**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по практической работе №7 МДК 03.02**

**Инструментальные средства разработки ПО**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ПКС-91 | |  |  | Васильев Р.О.  Рангелов Д.В. |
|  | (Группа) | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Преподаватель | | |  |  | Кривополенов Д.А  Садков А.А |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2023

# Практическая работа №7

**Применение сборочного программирования**

1. **Цель работы**

**1.1** Освоить методы сборочного программирования.

**1.2** Приобрести навыки создания формальных моделей и разработки ПО.

1. **Задание**
   1. На основе приложения разработанного на практической работе №6 составить программу для пакетного шифрования фалов. На вход подается путь к каталогу, который содержит файлы для шифрования и путь к каталогу для сохранения шифрованных файлов.

|  |
| --- |
| namespace PractWork5  {  public partial class MainForm : Form  {  public MainForm()  {  InitializeComponent();  }  private bool IsSimple(int n)  {  if (n < 2) return false;  if (n == 2) return true;  for (int i = 2; i < n; i++)  {  if (n % i == 0)  return false;  }  return true;  }  int[] openKey;  int[] privateKey;  string currentPath;  string currentPathRes;  private void GetKeys()  {  int p = Convert.ToInt32(PTextBox.Text);  int q = Convert.ToInt32(QTextBox.Text);  if (IsSimple(p) && IsSimple(q))  {  if (p \* q < 255)  {  MessageBox.Show("Неккоректные значения, p\*q должно быть больше 127", "Ошибка");  return;  }  if (p \* q > 1681)  {  MessageBox.Show("Значение p\*q слишком большое", "Ошибка");  return;  }  int n = p \* q;  int fn = (p - 1) \* (q - 1);  int exp = 0;  for (int i = 2; i < fn; i++)  {  if (fn % i != 0)  {  exp = i;  break;  }  }  openKey = new int[2] { exp, n };  int d = 0;  while (true)  {  if ((d \* exp) % fn == 1)  {  break;  }  d++;  }  privateKey = new int[2] { d, n };  }  }  private void Encrypt(string path)  {  try  {  GetKeys();  int symb;  int siphCounter = 1;  byte[] arr;  byte[] asciiBytes;  DirectoryInfo d = new DirectoryInfo(currentPath);  FileInfo[] files = d.GetFiles("\*.\*", SearchOption.AllDirectories);  foreach (FileInfo file in files)  {  arr = File.ReadAllBytes(file.FullName);  asciiBytes = new byte[arr.Length \* 2];  for (int i = 0; i < arr.Length; i++)  {  symb = (int)arr[i];  for (int j = 0; j < openKey[0]; j++)  {  siphCounter = symb \* siphCounter % openKey[1];  }  BitConverter.GetBytes((ushort)siphCounter).CopyTo(asciiBytes, i \* 2);  siphCounter = 1;  }  File.WriteAllBytes(Path.Combine(currentPathRes, file.Name), asciiBytes);  }  MessageBox.Show("Файлы зашифрованы", "Поздравляем!", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show(ex.Message, "Ошибка");  }  }  private void Decrypt()  {  byte[] asciiBytesGet;  int[] massCiph;  int symb;  int siphCounter = 1;  int[] arrCiph;  byte[] asciiBytes;  try  {  DirectoryInfo d = new DirectoryInfo(currentPathRes);  FileInfo[] files = d.GetFiles("\*.\*", SearchOption.AllDirectories);  foreach (FileInfo file in files)  {  asciiBytesGet = File.ReadAllBytes(file.FullName);  massCiph = new int[asciiBytesGet.Length / 2];  arrCiph = new int[massCiph.Length];  asciiBytes = new byte[arrCiph.Length];  for (int i = 0; i < asciiBytesGet.Length / 2; i++)  {  massCiph[i] = BitConverter.ToUInt16(asciiBytesGet, i \* 2);  }  //дешифрование  for (int i = 0; i < massCiph.Length; i++)  {  symb = massCiph[i];  for (int j = 0; j < privateKey[0]; j++)  {  siphCounter = symb \* siphCounter % privateKey[1];  }  arrCiph[i] = siphCounter;  siphCounter = 1;  }  for (int i = 0; i < arrCiph.Length; i++)  {  asciiBytes[i] = Convert.ToByte(arrCiph[i]);  }  File.WriteAllBytes(file.FullName, asciiBytes);  }  MessageBox.Show("Файлы расшифрованы", "Поздравляем!", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show(ex.Message, "Ошибка");  }  }  private void EncryptButton\_Click(object sender, EventArgs e)  {  Encrypt(currentPath);  }  private void DecryptButton\_Click(object sender, EventArgs e)  {  Decrypt();  }  private void PathButton\_Click(object sender, EventArgs e)  {  FolderBrowserDialog dialog = new FolderBrowserDialog();  if (dialog.ShowDialog() == DialogResult.OK)  {  currentPath = OriginTextBox.Text = dialog.SelectedPath;  }  }  private void PurposePathButton\_Click(object sender, EventArgs e)  {  FolderBrowserDialog dialog = new FolderBrowserDialog();  if (dialog.ShowDialog() == DialogResult.OK)  {  currentPathRes = PurposeTextBox.Text = dialog.SelectedPath;  }  }  private void PTextBox\_KeyPress(object sender, KeyPressEventArgs e)  {  if ((e.KeyChar >= '0' && e.KeyChar <= '9') || e.KeyChar == (char)Keys.Back)  {  }  else  {  e.Handled = true;  }  }  private void QTextBox\_KeyPress(object sender, KeyPressEventArgs e)  {  if ((e.KeyChar >= '0' && e.KeyChar <= '9') || e.KeyChar == (char)Keys.Back)  {  }  else  {  e.Handled = true;  }  }  private void OriginTextBox\_KeyPress(object sender, KeyPressEventArgs e)  {  if ((e.KeyChar >= 'A' && e.KeyChar <= 'Z') || (e.KeyChar >= 'a' && e.KeyChar <= 'z') || (e.KeyChar >= '0' && e.KeyChar <= '9') || e.KeyChar == (char)Keys.Back)  {  }  else  {  e.Handled = true;  }  }  }  } |

**3** Вывод

**3.1**В ходе работы освоены методы сборочного программирования, приобретены навыки создания формальных моделей и разработки ПО.