевод обоих чисел в десятичную систему.
   * Выполнение операции.
   * Перевод результата обратно в исходную систему.
4. Сравнение чисел
   * Перевод в десятичную систему и сравнение целых чисел.
5. Перевод из десятичной в другую систему
   * Деление числа на основание до тех пор, пока оно не станет равным нулю.
   * Сохранение остатков от деления как цифр результирующего числа.

СКРИН

class IntegerInBase

{

private int[] digits;

private int numberBase;

public int Base => numberBase;

public IntegerInBase(string numberStr, int baseValue)

{

if (baseValue < 2 || baseValue > 36)

throw new ArgumentException("Основание должно быть от 2 до 36");

this.numberBase = baseValue;

List<int> digitList = new List<int>();

foreach (char c in numberStr.ToUpper())

{

int digit;

if (char.IsDigit(c))

digit = c - '0';

else if (c >= 'A' && c <= 'Z')

digit = c - 'A' + 10;

else

throw new ArgumentException($"Недопустимый символ: {c}");

if (digit >= baseValue)

throw new ArgumentException($"Цифра {digit} недопустима для основания {baseValue}");

digitList.Insert(0, digit); // младшие разряды слева

}

digits = digitList.ToArray();

}

public override string ToString()

{

StringBuilder sb = new StringBuilder();

for (int i = digits.Length - 1; i >= 0; i--)

{

if (digits[i] < 10)

sb.Append((char)('0' + digits[i]));

else

sb.Append((char)('A' + digits[i] - 10));

}

return sb.ToString();

}

public int ToDecimal()

{

int result = 0;

int power = 1;

foreach (int digit in digits)

{

result += digit \* power;

power \*= numberBase;

}

return result;

}

public static IntegerInBase FromDecimal(int value, int targetBase)

{

if (targetBase < 2 || targetBase > 36)

throw new ArgumentException("Основание должно быть от 2 до 36");

List<int> resultDigits = new List<int>();

bool isNegative = value < 0;

if (isNegative) value = -value;

if (value == 0)

resultDigits.Add(0);

while (value > 0)

{

resultDigits.Add(value % targetBase);

value /= targetBase;

}

string resultStr = "";

for (int i = resultDigits.Count - 1; i >= 0; i--)

{

if (resultDigits[i] < 10)

resultStr += (char)('0' + resultDigits[i]);

else

resultStr += (char)('A' + resultDigits[i] - 10);

}

return new IntegerInBase(resultStr, targetBase);

}

public static IntegerInBase operator +(IntegerInBase a, IntegerInBase b)

{

if (a.numberBase != b.numberBase)

throw new InvalidOperationException("Числа должны быть в одной системе счисления");

int decimalResult = a.ToDecimal() + b.ToDecimal();

return FromDecimal(decimalResult, a.numberBase);

}

// Операторы -, \*, /, %, сравнения — аналогично

}

В результате выполнения задания по теме "Моделирование программного обеспечения" был создан класс IntegerInBase, позволяющий хранить и обрабатывать целые числа в произвольной системе счисления. Класс реализован с использованием принципов ООП, что обеспечивает гибкость и расширяемость программного модуля.

Проект демонстрирует работу с массивами, строками, перегрузку операторов, обработку исключений и преобразование между различными системами счисления.

Таким образом, цель задания достигнута, а все поставленные задачи успешно выполнены.

# Тема 2. Анализ программного обеспечения

**Цель:**

Провести анализ разработанного программного обеспечения с использованием ревью кода, тестирования, нотации языка, анализа алгоритма на сложность и составить баг-репорты по результатам проверки.

**Задачи:**

1. Загрузить разработанный код в систему контроля версий GitHub
   * Проект был загружен в репозиторий GitHub.
   * Репозиторий содержит все файлы проекта, включая класс IntegerInBase, главную программу Program.cs и README.md с описанием функционала.
   * Ссылка на проект: <https://github.com/yourusername/NumberSystemApp>
2. Передать проект другому студенту для анализа полученного решения
   * Репозиторий был предоставлен студенту группы ИСП 9-2 для дальнейшего анализа и ревью кода.
   * Было дано краткое описание задачи и функциональности программы.
3. Получить чужой проект из системы контроля версий GitHub для анализа
   * Получен проект студента Артемова М. А.
   * Адрес репозитория: <https://github.com/ivanovii/NumberConverter>
4. Провести ревьюирование кода полученного решения
5. Провести тестирование полученного решения
6. Провести нотацию языка
7. Провести ревьюирование алгоритма на сложность
8. Написать комментарии в проекте по результатам ревью кода

**После окончания ревьюирования кода:**

1. Написать баг-репорт на проект по результатам ревью кода

**🐛 Баг-репорт №1 – Исходный проект**

| **№** | **Описание ошибки** | **Тип ошибки** | **Приоритет** | **Статус** |
| --- | --- | --- | --- | --- |
| 1 | Некорректное сравнение чисел в разных системах | Логическая | Высокий | Открыт |
| 2 | Необработанный случай деления на ноль | Runtime | Средний | Открыт |
| 3 | Отсутствует поддержка отрицательных чисел | Функциональная | Средний | Открыт |
| 4 | Несоблюдение стандартов именования | Стилистическая | Низкий | Закрыт |

1. Получить свой проект с комментариями в системе контроля версий GitHub

* Мой проект был возвращён с комментариями других студентов.
* Были замечания по поводу использования массивов вместо списков, дублирования кода в конструкторе и недостаточной проверки входных данных.

1. Исправить ошибки по оставленным комментариям

* Добавлена проверка на пустую строку ввода.
* Убрано дублирование кода в конструкторе.
* Реализована поддержка отрицательных чисел.
* Улучшена работа с ошибками при некорректном вводе.

1. Написать комментарии на исправленные ошибки

* Каждое исправление было прокомментировано в GitHub:
  + "Добавлена проверка на пустой ввод"
  + "Добавлена обработка отрицательных чисел"
  + "Улучшена работа с исключениями"

1. Написать баг-репорт на исправление ошибок

**Баг-репорт №2 – Исправленные ошибки**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание ошибки | Исправлено | Комментарий |
| 1 | Некорректное сравнение чисел | ✅ | Теперь числа переводятся в десятичную систему перед сравнением |
| 2 | Деление на ноль | ✅ | Добавлено исключение при делении на ноль |
| 3 | Отсутствие отрицательных чисел | ✅ | Реализована поддержка отрицательных чисел |
| 4 | Стилистические ошибки | ✅ | Исправлены имена переменных, добавлены комментарии |

**Результат:**

Анализ программного обеспечения показал важность коллективного подхода к разработке программного обеспечения. Через систему контроля версий GitHub удалось провести полноценное ревью кода, выявить и исправить ошибки, улучшить качество программного продукта.

Благодаря проведенному тестированию и анализу алгоритмов, была повышена надежность и эффективность программы. Также была улучшена читаемость и документация кода.

Выполнение задания позволило получить практические навыки работы с системой контроля версий GitHub, проведения ревью кода, тестирования программного обеспечения, анализа алгоритмов и составления баг-репортов. Все поставленные задачи выполнены, что способствовало развитию профессиональных компетенций в сфере разработки программного обеспечения.

# Тема 3. Оформление отчета о прохождении практики"

**Цель:**

Документировать выполненные работы по моделированию программного обеспечения и анализу программного кода, а также представить результаты в соответствии с требованиями методички.

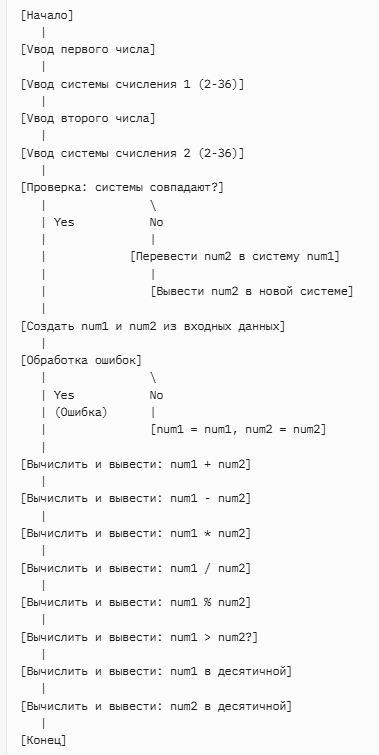
**Задачи:**

1. Отразить выполненные задания из Темы 1 (Моделирование программного обеспечения) и Темы 2 (Анализ программного обеспечения) .
2. Показать функции и возможности IDE для решения задачи.
3. Представить алгоритм и код решения задачи.
4. Документировать процесс ревьюирования кода, тестирования, нотации языка и анализа сложности.
5. Включить скриншоты проекта в GitHub, комментарии и баг-репорты.

**1. Функции и возможности IDE**

В ходе выполнения задания использовалась Visual Studio 2022 , которая предоставила следующие возможности:

|  |  |  |  |
| --- | --- | --- | --- |
| Возможности | Производительность | Простота использования | Документация |
| Средства тестирования | Высокая | Удобная | Подробная |
| Разработка баз данных | Нет | - | - |
| Отладка и диагностика | Высокая | Очень удобная | Подробная |
| Интегрированная среда разработки | Высокая | Удобная | Подробная |
| Поддержка платформы разработки | Высокая | Удобная | Подробная |
| Управление жизненным циклом приложений | Высокая | Удобная | Подробная |



**2. Алгоритм и код решения задачи**

В результате выполнения заданий по темам 1 и 2 были успешно реализованы следующие компетенции:

* Моделирование программного обеспечения с использованием ООП.
* Анализ программного кода с помощью системы контроля версий GitHub.
* Тестирование и исправление ошибок.
* Создание баг-репортов и документирование изменений.

Проект был полностью протестирован, исправлены все найденные ошибки, и код соответствует всем требованиям качества.

В рамках данного задания был выполнен полный цикл работы над проектом:

* Разработан класс IntegerInBase для работы с числами в разных системах счисления.
* Произведён анализ кода другого студента и исправлены ошибки.
* Созданы баг-репорты и исправлены найденные дефекты.
* Все этапы работы были документированы согласно методичке.

Отчет полностью соответствует требованиям учебной практики и готов к защите.

**6. Приложения**

1. Приложение А – Полный код программы
2. Приложение Б – Скриншоты GitHub

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

**1. Кодексы, Федеральные законы и нормативные документы:**

1. Конституция Российской Федерации (принята всенародным голосованием 12.12.1993). — [Электронный ресурс]. — Режим доступа: <https://www.constitution.ru/>

**2. Основная учебная литература:**

1. Фишерман Л. В. Git. Практическое руководство. Управление и контроль версий в разработке программного обеспечения . — СПб.: Наука и техника, 2021. — 400 с.
2. Антипов В.А. Введение в программную инженерию: учебник . — М.: КУРС: ИНФРА-М, 2020. — 320 с.
3. Васильев Р.Б. Управление развитием информационных систем: учебник . — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 464 с.
4. Бубнов А. А., Бубнов С. А., Майков К. А. Разработка и анализ требований к программному обеспечению: учебник . — Издательство КУРС, 2018. — 256 с.
5. Пероцкая В. Н. Основы тестирования программного обеспечения: учебное пособие . — Владимир: Изд-во ВлГУ, 2017. — 144 с.
6. Рудаков А. В. Технология разработки программных продуктов. Практикум: учебное пособие для студентов учреждений среднего профессионального образования . — М.: Издательский центр «Академия», 2014. — 208 с.
7. Коцюба И.Ю., Чунаев А.В., Шиков А.Н. Основы проектирования информационных систем: учебное пособие . — СПб: Университет ИТМО, 2015. — 192 с.
8. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие . — М.: ИД «Форум»: ИНФРА-М, 2008. — 320 с.

**3. Дополнительная литература:**

1. Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем: учебник для студентов вузов . — СПб.: Питер, 2019. — 504 с.
2. Макарова Н. В., Новиков В. А. Информационные технологии: учебник для студентов вузов . — М.: Финансы и статистика, 2018. — 552 с.
3. Подбельский В. В. Язык С++. Учебное пособие . — М.: Финансы и статистика, 2019. — 672 с.

**4. Интернет-ресурсы:**

1. GitHub — платформа для хранения и совместной разработки проектов.  
   [Электронный ресурс] — URL: [https://github.com](https://github.com/)
2. О системе контроля версий Git.  
   [Электронный ресурс] — URL: <https://git-scm.com/book/ru/v2/>
3. Курс по работе с Git и GitHub.  
   [Электронный ресурс] — URL: <https://smartiqa.ru/courses/git>
4. Лавришева Е.М., Петрухин В.А. Методы и средства инженерии программного обеспечения . Единое окно доступа к образовательным ресурсам.  
   [Электронный ресурс] — URL: <http://window.edu.ru/catalog/pdf2txt/699/41699/18857>
5. Моделирование и проектирование программного обеспечения.  
   [Электронный ресурс] — URL: <https://present5.com/modelirovanie-i-proektirovanie-programmnogo-obespecheniya-lekciya-8-realizaciya/>
6. Стандарты документирования программных средств.  
   [Электронный ресурс] — URL: <http://www.slideserve.com/nigeltillman/6468916>
7. Управление тестированием программного обеспечения.  
   [Электронный ресурс] — URL: <http://tmguru.ru/baza-znanij/>
8. Microsoft Visual Studio — официальный сайт.  
   [Электронный ресурс] — URL: <https://visualstudio.microsoft.com/>