Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Лабораторная работа № 3**

« Основы теории чисел и их использование в криптографии»

Выполнил:

Студент: Герман А.Е.

ФИТ 3 курса 5 группы

Преподаватель: Савельева М.

Минск 2024

## **Описание приложения**

Приложение написано на языке программирования C# и позволяет провести расчет НОД двух чисел, а также найти все простые числа в выбранном промежутке.

Теория чисел или высшая арифметика – раздел математики, изучающий  
натуральные числа и иные похожие величины. В зависимости от используемых методов в теории чисел рассматривают несколько направлений. Нас будут интересовать вопросы делимости целых чисел, вычисления наибольшегообщего делителя (НОД).

## **Методика выполнения расчетов**

В данной лабораторной работе была поставлена цель создания приложения, позволяющее произвести расчёт НОД двух либо трёх чисел, а также выполнить поиск простых чисел из диапазона. На листинге 2.1 представлен класс, реализующий данную функциональность.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace crypt\_lab\_3  {  internal class NumCrypt  {  public static int CalculateLeastCommonDivisor(int x, int y)  {  while (x != 0 && y != 0)  {  if (x > y)  {  x -= y;  }  else  {  y -= x;  }  }  return Math.Max(x, y);  }  public static int CalculateLeastCommonDivisor(int a, int b, int c)  {  int Nod = Math.Min(a, Math.Min(b, c));  for (; Nod > 1; Nod--)  {  if (a % Nod == 0 && b % Nod == 0 && c % Nod == 0)  break;  }  return Nod;  }  private static bool isPrime(int n)  {  if (n > 1)  {  for (int i = 2; i < n; i++)  {  if (n % i == 0) return false; // если n делится без остатка на i - возвращаем false (число не простое)  }  return true; // если программа дошла до данного оператора, то возвращаем true (число простое) - проверка пройдена  }  else return false;  }  public static void FindPrimeNumberInInterval(int m, int n)  {  int counter = 0;  if (n < m)  {  Console.WriteLine("Неверный промежуток");  }  Console.Write($"Простые числа интервала [{m},{n}]: ");  for (int i = m; i <= n; i++)  {  if (isPrime(i))  {  Console.Write(i.ToString() + " ");  counter++;  }  }  Console.WriteLine();  Console.WriteLine($"Количество простых чисел: {counter}");  }  }  } |

Листинг 2.1 – Класс, реализующий необходимый функционал

Метод **CalculateLeastCommonDivisor** используется для получения НОД двух или чисел путём исполнения алгоритма Евклида в двух вариациях с помощью переопределения. Метод **isPrime** осуществляет проверку на то, является ли переданное в него в качестве параметра число простым. Метод **FindPrimeNumberInInterval** находит все простые числа в диапазоне, границы которого переданы в него в качестве двух параметров.

## **Результаты работы приложения**

Для выполнения расчетов достаточно необходимо запустить приложение, выбрать необходимый пункт задания и ввести числа. Рисунки 3.1 - 3.4 показывают необходимые расчеты и вызовы методов, требуемые в данной лабораторной работе.

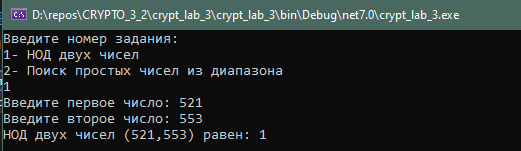


Рисунок 3.1 – Результат работы метода **CalculateLeastCommonDivisor** с 2 параметрами

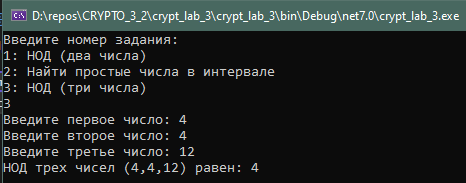
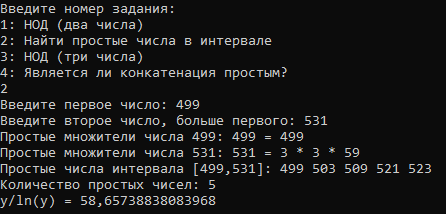


Рисунок 3.2 – Результат работы метода **CalculateLeastCommonDivisor** с 3 параметрами



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **499** | ~~500~~ | ~~501~~ | ~~502~~ | **503** | ~~504~~ | ~~505~~ | ~~506~~ | ~~507~~ | ~~508~~ |
| **509** | ~~510~~ | ~~511~~ | ~~512~~ | ~~513~~ | ~~514~~ | ~~515~~ | ~~516~~ | ~~517~~ | ~~518~~ |
| ~~519~~ | ~~520~~ | **521** | ~~522~~ | **523** | ~~524~~ | ~~525~~ | ~~526~~ | ~~527~~ | ~~528~~ |
| ~~529~~ | ~~530~~ | ~~531~~ |

Рисунок 3.3 – Результат работы методе **FindPrimeNumberInInterval**

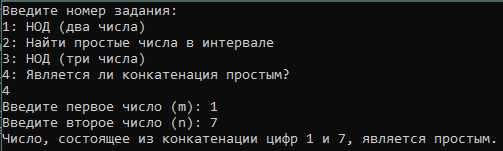


Рисунок 3.4 – Результат работы методе **IsConcatenationPrime**

## **Вывод**

В ходе изучения теоретических материалов лабораторной работы и выполнения её практической части были изучены основы теории чисел, используемые в криптографии. Изучены основные свойства простых и составных чисел, взаимно простых чисел, а также признаки делимости чисел. На практике полученные знания были закреплены через создание собственного приложения, реализующего функционал вычисления НОД двух и трёх чисел и получения списка простых чисел в заданном диапазоне.