**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчет по практическим работам**

**МДК 03.02**

**Инструментальные средства разработки программного обеспечения**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ПКС-91 | |  | 31.01.23 | *Н.А. Шефов*  *Д.А. Елисеев* |
|  | (Группа) | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Преподаватель | | |  | 31.01.23 | *Садков А.А* |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2023

# Практическая работа №6

# Разработка программных модулей

1. **Цель работы**
   1. Освоить методы разработки программных моделей.
   2. Приобрести навыки создания формальных моделей и разработки модульных систем.
2. **Ход работы**
   1. Используя доступные средства визуального проектирования составить проектную схему приложения для шифрования данных по алгоритму «ADFGX».
   2. Используя спроектированную схему разработать приложение.
   3. Составить отчет по проделанной работе.
   4. Защитить проделанную работу.

|  |
| --- |
| internal class Cryptograph  {  private static int[] primeNumbers = GetPrimeNumbers();  private int publicKey;  public int PublicKey { get => publicKey; private set => publicKey = value; }  private int privateKey;  private int eulersFunction;  public int n;  private const int power = 128;  public Cryptograph()  {  GenerateCryptoKeys();  }  private static bool IsPrimeNumber(int number)  {  int sqrtNumber = (int)(Math.Sqrt(number));  for (int i = 2; i <= sqrtNumber; i++)  {  if (number % i == 0)  return false;  }  return true;  }  public static int[] GetPrimeNumbers()  {  List<int> numbers = new List<int>();  int maximumPrimeNumber = 255;  for (int i = 11; i < maximumPrimeNumber; i++)  {  if (IsPrimeNumber(i))  {  numbers.Add(i);  }  }  return numbers.ToArray();  }  public void GenerateCryptoKeys()  {  Random random = new Random();  int p = primeNumbers[random.Next(primeNumbers.Length)];  int q = primeNumbers[random.Next(primeNumbers.Length)];  while (p == q)  {  q = primeNumbers[random.Next(primeNumbers.Length)];  }  n = p \* q;  eulersFunction = (p - 1) \* (q - 1);  PublicKey = CalculateE(eulersFunction);  privateKey = CalculateD(PublicKey, eulersFunction);  }  private static int CalculateE(int eulersFunction)  {  int e = 2;  for (int i = 2; i <= eulersFunction; i++)  if ((eulersFunction % i == 0) && (e % i == 0))  {  e++;  i = 1;  }  return e;  }  private static int CalculateD(int e, int eulersFunction)  {  int d = 1;  while (true)  {  if ((d \* e) % eulersFunction == 1)  break;  else  d++;  }  return d;  }  public void EncryptFile(string path, string outputPath)  {  var bytes = File.ReadAllBytes(path);  ushort[] output = new ushort[bytes.Length];  for (int i = 0; i < bytes.Length; i++)  {  output[i] = Convert.ToUInt16(FastPowFunc(bytes[i], publicKey, this.n));  }  List<byte> encoded = new List<byte>();  for (int i = 0; i < output.Length; i++)  {  encoded.AddRange(BitConverter.GetBytes(output[i]));  }  File.WriteAllBytes(outputPath, encoded.ToArray());  }  public void DecryptFile(string path, string outputPath)  {  var fileBytes = File.ReadAllBytes(path);  List<ushort> bytes = new List<ushort>();  for (int i = 0; i < fileBytes.Length; i += 2)  {  var intBytes = new byte[] { fileBytes[i], fileBytes[i + 1] };  bytes.Add(BitConverter.ToUInt16(intBytes, 0));  }  int[] output = new int[bytes.Count];  for (int i = 0; i < bytes.Count; i++)  {  output[i] = Convert.ToInt32(FastPowFunc(bytes[i], privateKey, this.n));  }  byte[] decoded = new byte[bytes.Count];  for (int i = 0; i < decoded.Length; i++)  {  decoded[i] = Convert.ToByte(output[i]);  }  File.WriteAllBytes(outputPath, decoded);  }  public Int64 FastPowFunc(Int64 Number, Int64 Pow, Int64 Mod)  {  Int64 Result = 1;  Int64 Bit = Number % Mod;  while (Pow > 0)  {  if ((Pow & 1) == 1)  {  Result \*= Bit;  Result %= Mod;  }  Bit \*= Bit;  Bit %= Mod;  Pow >>= 1;  }  return Result;  }  }  } |

1. **Вывод**
   1. Освоены методы разработки программных моделей.
   2. Приобретены навыки создания формальных моделей и разработки модульных систем.