Министерство просвещения Приднестровской Молдавской Республики

Государственное образовательное учреждение среднего профессионального образования «Тираспольский техникум информатики и права»

ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

**«Разработка консольного приложения нахождения цифрового корня натурального числа»**

по учебной дисциплине «Информатика»

по специальности 09.02.07 «Информационные системы и программирование»

Выполнил Палий А.А.

обучающийся I курса

специальность 09.02.07 «Информационные системы и программирование»

Руководитель Шандригоз Наталья Николаевна

Преподаватель информатики высшей квалификационной категории

Допущен к защите

«\_\_\_»\_\_\_\_\_\_\_202\_\_ г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тирасполь 20\_\_

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| ВВЕДЕНИЕ | **3** |
| 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНДИВИДУАЛЬНОГО ПРОЕКТА |  |
| 1.1. Общие сведения о цифровом корне |  |
| 1.2. История цифрового корняо |  |
| 1.3. Интересные факты о цифровом корне |  |
| 1. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ИНДИВИДУАЛЬНОГО ПРОЕКТА |  |
| * 1. Постановка задачи |  |
| * 1. Программно-техническое обеспечение решения задачи |  |
| * 1. Описание программного кода |  |
| * 1. Тестирование приложения |  |
| ЗАКЛЮЧЕНИЕ |  |
| СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ |  |
| ПРИЛОЖЕНИЯ |  |

**ВВЕДЕНИЕ**

Понятие цифрового корня натурального числа остается актуальным и востребованным в различных областях, от математики до криптографии и образования.

Актуальность темы проектной работы состоит в изучении понятия цифрового корня натурального числа и проявлении в аспектах, которые охватывают как теоретическую, так и практическую стороны математики и других дисциплин.

Теоретическая значимость проектной работы заключается в изучении цифрового корня за рамками простой арифметической операции, охватывая широкий спектр математических и междисциплинарных исследований.

Практическая целесообразность работы состоит в следующем**:**

* *изучении теоретических материалов о определении цифрового корня, его математических свойств, исторических и практических аспектов*
* *разработке программного кода для нахождения цифрового корня из натурального числа*

Цель проектной работы – изучить структуру, программный код и теоретические сведения о цифровом корне из натурального числа, а также разработать программный код на C#

Задачи исследования:

*1. Определить-что такое цифровой корень из натурального числа  
 -определение цифрового корня*

*2. Раскрыть* теоретические аспекты вычисления цифрового корня, включая классический итеративный подход и связь с операцией по модулю 9

*3. Разработать* программное средство на языке C# для вычисления цифрового корня и провести его тестирование в программе Visual Studio 2022

*4. «Исследовать* результаты работы программы, оценить эффективность и корректность алгоритма, *и дать рекомендации* по дальнейшему совершенствованию или применению разработанного решения.*»*

Предмет исследования – Цифровой корень натурального числа   
Объект исследования–программа для нахождения цифрового корня из натурального числа

**1.ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНДИВИДУАЛЬНОГО ПРОЕКТА**

* 1. **Общие сведения о цифровом корне**

Древние цивилизации

1. Древний Египет и Месопотамия: В этих культурах использовались различные системы счисления для ведения записей и расчетов. Хотя термина "цифровой корень" не существовало, идеи о суммировании и редукции чисел были известны.
2. Древняя Индия: Индийские математики, такие как Брахмагупта, занимались числовыми свойствами и разработали методы для работы с числами, включая их сокращение и преобразование.

Средние века

1. Арабская математика: Арабы перенесли и развили идеи, пришедшие из Индии, включая систему десятичного счисления. Они использовали различные методы для манипуляции с числами, включая суммирование цифр.

Современное время

1. Математика 19 века: Концепция цифрового корня начала обретать более формальное определение. Математики начали исследовать свойства чисел и их связи, включая такие операции, как нахождение цифрового корня.
2. Теория чисел: Цифровой корень стал частью более широкой области теории чисел, где исследуются свойства чисел и их взаимосвязи. Важным открытием стало то, что цифровой корень числа может быть найден с помощью простого правила: остатка от деления на 9 (для чисел, не равных нулю).

Использование в различных областях

1. Современные приложения: Цифровой корень используется в различных областях, включая криптографию, компьютерные науки и даже в некоторых играх и головоломках. Он также может быть полезен в проверке правильности вычислений.

**1.3.** Интересные факты о цифровом корне:  
**1.Связь с арифметикой по модулю 9:**Цифровой корень натурального числа можно вычислить с помощью операции по модулю 9. Если число не равно нулю, его цифровой корень равен 1+(n−1)mod  91 + (n - 1) \mod 91+(n−1)mod9.

**2.Метод «выброса девяток»:**  
В древности использовался метод «casting out nines» для проверки правильности арифметических вычислений. Если сумма цифр результата совпадала с цифровым корнем исходных чисел, вычисление считалось верным.  
**3.Инвариантность при перестановке цифр:**  
Независимо от порядка цифр в числе, его цифровой корень остается неизменным. Это связано с аддитивными свойствами цифр и объясняет, почему цифры можно «перемешивать», не меняя итоговую сумму по модулю 9.  
**4.Математические иллюзии:**  
Многие фокусы и головоломки основаны на неизменности цифрового корня, демонстрируя, как повторное сложение цифр всегда приводит к предсказуемому результату независимо от порядка их расположения.

**2.ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ИНДИВИДУАЛЬНОГО ПРОЕКТА**

2.1. Постановка задачи   
Постановка задачи включает в себя:  
 Анализ задачи:

* Основная цель приложения — расчет цифрового корня натурального числа с использованием быстрых и эффективных алгоритмов.

 Формулировка условий:

* Пользователь вводит натуральное число, для которого приложение вычисляет его цифровой корень.

 Связь между данными:

* Входные данные: натуральное число, предоставленное пользователем.
* Выходные данные: цифровой корень введенного числа.

 Описание входных и выходных данных:

* Ввод данных:
  + Пользователь вводит положительное целое число.
* Обработка данных:
  + Проверка правильности ввода (число должно быть натуральным).
* Вывод данных  
   Отображение результата – цифрового корня,полученного из введенного числа

Цифровой корень натурального числа можно получить следующим образом:

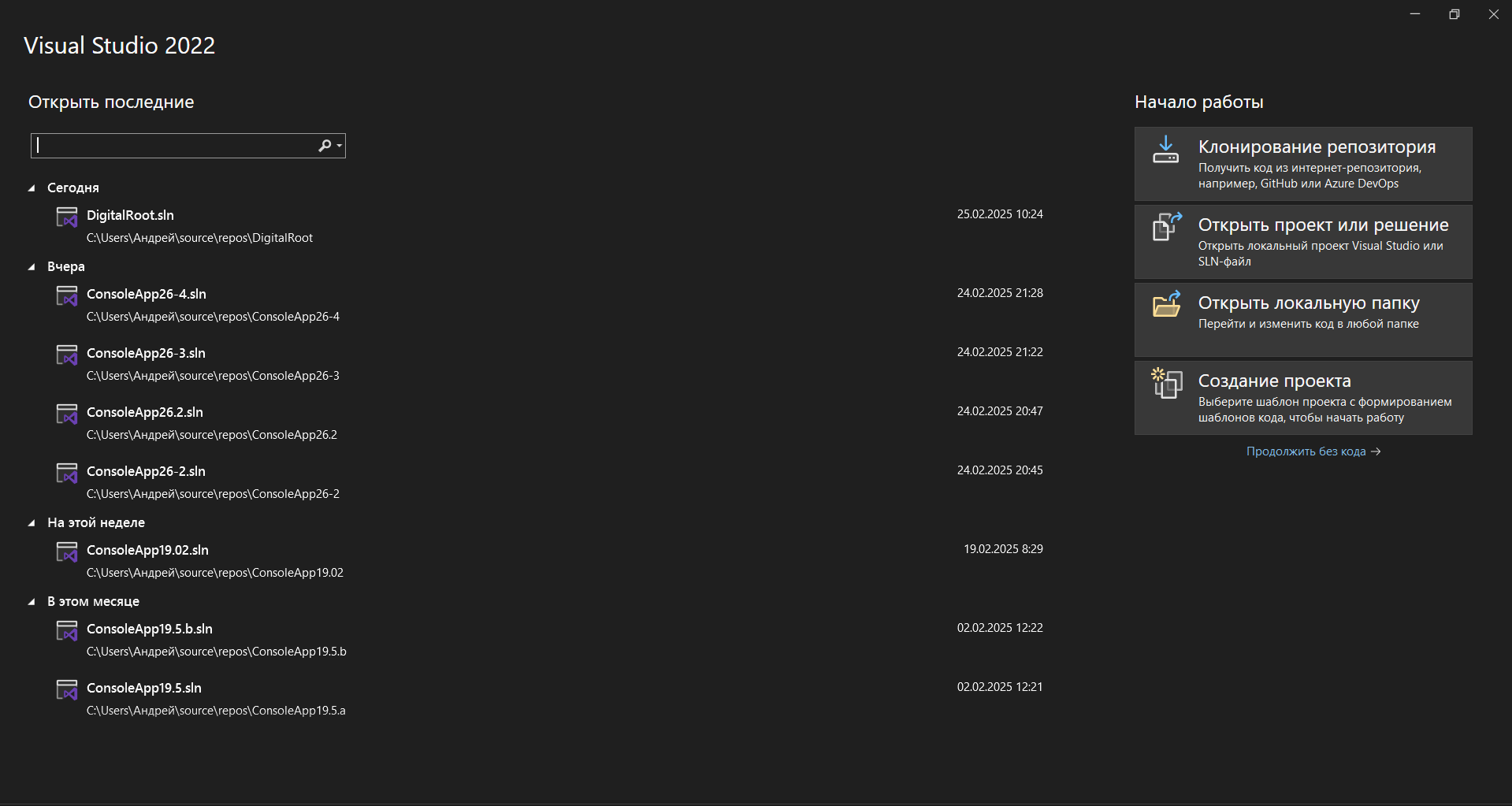
1. сложить все цифры числа;
2. если сумма состоит из более, чем одной цифры, повторить процесс, пока не останется однозначное число.

Пример. Для числа 9875:

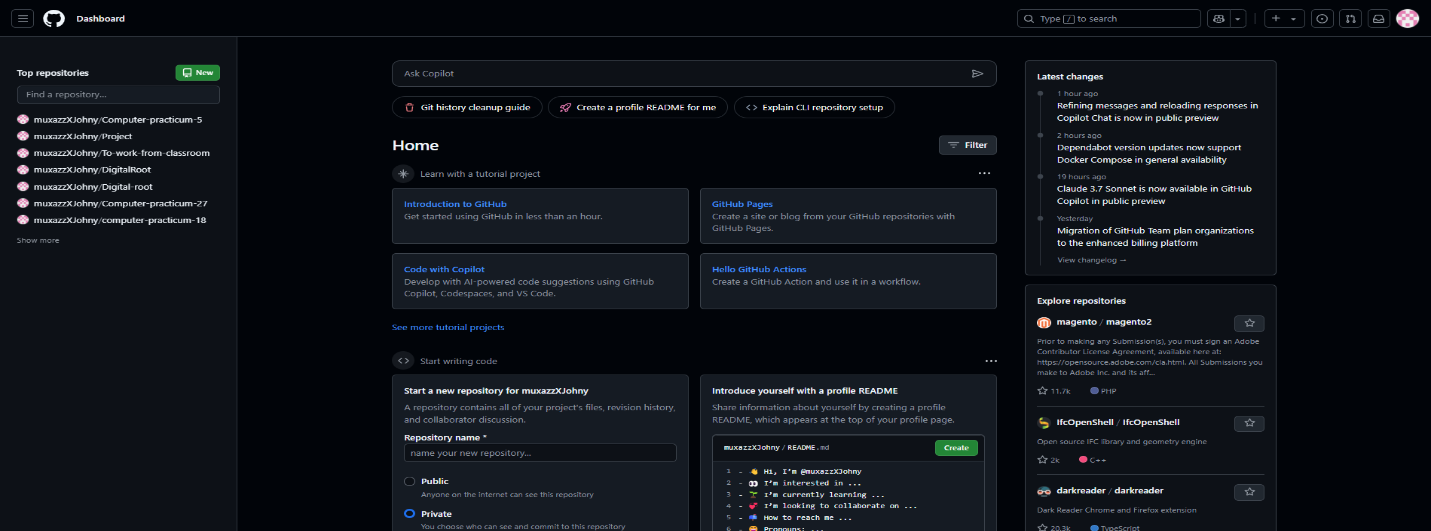
1. 9 + 8 + 7 + 5 = 29
2. 2 + 9 = 11
3. 1 + 1 = 2

Цифровой корень = 2

* 1. **Программно-техническое обеспечение решения задачи**

Для разработки программного кода использована интегрированная среда разработки (IDE) – MS Visual Studio 2022  
Вот так выглядит общее окно VS 2022  
Visual Studio 2022 — это интегрированная среда разработки (IDE), созданная компанией Microsoft. Она предоставляет широкий набор инструментов и функций для разработки программного обеспечения на различных платформах, включая Windows, macOS, Linux, Android и iOS.

Visual Studio 2022 поддерживает множество языков программирования, таких как C++, C#, Python, JavaScript и другие. Она предоставляет возможности для редактирования кода, отладки, тестирования, создания пользовательского интерфейса и развертывания приложений.

С помощью Visual Studio 2022 можно разрабатывать различные типы приложений, включая веб-приложения, мобильные приложения, настольные приложения, облачные приложения и игры. Она также предоставляет интеграцию с другими инструментами и сервисами Microsoft, такими как Azure DevOps и GitHub.  
Общий вид GitHub  
Для хранения и управления кодом был использован репозиторий GitHub:  
  
GitHub- это веб-платформа, предоставляющая разработчикам возможность хранить свой код и совместно работать над проектами. Она основана на системе контроля версий Git, что позволяет отслеживать изменения в коде и возвращаться к предыдущим версиям. GitHub также предоставляет различные инструменты для совместной работы, такие как возможность создавать задачи, обсуждать изменения и просматривать чужой код.

GitHub можно назвать социальной сетью для разработчиков, поскольку он позволяет общаться с другими программистами, делиться своими проектами и находить единомышленников. Платформа также предоставляет возможность создавать собственные проекты с открытым исходным кодом, что способствует развитию сообщества разработчиков.

GitHub является незаменимым инструментом для любого программиста, который хочет работать над проектами совместно с другими людьми, а также делиться своими достижениями с миром. Он предоставляет удобный интерфейс для работы с Git, а также множество дополнительных функций, которые делают процесс разработки более эффективным и удобным.  
  
Для разработки консольного приложения использовался язык программирования C#.

Язык C# - это объектно-ориентированный язык программирования, разработанный компанией Microsoft, чтобы создавать приложения для Windows.

#### Технические требования:

1. Программа должна принимать натуральное число от пользователя.
2. Программа должна выводить цифровой корень этого числа.
3. Обработка ошибок (например, если введено не натуральное число).
4. Код должен быть хорошо структурирован и документирован.

**2.3 Описание программного кода**

using System;  
using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DigitalRoot

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Введите натуральное число: ");

string input = Console.ReadLine();

if (long.TryParse(input, out long number) && number > 0)

{

int digitalRoot = CalculateDigitalRoot(number);

Console.WriteLine($"Цифровой корень числа {number} равен: {digitalRoot}");

}

else

{

Console.WriteLine("Пожалуйста, введите корректное натуральное число.");

}

Console.ReadKey();

}

static int CalculateDigitalRoot(long number)

{

while (number >= 10) // Продолжаем, пока число не станет однозначным

{

number = SumOfDigits(number); // Суммируем цифры

}

return (int)number; // Возвращаем одноцифровое число

}

public static int DigitalRootNumber2(int number)

{

return (number - 1) % 9 + 1;

}

static long SumOfDigits(long number)

{

long sum = 0;

while (number > 0)

{

sum += number % 10; // Добавляем последнюю цифру к сумме

number /= 10; // Убираем последнюю цифру

}

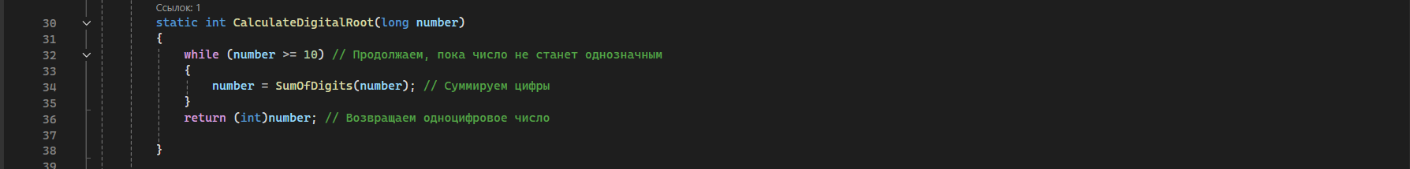
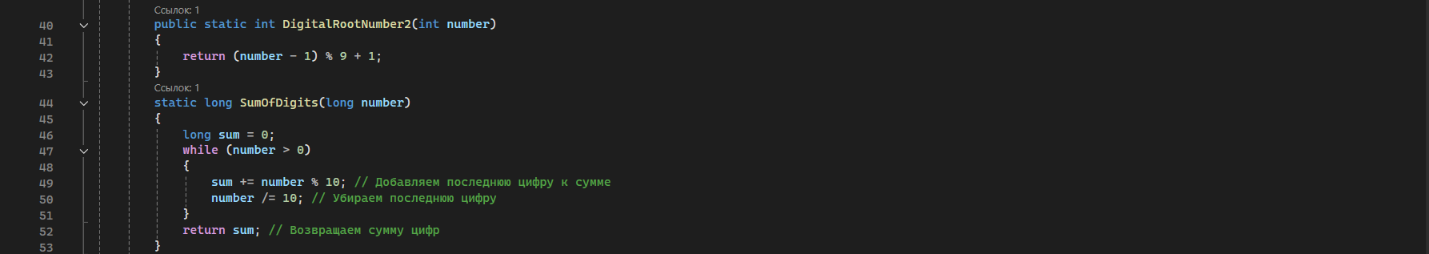
return sum; // Возвращаем сумму цифр

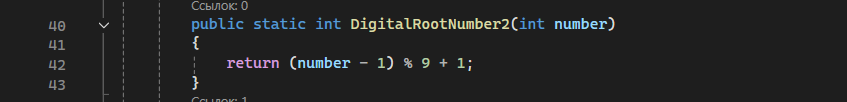
}

}

}

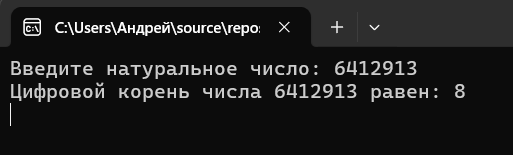
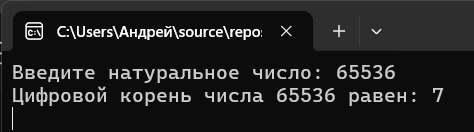
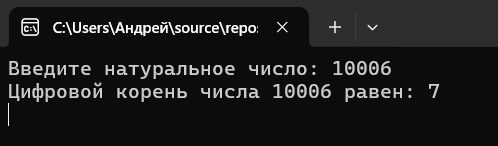
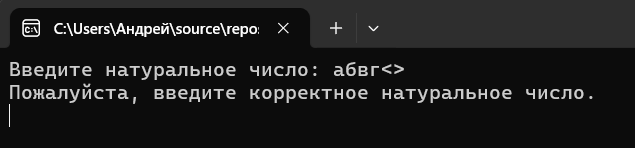
### Описание кода:

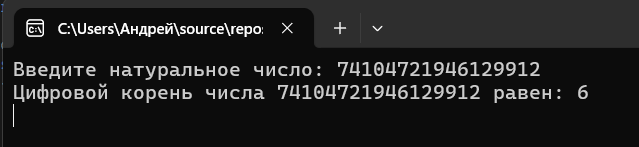
1. **Main:** Основной метод, который запрашивает у пользователя ввод числа и проверяет его на корректность.С помощью long.TryParse
2. **CalculateDigitalRoot:** Метод, который вычисляет цифровой корень, используя цикл,   
   пока число не станет однозначным.
3. **SumOfDigits:** Метод, который вычисляет сумму цифр числа.

4.DigitalRootNumber2:Метод,который вычисляей цифровой корень с помощью формулы  


* 1. **Тестирование приложения**

Тестирование приложения – это процесс проверки приложения на соответствие требованиям, заданным в начале разработки.Его проводят,что бы убедиться,что программа удобна для пользователя,а все функции работают так, как задумано

** Вводим натуральное число 65536  
  
  
  
   
  
  
Был просчитан правильный результат:7  
  
Далее я ввел число 6412913  
  
  
  
  
  
  
  
Результатом оказалось число 8  
  
Вводим число 10006**  
Получаем результат 7;можно заметить,что у совершенно разных чисел одинаковый цифровой корень  
Если ввести некорректные символы(не натуральное число)  
Программа уведомит об этом и попросит ввести корректное натуральное число

Если ввести очень большое число, программа не сломается ,а просчитает цифровой корень

**Заключение**   
  
В процессе разработки консольного приложения для вычисления цифрового корня натурального числа был достигнут ряд ключевых результатов:  
  
1.Была изучена математическая база цифрового корня: рассмотрены определения, свойства натуральных чисел, а также алгоритмы последовательного суммирования цифр и использование модульной арифметики для оптимизации вычислений.  
   
2. Приложение обеспечивает ввод данных, позволяя пользователю легко и быстро вводить натуральное число. Важной частью реализации является валидация вводимых данных, что позволяет предотвратить ошибки и гарантировать, что только корректные значения будут обработаны.  
  
3.**Были использованы современные алгоритмы,** что обеспечивает высокую производительность приложения и скорость обработки данных и что позволяет обеспечивать высокую производительность даже при работе с большими числами.  
  
4. **Обработка ошибок:** Внедрена система информирования пользователя о некорректном вводе, что улучшает взаимодействие с приложением и способствует более легкому решению возможных проблем.  
 В процессе разработки консольного приложения для вычисления цифрового корня натурального числа был рассмотрен комплексный подход к решению поставленной задачи. Основной акцент был сделан на создании удобного и эффективного инструмента, который не только отвечает требованиям пользователей, но и демонстрирует высокую производительность при выполнении вычислений. Таким образом, консольное приложение не только успешно выполняет свою основную задачу, но и предлагает пользователям дополнительные возможности для понимания и освоения процесса вычисления цифрового корня. В будущем проект может быть расширен новыми функциями и улучшениями

**СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ:**1. Семакин И.Г. Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 1/ И.Г.Семакин, Т.Ю.Шеина, Л.В.Шестакова. – М.:БИНОМ. Лаборатория знаний, 2014. – 184 с.

2. Семакин И.Г. Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 2/ И.Г.Семакин, Т.Ю.Шеина, Л.В.Шестакова. – М.:БИНОМ. Лаборатория знаний, 2014. – 232 с.

3. Семакин И.Г. Информатика. Углубленный уровень: учебник для 11 класса: в 2 ч. Ч. 1/ И.Г.Семакин, Е.К.Хеннер, Л.В.Шестакова. – М.:БИНОМ. Лаборатория знаний, 2014. – 176 с.

4. Семакин И.Г. Информатика. Углубленный уровень: учебник для 11 класса: в 2 ч. Ч. 2/ И.Г.Семакин, Е.К.Хеннер, Л.В.Шестакова. – М.:БИНОМ. Лаборатория знаний, 2014. – 216 с.

5. Мартынов H. H. C# для начинающих - М.: КУДИЦ-ПРЕСС, 2007. - 272 с.

6. Платформа .NET Framework. Язык C#. : учеб.-метод. пособие : Сост.: Якубович Д. А., Еропова Е. С. / Мин-во образования и науки Рос. Федерации, ФГБОУ ВО «Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых». – Владимир : Издательство «Шерлок-пресс», 2018, Ч. 1. – 48 с.

7. Залогова Л. А. Основы объектно\_ориентированного программирования на базе языка С# : учебное пособие для СПО / Л. А. Залогова. — Санкт\_Петербург : Лань, 2020. — 192 с.