Министерство просвещения Приднестровской Молдавской Республики

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

**ИНДИВИДУАЛЬНЫЙ ПРОЕКТ**

«Алгоритм Евклида»

по учебной дисциплине «Информатика»

по специальности 09.02.07 «Информационные системы и программирование»

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_ Д.Ю.Дидур

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 2024

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| ВВЕДЕНИЕ | 3 |
| 1. Теоретические аспекты проектной работы | 4 |
| * 1. Исторические сведения | 4 |
| 1.2. Алгоритм Евклида | 6 |
| 1.3. Примеры использование алгоритма Евклида | 7 |
| 2. Практическая реализация индивидуального проекта | 9 |
| 2.1 Постановка задачи | 9 |
| 2.2 Программно-техническое обеспечение решения задачи | 9 |
| 2.3. Тестирование приложения | 13 |
| ЗАКЛЮЧЕНИЕ | 14 |
| СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ | 15 |

**ВВЕДЕНИЕ**

Актуальность данного проекта также заключается в том, что рассматривается реализация алгоритма Евклида на языке программирования C# для вычисления наибольшего общего делителя.

В мире компьютерной науки и математики существует множество алгоритмов, которые играют ключевую роль в различных областях. Один из таких фундаментальных алгоритмов - алгоритм Евклида.

Теоретическая значимость проектной работы заключается в изучении алгоритма Евклида и практическом применении при составлении программного кода с использованием цикла и условных конструкций.

Цель проекта – автоматизация вычисления наибольшего общего делителя с помощью алгоритма Евклида для консольного приложения на языке программирования C#.

Задачи:

1. Актуализировать теоретические знания по алгоритму Евклида.
2. Разработать программный код вычисления НОД(Наибольший общий делитель) с использованием цикла и условных конструкций на языке C#
3. Протестировать программный код
4. Создать репозиторий для хранения программного кода.

Предмет исследования – разработка программного кода с использованием условной конструкции для вычисления площади треугольника.

Объект исследования – Наибольший общий делитель

Период исследования – февраль 2024 года.

1. **ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ПРОЕКТНОЙ РАБОТЫ**
   1. **Исторические сведения**

Евклид и его время: Алгоритм Евклида был придуман в древней Греции Евклидом, который жил приблизительно в III веке до нашей эры. Евклид был известным математиком и геометром, и его работы влияли на развитие математики на протяжении многих столетий.

«Элементы»: Основные принципы алгоритма Евклида были описаны в его знаменитом труде «Элементы». В «Элементах» Евклид изложил основы геометрии и алгебры и ввел множество важных математических концепций, включая алгоритм нахождения наибольшего общего делителя (НОД) двух чисел.

Нахождение НОД: Основная задача алгоритма Евклида состоит в нахождении наибольшего общего делителя (НОД) двух целых чисел. Этот алгоритм основан на принципе последовательного вычитания, при котором из большего числа вычитается меньшее, пока они не станут равными. Затем полученное число вычитается из исходного, и так продолжается до тех пор, пока не будет получено нулевое значение. Последнее ненулевое число будет НОДом.

Применение в математике и криптографии: Алгоритм Евклида широко используется в различных областях математики, включая алгебру, теорию чисел и криптографию. Например, в криптографии НОД используется для генерации ключей шифрования и проверки их прочности.

Историческое значение: Алгоритм Евклида имеет огромное историческое значение, так как он был одним из первых методов для решения проблемы нахождения НОДа и оказал огромное влияние на развитие математики в целом. Его простота и эффективность сделали его фундаментальным инструментом в изучении алгебры и теории чисел.

* 1. **Формула Алгоритма Евклида**

Алгоритм Евклида – это метод нахождения наибольшего общего делителя (НОД) двух целых чисел. Он основан на принципе последовательных вычитаний и имеет следующую формулу:

Gcd(a,b) = Gcg(b,a mod b)

Где gcd(a,b) обозначает НОД aи *b,a* mod b – это остаток от деления a на b.

Алгоритм начинается с двух исходных чисел *a* и *b.* Затем *b* заменяется на *a* *mod b* и процесс повторяется до тех пор, пока *b* не станет равным нулю. Последнее значение *a* будет *НОДом* исходных чисел *a* и *b.*

*Этот метод довольно эффективен и прост в реализации, что делает его одним из ключевых инструментов в алгебре, теории чисел и криптографии*

* 1. **Примеры использования алгоритма Евклида**

1. **Нахождение НОД двух чисел**: Предположим, нам нужно найти НОД чисел 48 и 18. Мы можем использовать алгоритм Евклида следующим образом:
   * *gcd(48,18)=gcd(18,48mod  18)=gcd(18,12)gcd(48,18)=gcd(18,48mod18)=gcd(18,12)*
   * *gcd(18,12)=gcd(12,18mod  12)=gcd(12,6)gcd(18,12)=gcd(12,18mod12)=gcd(12,6)*
   * *gcd(12,6)=gcd(6,12mod  6)=gcd(6,0)gcd(12,6)=gcd(6,12mod6)=gcd(6,0)* Последнее значение 6 является НОДом чисел 48 и 18.
2. **Проверка взаимной простоты**: Алгоритм Евклида также может использоваться для определения того, являются ли два числа взаимно простыми (т.е. имеют ли они НОД равный 1). Например, для чисел 35 и 12:
   * *gcd(35,12)=gcd(12,35mod  12)=gcd(12,11)=1gcd(35,12)=gcd(12,35mod12)=gcd(12,11)=1* Так как НОД чисел 35 и 12 равен 1, они взаимно просты.
3. **Расширенный алгоритм Евклида для нахождения обратного элемента по модулю**: Алгоритм Евклида также может быть расширен для нахождения обратного элемента по модулю. Например, мы хотим найти *x* такое, что *3x ≡1(mod7)3x≡1(mod7)*:
   * Мы начинаем с применения расширенного алгоритма Евклида для нахождения НОД(3, 7) и коэффициентов Безу.
   * После этого, используя коэффициенты Безу, мы можем найти обратный элемент *x*, удовлетворяющий условию *3x≡1(mod7)3x≡1(mod7).*

Эти примеры демонстрируют различные способы использования алгоритма Евклида для решения различных задач в теории чисел и алгебре.

**2. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ИНДИВИДУАЛЬНОГО ПРОЕКТА**

**2.1 Постановка задачи**

Пример практической реализации алгоритма Евклида для нахождения НОДа чисел 48 и 18:

1. **Шаг 1: Начальные числа**: Возьмем два исходных числа: *a=48 и b=18.*
2. **Шаг 2: Итерации алгоритма**: Применяем алгоритм Евклида:
   * *gcd(48,18)=gcd(18,48mod  18)=gcd(18,12)gcd(48,18)=gcd(18,48mod18)=gcd(18,12)*
   * *gcd(18,12)=gcd(12,18mod  12)=gcd(12,6)gcd(18,12)=gcd(12,18mod12)=gcd(12,6)*
   * *gcd(12,6)=gcd(6,12mod  6)=gcd(6,0)gcd(12,6)=gcd(6,12mod6)=gcd(6,0)*
3. **Шаг 3: Наибольший общий делитель (НОД)**: Последнее ненулевое число в паре будет НОДом исходных чисел:
   * НОД(48, 18) = 6.

Таким образом, применяя алгоритм Евклида к числам 48 и 18, мы получаем, что их НОД равен 6. Это и является практической реализацией алгоритма Евклида для данных чисел. Вы можете представить этот пример в вашем проекте, просто описав процесс применения алгоритма к конкретным числам без предоставления кода.

**2.2 Программно-техническое обеспечение решения задачи**

Для разработки программного кода использована интегрированная среда разработки (IDE) – MS Visual Studio 2010/2022 — это лучшая интегрированная среда разработки для создания многофункциональных, привлекательных кроссплатформенных приложений для Windows, Mac, Linux, iOS и Android, так же в ходе работы использовался язык программирования C#.

Язык C# - это объектно-ориентированный язык программирования, разработанный компанией Microsoft. Изначально его создавали для проектов под Windows, но теперь это по-настоящему универсальный язык: на нём пишут игры, десктопные приложения, веб-сервисы, нейросети и даже графику для метавселенных.

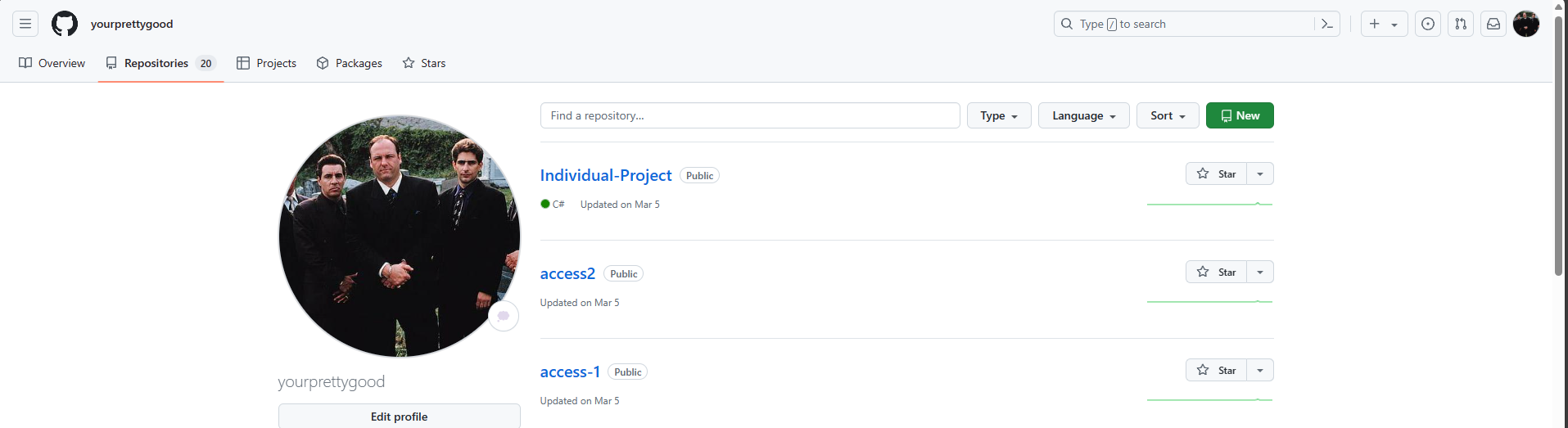
 Для хранения программы использован репозиторий, созданный в сервисе Git Hub (Рисунок 1). Система позволяет создавать и развивать ветви проекта, откатываться к предыдущим версиям, клонировать.

Рисунок 1.

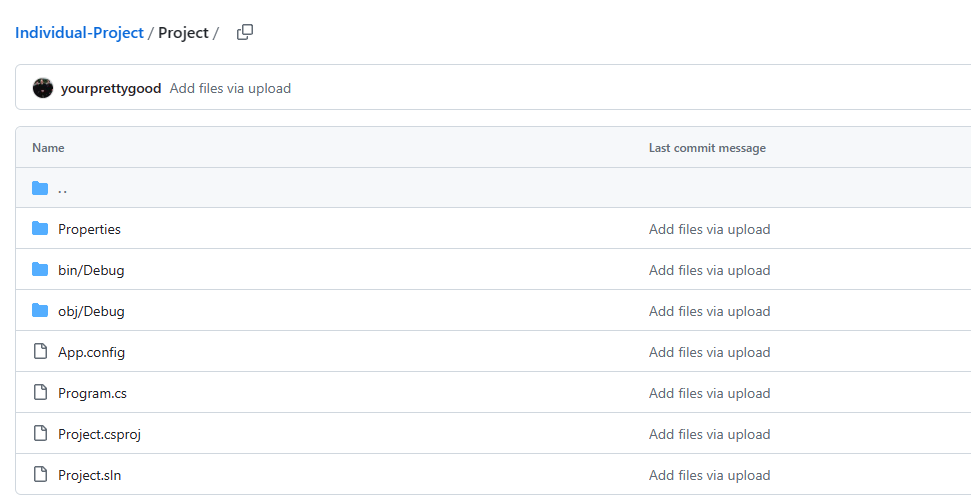


Рисунок 2.

Код индивидуального проекта сохраненный в репозитории в ветке «Project»

Программный код вычисления НОД, путем алгоритма Евклида:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Project

{

internal class Program

{

static void Main(string[] args)

{

Console.ForegroundColor =ConsoleColor.Green;

Console.Title = "Дидур Даниил, Индивидуальный проект на тему: Алгоритм Евклида ";

Console.Write("Введите 1-ое число: ");//Запрашиваем у пользователя 1-ое число

int a = int.Parse(Console.ReadLine());//Объявление переменной, которая будет хранить 1-ое число пользователя

Console.Write("Введите 2-ое число: ");//Запрашиваем у пользователя 2-ое число

int b = int.Parse(Console.ReadLine());//Объявление переменной, которая будет хранить 2-ое число пользователя

while (a != b) //Создаем цикл while, с условием, что 1-ое число будет меньше 2-ого

{

if (a > b)//В этом цикле проверяем, если 1-ое число больше 2-ого.

{

a = a - b;//Операция, которая будет выполняться при выполнения условия

}

else

{

b = b - a;//Операция, которая будет выполняться при невыполнении условия выше

}

}

Console.WriteLine($"Нод = {b} ");//Выводим в консоль значение НОД'а

Console.ReadLine();

}

}

}

**2.3. Тестирование приложения**

Тестирование программного обеспечения – длительный и обширный процесс.

Тестирование – проверка программного обеспечения которая показывает, соответствует ли оно ожиданиям разработчиков и правильно ли работает.

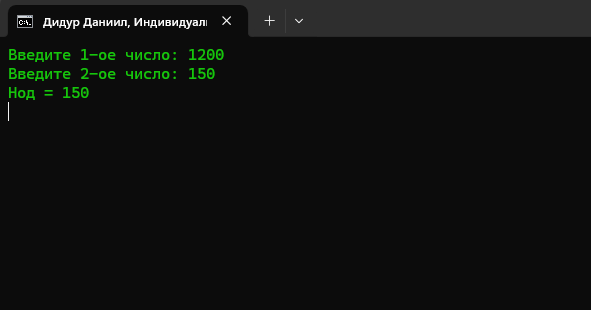


Рисунок 3. Пример 1.

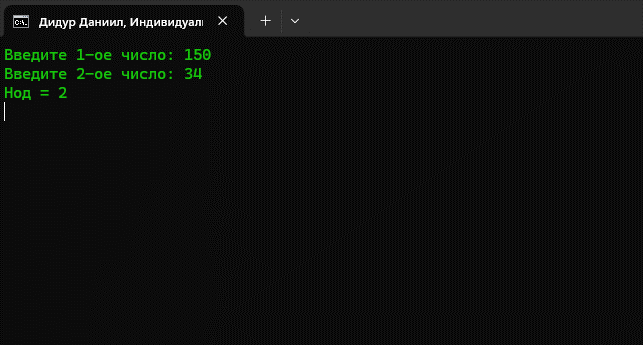
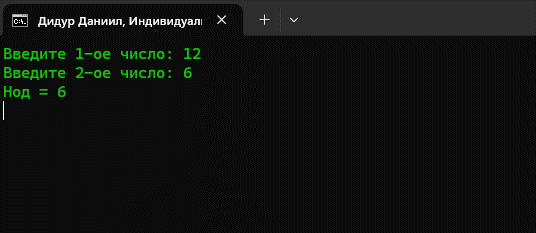
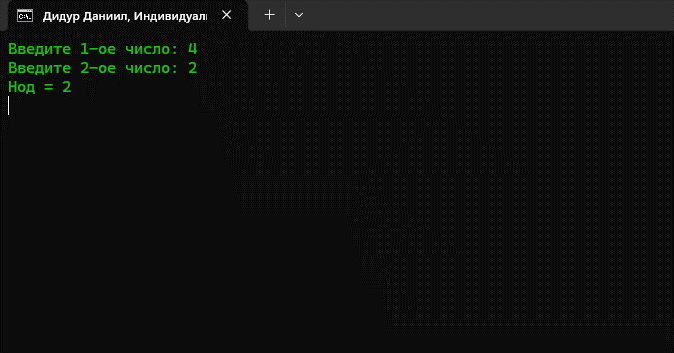


Рисунок 4. Пример 2.

 Рисунок 5. Пример 3.

 Рисунок 6. Пример 4.

**ЗАКЛЮЧЕНИЕ**

Алгебра – интересный предмет. При решении задач можно использовать различные методы и формулы, решая одну и ту же задачу можно пойти разными путями. Решение задач по алгебре играет важную роль в формировании математической культуры, а также очень хорошо развивает логическое мышление.

Алгоритм Евклида достаточно прост для понимания и реализации. Это делает его идеальным выбором для широкого спектра проектов, включая учебные задания, программирование и исследования.

**Высокая эффективность**: Алгоритм Евклида обладает хорошей скоростью работы, особенно при работе с большими числами. Он может быстро находить наибольший общий делитель даже для очень больших чисел.

**Широкое применение**: НОД является важной математической операцией, используемой в различных областях, включая криптографию, теорию чисел, алгебру и компьютерные науки. Алгоритм Евклида обеспечивает простой и эффективный способ вычисления НОДа, что делает его очень полезным во многих приложениях.

**Поддержка различных типов данных**: Алгоритм Евклида может быть легко адаптирован для работы с различными типами данных, включая целые числа, вещественные числа и даже полиномы.

**Историческое значение**: Алгоритм Евклида имеет богатую историю и является одним из фундаментальных методов в математике. Его использование в проекте может подчеркнуть важность классических математических концепций и их применимость в современных технологиях.

В ходе выполнения проектной работы, решены следующие задачи:

* Изучена история алгоритма Евклида
* Проанализирован теоретический материал и исследована формула алгоритма Евклида для вычисления НОДа
* Разработан программный код для вычисления НОДа путем алгоритма Евклида
* Приложение протестировано для различных чисел

**СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Euclidean Algorithm -- from Wolfram MathWorld [https://mathworld.wolfram.com/EuclideanAlgorithm.html#:~:text=The%20Euclidean%20algorithm%20is%20an,(Bach%20and%20Shallit%201996).&text=divisor%20of%20and%20%2C-,.,approximate%20relations%20(Ferguson%20et%20al.]
2. Алгоритм Евклида - нахождение наибольшего общего делителя - Python (younglinux.info)[ https://younglinux.info/algorithm/euclidean]
3. Euclidean Algorithm (mathsisfun.com)[ https://www.mathsisfun.com/numbers/euclidean-algo.html#:~:text=3%20R%205-,The%20Euclidean%20Algorithm,the%20size%20of%20the%20numbers.&text=In%20other%20words%2C%20divide%20a,a)%20and%20the%20remainder%20r.]
4. Алгоритм Евклида — Википедия (wikipedia.org)[ https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4%D0%B0]
5. Алгоритм Евклида (scienceland.info)[ https://scienceland.info/algebra8/euclid-algorithm]