

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)**

Факультет Информатика и вычислительная техника

(наименование факультета)

Кафедра Кибербезопасность информационных систем

(наименование кафедры)

**ОТЧЕТ**

по лабораторным работам по предмету: Языки Программирования

Выполнил:

ст. гр. ВКБ31 Котелевец К.А.

Ростов-на-Дону

2024

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Разработка Windows приложений на C#. Введение в разработку форм.**

**Цель**: изучение методов построения форм Windows и получение навыков по настройке форм, созданию непрямоугольных и наследуемых форм в Microsoft Visual Studio, используя С#.

**Ход работы**

Для начала выполнения заданий, я сделаю форму, в которой смогу выбирать нужное мне задание (их всего 3). На рисунке 1.1 представлено окно с выбором.

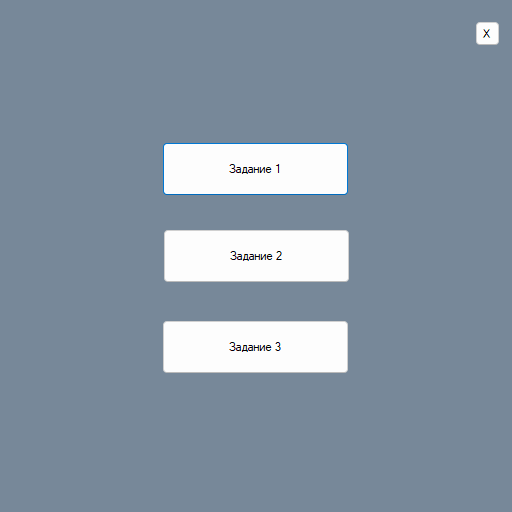


Рисунок 1.1 – Окно выбора заданий

**Задание 1.** Создайте пользовательскую форму, которая во время выполнения будет иметь овальное очертание. Данная форма должна содержать функциональность, дающую возможность пользователю закрывать ее во время выполнения. Рекомендация: при разработке формы в виде эллипса используйте следующий код: myPath.AddEllipse(0, 0, this.Width, this.Height);

На рисунке 1.2 представлено данное овальное окно с кнопкой по центру.

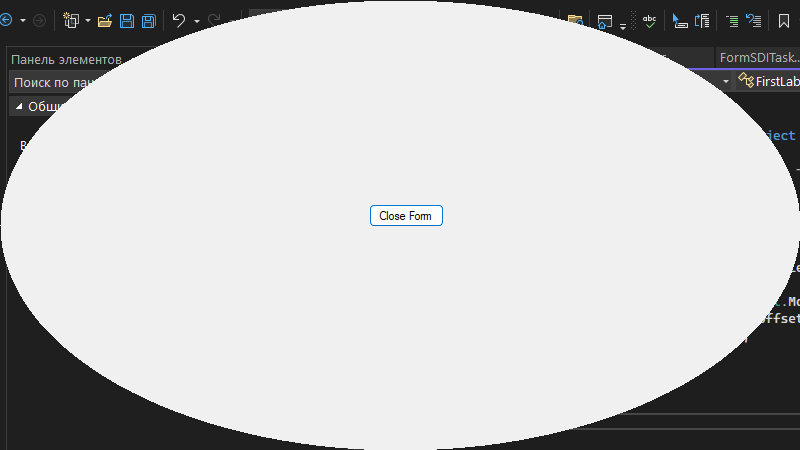


Рисунок 1.2 – Форма первого задания

**Задание 2.** Создайте приложение с двумя формами и установите вторую форму как стартовую. Сделайте так, чтобы при запуске стартовая форма разворачивалась до максимальных размеров и содержала функциональность, дающую возможность пользователю открыть первую форму, отображающуюся в виде ромба зеленого цвета с кнопкой (в центре ромба) закрытия формы с надписью GREENPEACE. Первая форма должна в тайтле выводить ФИО студента и группу обучения.

На рисунке 1.3 и 1.4 представлены соответственно стартовая и вторая формы.

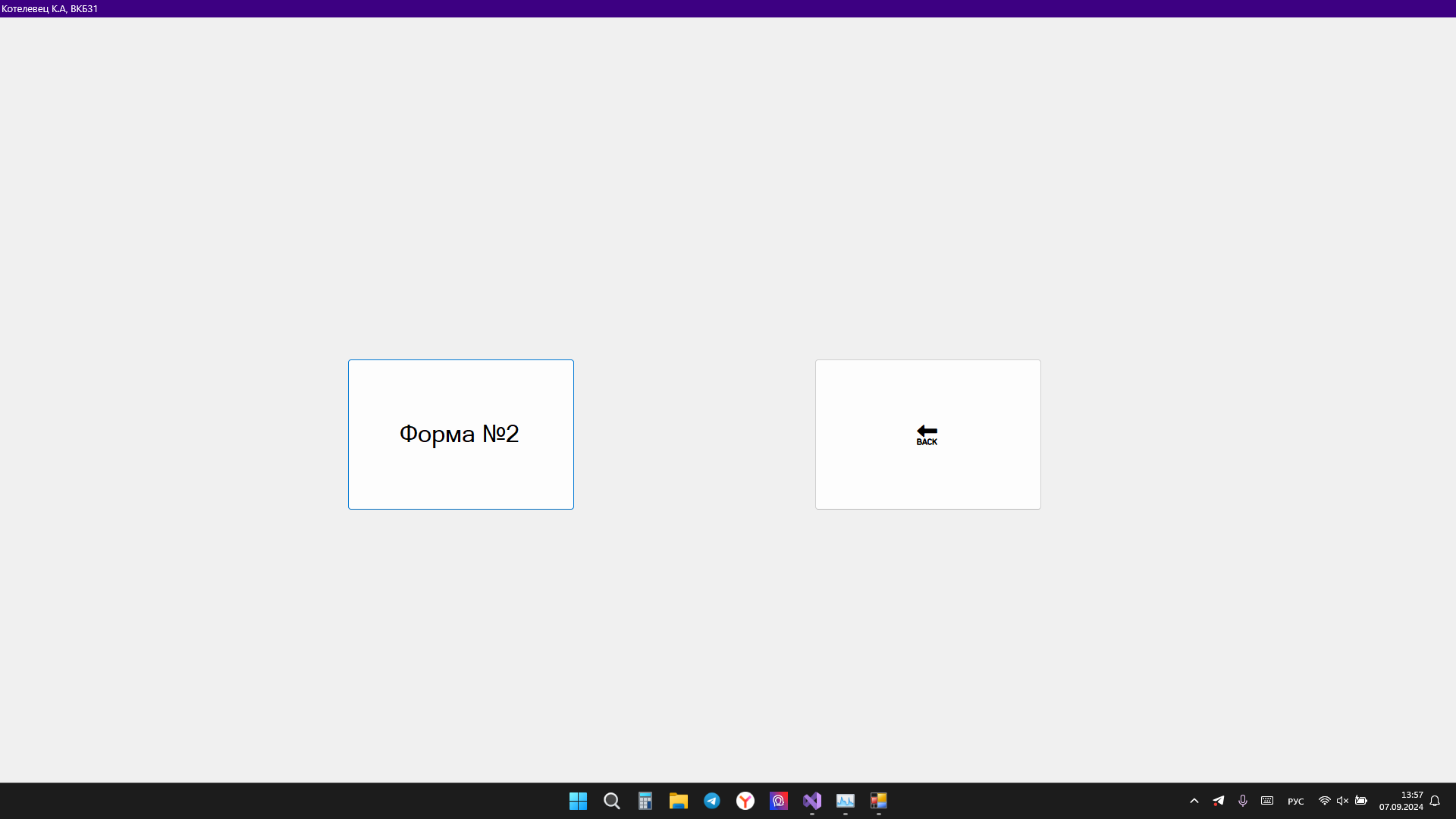


Рисунок 1.3 – Стартовое окно

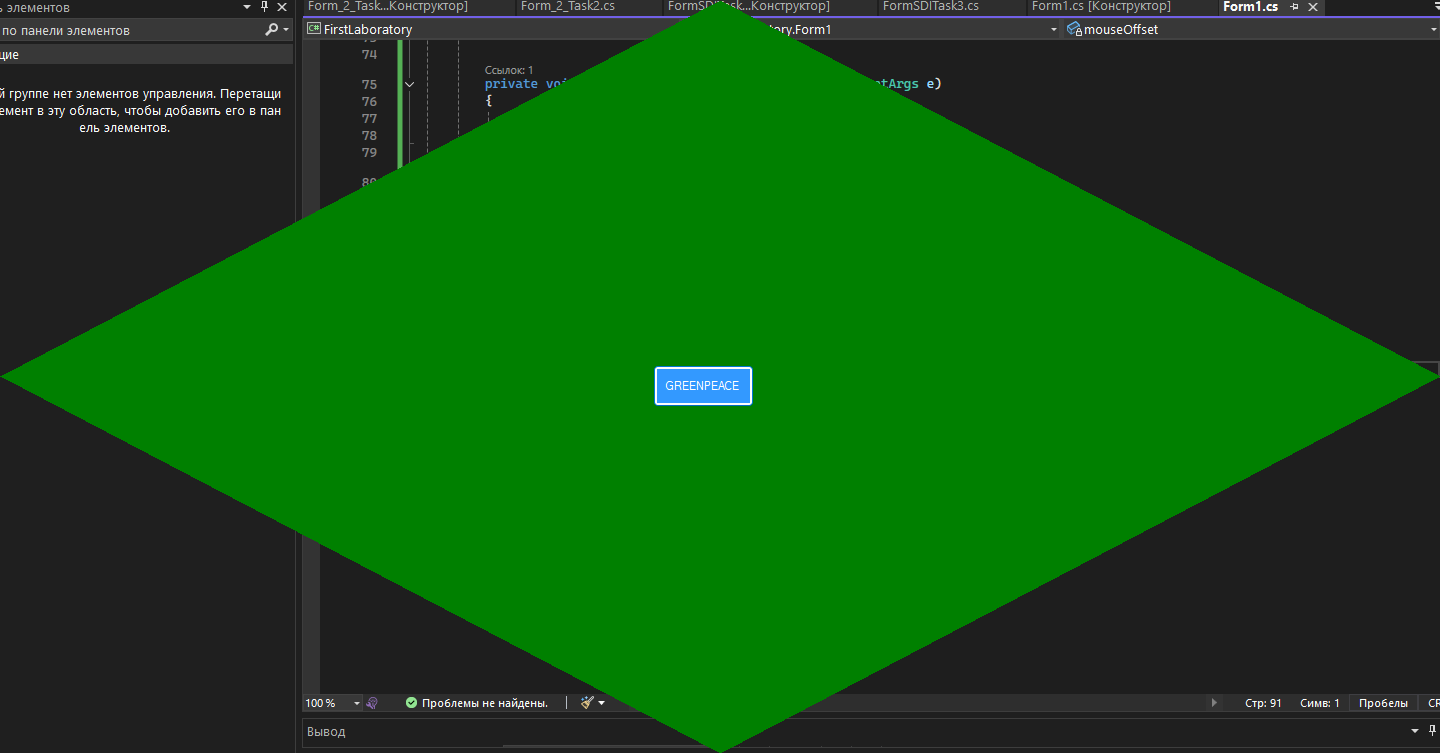


Рисунок 1.4 – Вторая форма

**Задание 3.** Создайте приложение, которое будет поддерживать организацию форм MDI. При создании MDI формы, на ней должна быть кнопка для открытия формы SDI. Форма SDI при открытии должна быть черным цветом и не иметь компонентов управления, при этом ее размеры можно было менять, так же форма должна иметь в центре кнопку её закрытия.

На рисунке 1.5 представлено окно MDI формы, а на рисунке 1.6 уже окно SDI формы.

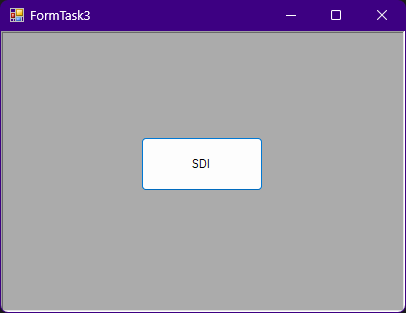


Рисунок 1.5 – окно MDI формы

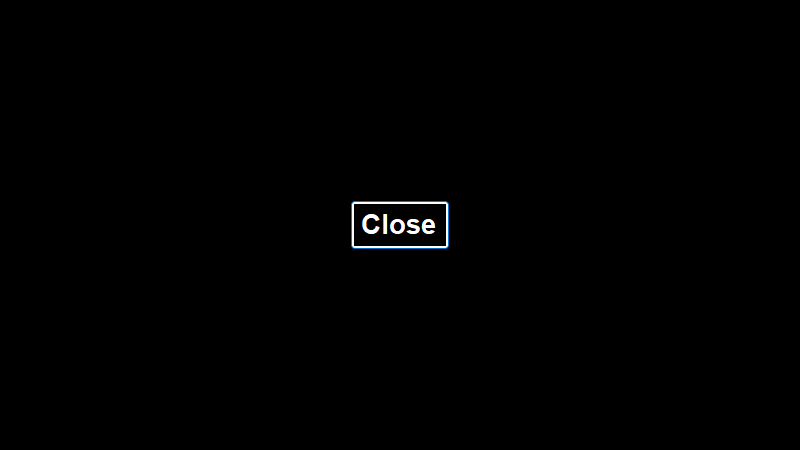


Рисунок 1.6 – окно SDI формы

Ниже представлены листинги программы каждой формы.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace FirstLaboratory  {  public partial class Form1 : Form  {  private Form FormTask1;  private Form Form\_1\_Task2;  private Form FormTask3;  public Form1()  {  InitializeComponent();  this.MouseDown += new MouseEventHandler(Form\_MouseDown);  this.MouseMove += new MouseEventHandler(Form\_MouseMove);  }  private void button3\_Click(object sender, EventArgs e)  {  this.Opacity = 1;  }  private void button1\_Click(object sender, EventArgs e)  {  if (FormTask1 == null || FormTask1.IsDisposed)  FormTask1 = new FormTask1();  FormTask1.Show();  this.Hide();  }    private void Form1\_Load(object sender, EventArgs e)  {    }  private void BtnTask2\_Click(object sender, EventArgs e)  {  if (Form\_1\_Task2 == null || Form\_1\_Task2.IsDisposed)  Form\_1\_Task2 = new Form\_1\_Task2();  Form\_1\_Task2.Show();  this.Hide();  }  private void BtnTask3\_Click(object sender, EventArgs e)  {  if (FormTask3 == null || FormTask3.IsDisposed)  FormTask3 = new FormTask3();  FormTask3.Show();  this.Hide();  }  private void button1\_Click\_1(object sender, EventArgs e)  {  this.Close();  Application.Exit();  }  private Point mouseOffset;    private void Form\_MouseDown(object sender, MouseEventArgs e)  {  mouseOffset = new Point(-e.X, -e.Y);  }  private void Form\_MouseMove(object sender, MouseEventArgs e)  {  if (e.Button == MouseButtons.Left)  {  Point mousePos = Control.MousePosition;  mousePos.Offset(mouseOffset.X, mouseOffset.Y);  Location = mousePos;  }  }  }  } |

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace FirstLaboratory  {  public partial class FormTask1 : Form  {  private Form Form1;  public FormTask1()  {  InitializeComponent();  }  private void FormTask1\_Load(object sender, EventArgs e)  {  System.Drawing.Drawing2D.GraphicsPath myPath = new  System.Drawing.Drawing2D.GraphicsPath();  myPath.AddEllipse(0, 0, this.Width, this.Height);  Region myRegion = new Region(myPath); this.Region = myRegion;  }  private void BtnCloseForm\_Click(object sender, EventArgs e)  {  if (Form1 == null || Form1.IsDisposed)  Form1 = new Form1();  Form1.Show();  this.Close();  }  }  } |

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace FirstLaboratory  {  public partial class Form\_1\_Task2 : Form  {  private Form Form\_2\_Task2;  private Form Form1;  public Form\_1\_Task2()  {  InitializeComponent();  this.FormBorderStyle = FormBorderStyle.SizableToolWindow;  this.ControlBox = false;  this.Text = "Котелевец К.А, ВКБ31";  }  private void Form\_1\_Task2\_Load(object sender, EventArgs e)  {  this.WindowState = FormWindowState.Maximized;  }  private void button1\_Click(object sender, EventArgs e)  {  if (Form\_2\_Task2 == null || Form\_2\_Task2.IsDisposed)  Form\_2\_Task2 = new Form\_2\_Task2();  Form\_2\_Task2.Show();  this.Close();  }  private void BtnGoToForm1\_Click(object sender, EventArgs e)  {  if (Form1 == null || Form1.IsDisposed)  Form1 = new Form1();  Form1.Show();  this.Close();  }  }  } |

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Drawing.Drawing2D;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace FirstLaboratory  {  public partial class Form\_2\_Task2 : Form  {  private Form Form1;  private Button button1;  public Form\_2\_Task2()  {  InitializeComponent();  this.FormBorderStyle = FormBorderStyle.None;  this.MouseDown += new MouseEventHandler(Form\_MouseDown);  this.MouseMove += new MouseEventHandler(Form\_MouseMove);  }  private Point mouseOffset;  private void Form\_MouseDown(object sender, MouseEventArgs e)  {  mouseOffset = new Point(-e.X, -e.Y);  }  private void Form\_MouseMove(object sender, MouseEventArgs e)  {  if (e.Button == MouseButtons.Left)  {  Point mousePos = Control.MousePosition;  mousePos.Offset(mouseOffset.X, mouseOffset.Y);  Location = mousePos;  }  }    private void Form\_2\_Task2\_Load(object sender, EventArgs e)  {  this.BackColor = Color.Green;  System.Drawing.Drawing2D.GraphicsPath myPath = new System.Drawing.Drawing2D.GraphicsPath();  myPath.AddPolygon(new Point[] {  new Point(this.Width / 2, 0),  new Point(this.Width, this.Height / 2),  new Point(this.Width / 2, this.Height),  new Point(0, this.Height / 2)  });  Region myRegion = new Region(myPath);  this.Region = myRegion;  this.BackColor = Color.Green;  if (button1 != null)  {  button1.Location = new Point((this.Width - button1.Width) / 2, (this.Height - button1.Height) / 2);  }  }  private void button1\_Click(object sender, EventArgs e)  {  if (Form1 == null || Form1.IsDisposed)  Form1 = new Form1();  Form1.Show();  this.Close();  }  }  } |

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace FirstLaboratory  {  public partial class FormTask3 : Form  {  private Form FormSDITask3;  public FormTask3()  {  InitializeComponent();  }  private void FormTask3\_Load(object sender, EventArgs e)  {  }  private void button1\_Click(object sender, EventArgs e)  {  if (FormSDITask3 == null || FormSDITask3.IsDisposed)  FormSDITask3 = new FormSDITask3();  FormSDITask3.Show();  this.Close();  }  }  } |

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using static System.Windows.Forms.VisualStyles.VisualStyleElement.Window;  namespace FirstLaboratory  {  public partial class FormSDITask3 : Form  {  private Button closeButton;  private Form Form1;  private Point mouseOffset;  public FormSDITask3()  {  InitializeComponent();  this.Text = "";  this.FormBorderStyle = FormBorderStyle.None; // Удаление компонентов управления  this.BackColor = Color.Black;  closeButton = new Button();  closeButton.Text = "Close";  closeButton.Font = new Font("Arial", 20, FontStyle.Bold); // Установка шрифта Arial, размер 24, жирный  closeButton.ForeColor = Color.White; // Установка белого цвета шрифта  closeButton.Size = new Size(100, 50); // Установка размера кнопки  closeButton.Click += CloseButton\_Click;  this.Controls.Add(closeButton);  this.MouseDown += new MouseEventHandler(Form\_MouseDown);  this.MouseMove += new MouseEventHandler(Form\_MouseMove);  this.Resize += SDIForm\_Resize; // Обработка события изменения размера формы  this.ResizeEnd += SDIForm\_ResizeEnd; // Обработка события окончания изменения размера  }  private void FormSDITask3\_Load(object sender, EventArgs e)  {  }  private void CloseButton\_Click(object sender, EventArgs e)  {  if (Form1 == null || Form1.IsDisposed)  Form1 = new Form1();  Form1.Show();  this.Close();  }  private void SDIForm\_Resize(object sender, EventArgs e)  {  closeButton.Location = new Point(this.Width / 2 - closeButton.Width / 2, this.Height / 2 - closeButton.Height / 2);  }  private void SDIForm\_ResizeEnd(object sender, EventArgs e)  {  closeButton.Location = new Point(this.Width / 2 - closeButton.Width / 2, this.Height / 2 - closeButton.Height / 2);  }  private void Form\_MouseDown(object sender, MouseEventArgs e)  {  mouseOffset = new Point(-e.X, -e.Y);  }  private void Form\_MouseMove(object sender, MouseEventArgs e)  {  if (e.Button == MouseButtons.Left)  {  Point mousePos = Control.MousePosition;  mousePos.Offset(mouseOffset.X, mouseOffset.Y);  Location = mousePos;  }  }  protected override void WndProc(ref Message m)  {  const int WM\_NCHITTEST = 0x0084;  const int HTCLIENT = 1;  const int HTCAPTION = 2;  const int HTBOTTOM = 15;  const int HTBOTTOMLEFT = 16;  const int HTBOTTOMRIGHT = 17;  const int HTLEFT = 10;  const int HTRIGHT = 11;  const int HTTOP = 12;  const int HTTOPLEFT = 13;  const int HTTOPRIGHT = 14;  if (m.Msg == WM\_NCHITTEST)  {  Point point = PointToClient(Cursor.Position);  if (point.X < 5 && point.Y < 5)  m.Result = (IntPtr)HTTOPLEFT;  else if (point.X > Width - 5 && point.Y < 5)  m.Result = (IntPtr)HTTOPRIGHT;  else if (point.X < 5 && point.Y > Height - 5)  m.Result = (IntPtr)HTBOTTOMLEFT;  else if (point.X > Width - 5 && point.Y > Height - 5)  m.Result = (IntPtr)HTBOTTOMRIGHT;  else if (point.X < 5)  m.Result = (IntPtr)HTLEFT;  else if (point.X > Width - 5)  m.Result = (IntPtr)HTRIGHT;  else if (point.Y < 5)  m.Result = (IntPtr)HTTOP;  else if (point.Y > Height - 5)  m.Result = (IntPtr)HTBOTTOM;  else  m.Result = (IntPtr)HTCLIENT;  }  else  {  base.WndProc(ref m);  }  }  }  } |

**Контрольные вопросы:**

1. **Что такое C#?**

C# (C-Sharp) - это современный объектно-ориентированный язык программирования, разработанный компанией Microsoft как часть фреймворка .NET. Он был создан для замены Visual Basic и предоставить альтернативу Java для платформы Windows.

Ключевые особенности C#:

Объектно-ориентированный язык

Статически типизированный

Поддерживает многопоточность

Интегрируется с .NET Framework

1. **В чем отличие C# от других языков программирования?**

Основные отличия C# от других языков:

Более быстрая компиляция благодаря использованию JIT-компилятора

Улучшенная поддержка многопоточности

Интеграция с .NET Framework и широкий набор библиотек

Совместимость с другими языками .NET (F#, VB.NET)

Хорошая поддержка от Microsoft

1. **Какую парадигму программирования поддерживает С#?**

C# поддерживает несколько парадигм программирования:

ООП (объектно-ориентированное программирование)

Процедурное программирование

Функциональное программирование (через лямбда-выражения)

1. **Что такое Microsoft .NET Framework?**

Microsoft .NET Framework - это крупная платформа для разработки и выполнения приложений. Она включает:

Runtime (.NET Common Language Runtime)

Библиотеки классов

Средства разработки

API для взаимодействия с операционной системой

.NET Framework обеспечивает кроссплатформенность приложений C# и других .NET-языков.

1. **Что такое MDI?**

MDI (Multiple Document Interface) - это интерфейс пользовательского интерфейса, который позволяет открыть несколько документов в одном окне приложения.

1. **Что такое SDI?**

SDI (Single Document Interface) - это интерфейс пользовательского интерфейса, где каждое окно представляет собой отдельный документ или форму.

1. **Какие бывают формы?**

В Windows Forms существуют следующие виды форм:

Form (основная форма)

UserControl (пользовательский элемент управления)

MessageBox (окно сообщения)

OpenFileDialog (диалог выбора файла)

SaveFileDialog (диалог сохранения файла)

1. **Основные свойства формы?**

Основные свойства формы в Windows Forms:

Text - текст заголовка формы

Size - размер формы

Location - координаты положения формы

BackColor - цвет фона формы

ForeColor - цвет текста

Font - шрифт текста

Visible - видимость формы

TopLevel - является ли форма верхней (основной) формой

FormBorderStyle - стиль рамки формы

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Создание приложения Windows Form. Стандартные элементы управления и компоненты.**

**Цель**: изучить методы создания приложений с использованием Windows Form. Также изучить стандартные компоненты и научиться их применять на практике.

**Ход работы**

**Задание 1.**

Создать калькулятор, который полностью повторяет обычный калькулятор Windows. Интерфейс должен быть аналогичен калькулятору Windows 7. Необходимо, чтоб программа отображала ошибки пользователя.

На рисунке 2.1 представлен интерфейс моего калькулятора.

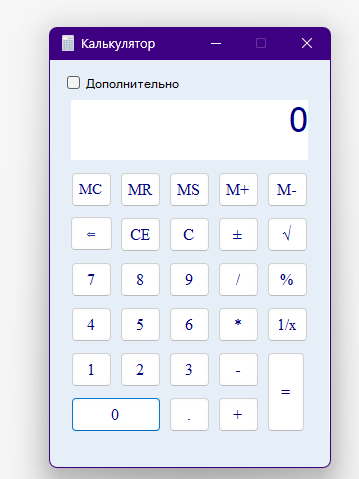


Рисунок 1.1 – Интерфейс моего калькулятора

Также она выводит ошибки, например, если делить на ноль (рисунок 1.2)

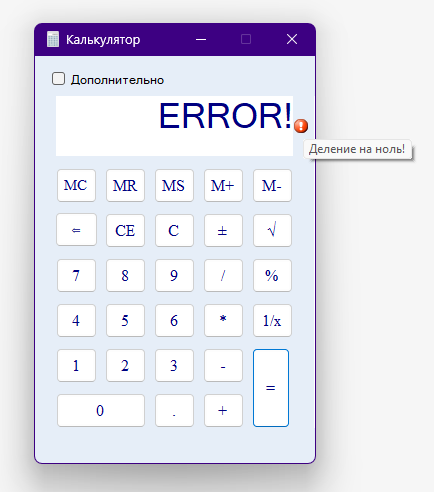


Рисунок 1.2 – Деление на ноль

Задание 2. Добавить в калькулятор функции согласно варианту (9) и вид калькулятора должен изменяться по событию. Используйте для решения поставленной задачи два дополнительных компонента из библиотеки, которые не изображены. Необходимо, чтобы программа отображала ошибки пользователя с использованием ErrorProvider. На рисунке 1.3 представлен интерфейс моего калькулятора с дополнительной функцией конвертации. Для ее запуска нужно просто нажать дополнительно.

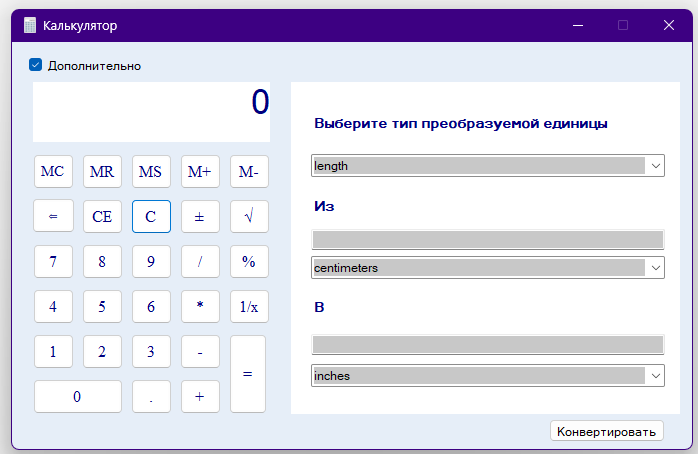


Рисунок 1.3 – Дополнительная функция калькулятора.

На рисунке 1.4 представлен вывод ошибки, если, например, ничего не ввести конвертировать.

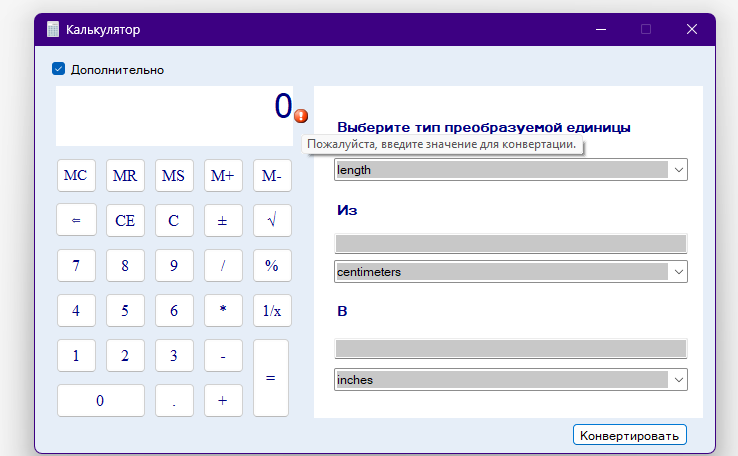


Рисунок 1.4 – Ошибка конвертации

Листинг программы:

Form1.cs

|  |
| --- |
| using System;  using System.Windows.Forms;  namespace Windows7Calculator  {  public partial class Form1 : Form  {  private double memory;  double firstNumber = 0;  double secondNumber = 0;  string operation = "";  bool operationPressed = false;  private const int MaxLength = 12;  private ErrorProvider errorProvider;  public Form1()  {  InitializeComponent();  errorProvider = new ErrorProvider();  }  private void InitializeCalculatorLayout()  {  errorProvider.BlinkStyle = ErrorBlinkStyle.AlwaysBlink;  errorProvider.BlinkRate = 1000;  }  private void CalculatorUI\_Load(object sender, EventArgs e)  {  // Убрать обводку поля  calculatorScreen.BorderStyle = BorderStyle.None;  InitializeCalculatorLayout();  comboBox1.SelectedIndex = 0;  comboBox2.SelectedIndex = 1;  comboBox3.SelectedIndex = 0;  }  private void eraseButton\_Click(object sender, EventArgs e)  {  string t = calculatorScreen.Text;  string res = "";  for (int i = 0; i < t.Length - 1; i++)  {  res = res + t[i];  }  calculatorScreen.Text = res;  }  private void calculatorScreen\_TextChanged(object sender, EventArgs e)  {  if (calculatorScreen.Text.Length > MaxLength)  {  calculatorScreen.Text = calculatorScreen.Text.Substring(0, MaxLength);  calculatorScreen.SelectionStart = calculatorScreen.Text.Length;  calculatorScreen.SelectionLength = 0;  }  }  private void cButton\_Click(object sender, EventArgs e)  {  calculatorScreen.Text = "0";  firstNumber = 0;  secondNumber = 0;  }  private void ceButton\_Click(object sender, EventArgs e)  {  calculatorScreen.Text = "0";  }  private void equalsButton\_Click(object sender, EventArgs e)  {  string[] parts = calculatorScreen.Text.Split(' ');  secondNumber = double.Parse(parts[parts.Length - 1]);  switch (operation)  {  case "+":  calculatorScreen.Text = (firstNumber + secondNumber).ToString();  break;  case "-":  calculatorScreen.Text = (firstNumber - secondNumber).ToString();  break;  case "\*":  calculatorScreen.Text = (firstNumber \* secondNumber).ToString();  break;  case "/":  calculatorScreen.Text = (firstNumber / secondNumber).ToString();  if (secondNumber != 0)  calculatorScreen.Text = (firstNumber / secondNumber).ToString();  else  ShowError("Деление на ноль!");  calculatorScreen.Text = "ERROR!";  break;  default:  break;  }  operationPressed = false;  }  private void numberButton\_Click(object sender, EventArgs e)  {  if (calculatorScreen.Text == "0")  calculatorScreen.Text = "";  Button b = (Button)sender;  calculatorScreen.Text = calculatorScreen.Text + b.Text;  }  private void operationButton\_Click(object sender, EventArgs e)  {  Button b = (Button)sender;  if (operationPressed == false)  firstNumber = double.Parse(calculatorScreen.Text);  calculatorScreen.Text = "";  operationPressed = true;  operation = b.Text;  }  private void PlusMinusButton\_Click(object sender, EventArgs e)  {  string currentText = calculatorScreen.Text;  double currentValue = double.Parse(currentText);  if (currentValue > 0)  {  calculatorScreen.Text = "-" + currentText;  }  else if (currentValue < 0)  {  calculatorScreen.Text = currentText.Substring(1); // remove the negative sign  }  }  private void fracButton\_Click(object sender, EventArgs e)  {  double number = double.Parse(calculatorScreen.Text);  number = 1 / number;  calculatorScreen.Text = number.ToString();  }  private void sqrtButton\_Click(object sender, EventArgs e)  {  double number = double.Parse(calculatorScreen.Text);  if (number >= 0)  calculatorScreen.Text = Math.Sqrt(number).ToString();  else  ShowError("Нельзя вычислить квадратный корень отрицательного числа!");  }  private void powerButton\_Click(object sender, EventArgs e)  {  memory = 0;  // Clear the memory  }  private void percentButton\_Click(object sender, EventArgs e)  {  double number = double.Parse(calculatorScreen.Text);  number /= 100;  calculatorScreen.Text = number.ToString();  }  private void MRButton\_Click(object sender, EventArgs e)  {  calculatorScreen.Text = memory.ToString();  }  private void button1\_Click(object sender, EventArgs e)  {  if (calculatorScreen.Text != "")  {  memory += double.Parse(calculatorScreen.Text);  }  }  private void MMinusButton\_Click(object sender, EventArgs e)  {  if (calculatorScreen.Text != "")  {  memory -= double.Parse(calculatorScreen.Text);  }  }  private void MSButton\_Click\_1(object sender, EventArgs e)  {  memory = double.Parse(calculatorScreen.Text);  }  private void checkBox1\_CheckedChanged(object sender, EventArgs e)  {  if (checkBox1.Checked)  {  this.Size = new System.Drawing.Size(700, 450);  }  if (!checkBox1.Checked)  {  this.Size = new System.Drawing.Size(300, 450);  }  }  private void label2\_Click(object sender, EventArgs e)  {  }  private void label3\_Click(object sender, EventArgs e)  {  }  private void ShowError(string message)  {  errorProvider.SetError(calculatorScreen, message);  }  private void ComboBox1\_SelectedIndexChanged(object sender, EventArgs e) {}  private void ComboBox2\_SelectedIndexChanged(object sender, EventArgs e) {}  private void ComboBox3\_SelectedIndexChanged(object sender, EventArgs e) {}  private double PerformConversion(double inputValue, string inputUnit, string outputUnit)  {  double result = 0;  switch (inputUnit)  {  case "centimeters":  switch (outputUnit)  {  case "inches":  result = ConvertCmToInches(inputValue);  break;  case "centimeters":  result = inputValue;  break;  }  break;  case "inches":  switch (outputUnit)  {  case "centimeters":  result = ConvertInchesToCm(inputValue);  break;  case "inches":  result = inputValue;  break;  }  break;  }  return result;  }  private double ConvertCmToInches(double cm)  {  return cm \* 0.393701;  }  private double ConvertInchesToCm(double inches)  {  return inches \* 2.54;  }  private void button1\_Click\_1(object sender, EventArgs e) //выполнение конвертизации  {  if (string.IsNullOrEmpty(textBox2.Text))  {  System.Console.Write(textBox2.Text);  ShowError("Пожалуйста, введите значение для конвертации.");  return;  }  double inputValue = double.Parse(textBox2.Text);  string inputUnit = comboBox2.SelectedItem.ToString();  string outputUnit = comboBox3.SelectedItem.ToString();  double result = PerformConversion(inputValue, inputUnit, outputUnit);  if (result != 0)  {  textBox1.Text = result.ToString("F5");  }  else  {  ShowError("Ошибка при выполнении операции.");  }  }  }  } |

Form1.Designer.cs

|  |
| --- |
| using System;  namespace Windows7Calculator  {  partial class Form1  {  private System.ComponentModel.IContainer components = null;  /// <summary>  /// Clean up any resources being used.  /// </summary>  /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>  protected override void Dispose(bool disposing)  {  if (disposing && (components != null))  {  components.Dispose();  }  base.Dispose(disposing);  }  /// <summary>  /// Required method for Designer support - do not modify  /// the contents of this method with the code editor.  /// </summary>  private void InitializeComponent()  {  System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Form1));  this.zeroButton = new System.Windows.Forms.Button();  this.decimalButton = new System.Windows.Forms.Button();  this.plusButton = new System.Windows.Forms.Button();  this.oneButton = new System.Windows.Forms.Button();  this.twoButton = new System.Windows.Forms.Button();  this.threeButton = new System.Windows.Forms.Button();  this.minusButton = new System.Windows.Forms.Button();  this.fourButton = new System.Windows.Forms.Button();  this.fiveButton = new System.Windows.Forms.Button();  this.sixButton = new System.Windows.Forms.Button();  this.multiplyButton = new System.Windows.Forms.Button();  this.sevenButton = new System.Windows.Forms.Button();  this.eightButton = new System.Windows.Forms.Button();  this.nineButton = new System.Windows.Forms.Button();  this.divideButton = new System.Windows.Forms.Button();  this.equalsButton = new System.Windows.Forms.Button();  this.cButton = new System.Windows.Forms.Button();  this.ceButton = new System.Windows.Forms.Button();  this.eraseButton = new System.Windows.Forms.Button();  this.PlusMinusButton = new System.Windows.Forms.Button();  this.percentButton = new System.Windows.Forms.Button();  this.fracButton = new System.Windows.Forms.Button();  this.MCButton = new System.Windows.Forms.Button();  this.sqrtButton = new System.Windows.Forms.Button();  this.MRButton = new System.Windows.Forms.Button();  this.MSButton = new System.Windows.Forms.Button();  this.MPlusButton = new System.Windows.Forms.Button();  this.MMinusButton = new System.Windows.Forms.Button();  this.tableLayoutPanel1 = new System.Windows.Forms.TableLayoutPanel();  this.checkBox1 = new System.Windows.Forms.CheckBox();  this.calculatorScreen = new System.Windows.Forms.TextBox();  this.panel1 = new System.Windows.Forms.Panel();  this.panel2 = new System.Windows.Forms.Panel();  this.label3 = new System.Windows.Forms.Label();  this.label2 = new System.Windows.Forms.Label();  this.label1 = new System.Windows.Forms.Label();  this.textBox2 = new System.Windows.Forms.TextBox();  this.textBox1 = new System.Windows.Forms.TextBox();  this.comboBox3 = new System.Windows.Forms.ComboBox();  this.comboBox2 = new System.Windows.Forms.ComboBox();  this.comboBox1 = new System.Windows.Forms.ComboBox();  this.button1 = new System.Windows.Forms.Button();  this.tableLayoutPanel1.SuspendLayout();  this.panel1.SuspendLayout();  this.panel2.SuspendLayout();  this.SuspendLayout();  //  // zeroButton  //  this.zeroButton.Anchor = ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Bottom)  | System.Windows.Forms.AnchorStyles.Left)  | System.Windows.Forms.AnchorStyles.Right)));  this.zeroButton.BackColor = System.Drawing.Color.White;  this.zeroButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.zeroButton.ForeColor = System.Drawing.Color.Navy;  this.zeroButton.Location = new System.Drawing.Point(13, 328);  this.zeroButton.Margin = new System.Windows.Forms.Padding(4);  this.zeroButton.Name = "zeroButton";  this.zeroButton.Size = new System.Drawing.Size(90, 35);  this.zeroButton.TabIndex = 0;  this.zeroButton.Text = "0";  this.zeroButton.UseVisualStyleBackColor = false;  this.zeroButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // decimalButton  //  this.decimalButton.BackColor = System.Drawing.Color.White;  this.decimalButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.decimalButton.ForeColor = System.Drawing.Color.Navy;  this.decimalButton.Location = new System.Drawing.Point(102, 229);  this.decimalButton.Margin = new System.Windows.Forms.Padding(4);  this.decimalButton.Name = "decimalButton";  this.decimalButton.Size = new System.Drawing.Size(41, 35);  this.decimalButton.TabIndex = 1;  this.decimalButton.Text = ".";  this.decimalButton.UseVisualStyleBackColor = false;  this.decimalButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // plusButton  //  this.plusButton.BackColor = System.Drawing.Color.White;  this.plusButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.plusButton.ForeColor = System.Drawing.Color.Navy;  this.plusButton.Location = new System.Drawing.Point(151, 229);  this.plusButton.Margin = new System.Windows.Forms.Padding(4);  this.plusButton.Name = "plusButton";  this.plusButton.Size = new System.Drawing.Size(41, 35);  this.plusButton.TabIndex = 2;  this.plusButton.Text = "+";  this.plusButton.UseVisualStyleBackColor = false;  this.plusButton.Click += new System.EventHandler(this.operationButton\_Click);  //  // oneButton  //  this.oneButton.BackColor = System.Drawing.Color.White;  this.oneButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.oneButton.ForeColor = System.Drawing.Color.Navy;  this.oneButton.Location = new System.Drawing.Point(4, 184);  this.oneButton.Margin = new System.Windows.Forms.Padding(4);  this.oneButton.Name = "oneButton";  this.oneButton.Size = new System.Drawing.Size(41, 35);  this.oneButton.TabIndex = 3;  this.oneButton.Text = "1";  this.oneButton.UseVisualStyleBackColor = false;  this.oneButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // twoButton  //  this.twoButton.BackColor = System.Drawing.Color.White;  this.twoButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.twoButton.ForeColor = System.Drawing.Color.Navy;  this.twoButton.Location = new System.Drawing.Point(53, 184);  this.twoButton.Margin = new System.Windows.Forms.Padding(4);  this.twoButton.Name = "twoButton";  this.twoButton.Size = new System.Drawing.Size(41, 35);  this.twoButton.TabIndex = 4;  this.twoButton.Text = "2";  this.twoButton.UseVisualStyleBackColor = false;  this.twoButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // threeButton  //  this.threeButton.BackColor = System.Drawing.Color.White;  this.threeButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.threeButton.ForeColor = System.Drawing.Color.Navy;  this.threeButton.Location = new System.Drawing.Point(102, 184);  this.threeButton.Margin = new System.Windows.Forms.Padding(4);  this.threeButton.Name = "threeButton";  this.threeButton.Size = new System.Drawing.Size(41, 35);  this.threeButton.TabIndex = 5;  this.threeButton.Text = "3";  this.threeButton.UseVisualStyleBackColor = false;  this.threeButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // minusButton  //  this.minusButton.BackColor = System.Drawing.Color.White;  this.minusButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.minusButton.ForeColor = System.Drawing.Color.Navy;  this.minusButton.Location = new System.Drawing.Point(151, 184);  this.minusButton.Margin = new System.Windows.Forms.Padding(4);  this.minusButton.Name = "minusButton";  this.minusButton.Size = new System.Drawing.Size(41, 35);  this.minusButton.TabIndex = 6;  this.minusButton.Text = "-";  this.minusButton.UseVisualStyleBackColor = false;  this.minusButton.Click += new System.EventHandler(this.operationButton\_Click);  //  // fourButton  //  this.fourButton.BackColor = System.Drawing.Color.White;  this.fourButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.fourButton.ForeColor = System.Drawing.Color.Navy;  this.fourButton.Location = new System.Drawing.Point(4, 139);  this.fourButton.Margin = new System.Windows.Forms.Padding(4);  this.fourButton.Name = "fourButton";  this.fourButton.Size = new System.Drawing.Size(41, 35);  this.fourButton.TabIndex = 7;  this.fourButton.Text = "4";  this.fourButton.UseVisualStyleBackColor = false;  this.fourButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // fiveButton  //  this.fiveButton.BackColor = System.Drawing.Color.White;  this.fiveButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.fiveButton.ForeColor = System.Drawing.Color.Navy;  this.fiveButton.Location = new System.Drawing.Point(53, 139);  this.fiveButton.Margin = new System.Windows.Forms.Padding(4);  this.fiveButton.Name = "fiveButton";  this.fiveButton.Size = new System.Drawing.Size(41, 35);  this.fiveButton.TabIndex = 8;  this.fiveButton.Text = "5";  this.fiveButton.UseVisualStyleBackColor = false;  this.fiveButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // sixButton  //  this.sixButton.BackColor = System.Drawing.Color.White;  this.sixButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.sixButton.ForeColor = System.Drawing.Color.Navy;  this.sixButton.Location = new System.Drawing.Point(102, 139);  this.sixButton.Margin = new System.Windows.Forms.Padding(4);  this.sixButton.Name = "sixButton";  this.sixButton.Size = new System.Drawing.Size(41, 35);  this.sixButton.TabIndex = 9;  this.sixButton.Text = "6";  this.sixButton.UseVisualStyleBackColor = false;  this.sixButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // multiplyButton  //  this.multiplyButton.BackColor = System.Drawing.Color.White;  this.multiplyButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.multiplyButton.ForeColor = System.Drawing.Color.Navy;  this.multiplyButton.Location = new System.Drawing.Point(151, 139);  this.multiplyButton.Margin = new System.Windows.Forms.Padding(4);  this.multiplyButton.Name = "multiplyButton";  this.multiplyButton.Size = new System.Drawing.Size(41, 35);  this.multiplyButton.TabIndex = 10;  this.multiplyButton.Text = "\*";  this.multiplyButton.UseVisualStyleBackColor = false;  this.multiplyButton.Click += new System.EventHandler(this.operationButton\_Click);  //  // sevenButton  //  this.sevenButton.BackColor = System.Drawing.Color.White;  this.sevenButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.sevenButton.ForeColor = System.Drawing.Color.Navy;  this.sevenButton.Location = new System.Drawing.Point(4, 94);  this.sevenButton.Margin = new System.Windows.Forms.Padding(4);  this.sevenButton.Name = "sevenButton";  this.sevenButton.Size = new System.Drawing.Size(41, 35);  this.sevenButton.TabIndex = 11;  this.sevenButton.Text = "7";  this.sevenButton.UseVisualStyleBackColor = false;  this.sevenButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // eightButton  //  this.eightButton.BackColor = System.Drawing.Color.White;  this.eightButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.eightButton.ForeColor = System.Drawing.Color.Navy;  this.eightButton.Location = new System.Drawing.Point(53, 94);  this.eightButton.Margin = new System.Windows.Forms.Padding(4);  this.eightButton.Name = "eightButton";  this.eightButton.Size = new System.Drawing.Size(41, 35);  this.eightButton.TabIndex = 12;  this.eightButton.Text = "8";  this.eightButton.UseVisualStyleBackColor = false;  this.eightButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // nineButton  //  this.nineButton.BackColor = System.Drawing.Color.White;  this.nineButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.nineButton.ForeColor = System.Drawing.Color.Navy;  this.nineButton.Location = new System.Drawing.Point(102, 94);  this.nineButton.Margin = new System.Windows.Forms.Padding(4);  this.nineButton.Name = "nineButton";  this.nineButton.Size = new System.Drawing.Size(41, 35);  this.nineButton.TabIndex = 13;  this.nineButton.Text = "9";  this.nineButton.UseVisualStyleBackColor = false;  this.nineButton.Click += new System.EventHandler(this.numberButton\_Click);  //  // divideButton  //  this.divideButton.BackColor = System.Drawing.Color.White;  this.divideButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.divideButton.ForeColor = System.Drawing.Color.Navy;  this.divideButton.Location = new System.Drawing.Point(151, 94);  this.divideButton.Margin = new System.Windows.Forms.Padding(4);  this.divideButton.Name = "divideButton";  this.divideButton.Size = new System.Drawing.Size(41, 35);  this.divideButton.TabIndex = 14;  this.divideButton.Text = "/";  this.divideButton.UseVisualStyleBackColor = false;  this.divideButton.Click += new System.EventHandler(this.operationButton\_Click);  //  // equalsButton  //  this.equalsButton.Anchor = ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Bottom)  | System.Windows.Forms.AnchorStyles.Left)  | System.Windows.Forms.AnchorStyles.Right)));  this.equalsButton.BackColor = System.Drawing.Color.White;  this.equalsButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.equalsButton.ForeColor = System.Drawing.Color.Navy;  this.equalsButton.Location = new System.Drawing.Point(209, 283);  this.equalsButton.Margin = new System.Windows.Forms.Padding(4);  this.equalsButton.Name = "equalsButton";  this.equalsButton.Size = new System.Drawing.Size(38, 80);  this.equalsButton.TabIndex = 15;  this.equalsButton.Text = "=";  this.equalsButton.UseVisualStyleBackColor = false;  this.equalsButton.Click += new System.EventHandler(this.equalsButton\_Click);  //  // cButton  //  this.cButton.BackColor = System.Drawing.Color.White;  this.cButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.cButton.ForeColor = System.Drawing.Color.Navy;  this.cButton.Location = new System.Drawing.Point(102, 49);  this.cButton.Margin = new System.Windows.Forms.Padding(4);  this.cButton.Name = "cButton";  this.cButton.Size = new System.Drawing.Size(41, 35);  this.cButton.TabIndex = 16;  this.cButton.Text = "C";  this.cButton.UseVisualStyleBackColor = false;  this.cButton.Click += new System.EventHandler(this.cButton\_Click);  //  // ceButton  //  this.ceButton.BackColor = System.Drawing.Color.White;  this.ceButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.ceButton.ForeColor = System.Drawing.Color.Navy;  this.ceButton.Location = new System.Drawing.Point(53, 49);  this.ceButton.Margin = new System.Windows.Forms.Padding(4);  this.ceButton.Name = "ceButton";  this.ceButton.Size = new System.Drawing.Size(41, 35);  this.ceButton.TabIndex = 17;  this.ceButton.Text = "CE";  this.ceButton.UseVisualStyleBackColor = false;  this.ceButton.Click += new System.EventHandler(this.ceButton\_Click);  //  // eraseButton  //  this.eraseButton.BackColor = System.Drawing.Color.White;  this.eraseButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.eraseButton.ForeColor = System.Drawing.Color.Navy;  this.eraseButton.Location = new System.Drawing.Point(3, 48);  this.eraseButton.Name = "eraseButton";  this.eraseButton.Size = new System.Drawing.Size(43, 35);  this.eraseButton.TabIndex = 20;  this.eraseButton.Text = "⇐";  this.eraseButton.UseVisualStyleBackColor = false;  this.eraseButton.Click += new System.EventHandler(this.eraseButton\_Click);  //  // PlusMinusButton  //  this.PlusMinusButton.BackColor = System.Drawing.Color.White;  this.PlusMinusButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.PlusMinusButton.ForeColor = System.Drawing.Color.Navy;  this.PlusMinusButton.Location = new System.Drawing.Point(151, 49);  this.PlusMinusButton.Margin = new System.Windows.Forms.Padding(4);  this.PlusMinusButton.Name = "PlusMinusButton";  this.PlusMinusButton.Size = new System.Drawing.Size(41, 35);  this.PlusMinusButton.TabIndex = 21;  this.PlusMinusButton.Text = "±";  this.PlusMinusButton.UseVisualStyleBackColor = false;  this.PlusMinusButton.Click += new System.EventHandler(this.PlusMinusButton\_Click);  //  // percentButton  //  this.percentButton.BackColor = System.Drawing.Color.White;  this.percentButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.percentButton.ForeColor = System.Drawing.Color.Navy;  this.percentButton.Location = new System.Drawing.Point(200, 94);  this.percentButton.Margin = new System.Windows.Forms.Padding(4);  this.percentButton.Name = "percentButton";  this.percentButton.Size = new System.Drawing.Size(41, 35);  this.percentButton.TabIndex = 22;  this.percentButton.Text = "%";  this.percentButton.UseVisualStyleBackColor = false;  this.percentButton.Click += new System.EventHandler(this.percentButton\_Click);  //  // fracButton  //  this.fracButton.BackColor = System.Drawing.Color.White;  this.fracButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.fracButton.ForeColor = System.Drawing.Color.Navy;  this.fracButton.Location = new System.Drawing.Point(200, 139);  this.fracButton.Margin = new System.Windows.Forms.Padding(4);  this.fracButton.Name = "fracButton";  this.fracButton.Size = new System.Drawing.Size(41, 35);  this.fracButton.TabIndex = 23;  this.fracButton.Text = "1/x";  this.fracButton.UseVisualStyleBackColor = false;  this.fracButton.Click += new System.EventHandler(this.fracButton\_Click);  //  // MCButton  //  this.MCButton.BackColor = System.Drawing.Color.White;  this.MCButton.Font = new System.Drawing.Font("Times New Roman", 11.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.MCButton.ForeColor = System.Drawing.Color.Navy;  this.MCButton.Location = new System.Drawing.Point(4, 4);  this.MCButton.Margin = new System.Windows.Forms.Padding(4);  this.MCButton.Name = "MCButton";  this.MCButton.Size = new System.Drawing.Size(41, 35);  this.MCButton.TabIndex = 24;  this.MCButton.Text = "MC";  this.MCButton.UseVisualStyleBackColor = false;  this.MCButton.Click += new System.EventHandler(this.powerButton\_Click);  //  // sqrtButton  //  this.sqrtButton.BackColor = System.Drawing.Color.White;  this.sqrtButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.sqrtButton.ForeColor = System.Drawing.Color.Navy;  this.sqrtButton.Location = new System.Drawing.Point(200, 49);  this.sqrtButton.Margin = new System.Windows.Forms.Padding(4);  this.sqrtButton.Name = "sqrtButton";  this.sqrtButton.Size = new System.Drawing.Size(41, 35);  this.sqrtButton.TabIndex = 25;  this.sqrtButton.Text = "√";  this.sqrtButton.UseVisualStyleBackColor = false;  this.sqrtButton.Click += new System.EventHandler(this.sqrtButton\_Click);  //  // MRButton  //  this.MRButton.BackColor = System.Drawing.Color.White;  this.MRButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.MRButton.ForeColor = System.Drawing.Color.Navy;  this.MRButton.Location = new System.Drawing.Point(53, 4);  this.MRButton.Margin = new System.Windows.Forms.Padding(4);  this.MRButton.Name = "MRButton";  this.MRButton.Size = new System.Drawing.Size(41, 35);  this.MRButton.TabIndex = 26;  this.MRButton.Text = "MR";  this.MRButton.UseVisualStyleBackColor = false;  this.MRButton.Click += new System.EventHandler(this.MRButton\_Click);  //  // MSButton  //  this.MSButton.BackColor = System.Drawing.Color.White;  this.MSButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.MSButton.ForeColor = System.Drawing.Color.Navy;  this.MSButton.Location = new System.Drawing.Point(102, 4);  this.MSButton.Margin = new System.Windows.Forms.Padding(4);  this.MSButton.Name = "MSButton";  this.MSButton.Size = new System.Drawing.Size(41, 35);  this.MSButton.TabIndex = 27;  this.MSButton.Text = "MS";  this.MSButton.UseVisualStyleBackColor = false;  this.MSButton.Click += new System.EventHandler(this.MSButton\_Click\_1);  //  // MPlusButton  //  this.MPlusButton.BackColor = System.Drawing.Color.White;  this.MPlusButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.MPlusButton.ForeColor = System.Drawing.Color.Navy;  this.MPlusButton.Location = new System.Drawing.Point(151, 4);  this.MPlusButton.Margin = new System.Windows.Forms.Padding(4);  this.MPlusButton.Name = "MPlusButton";  this.MPlusButton.Size = new System.Drawing.Size(41, 35);  this.MPlusButton.TabIndex = 28;  this.MPlusButton.Text = "M+";  this.MPlusButton.UseVisualStyleBackColor = false;  this.MPlusButton.Click += new System.EventHandler(this.button1\_Click);  //  // MMinusButton  //  this.MMinusButton.BackColor = System.Drawing.Color.White;  this.MMinusButton.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.MMinusButton.ForeColor = System.Drawing.Color.Navy;  this.MMinusButton.Location = new System.Drawing.Point(200, 4);  this.MMinusButton.Margin = new System.Windows.Forms.Padding(4);  this.MMinusButton.Name = "MMinusButton";  this.MMinusButton.Size = new System.Drawing.Size(41, 35);  this.MMinusButton.TabIndex = 29;  this.MMinusButton.Text = "M-";  this.MMinusButton.UseVisualStyleBackColor = false;  this.MMinusButton.Click += new System.EventHandler(this.MMinusButton\_Click);  //  // tableLayoutPanel1  //  this.tableLayoutPanel1.Anchor = ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Bottom)  | System.Windows.Forms.AnchorStyles.Left)  | System.Windows.Forms.AnchorStyles.Right)));  this.tableLayoutPanel1.AutoSizeMode = System.Windows.Forms.AutoSizeMode.GrowAndShrink;  this.tableLayoutPanel1.ColumnCount = 5;  this.tableLayoutPanel1.ColumnStyles.Add(new System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 20F));  this.tableLayoutPanel1.ColumnStyles.Add(new System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 20F));  this.tableLayoutPanel1.ColumnStyles.Add(new System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 20F));  this.tableLayoutPanel1.ColumnStyles.Add(new System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 20F));  this.tableLayoutPanel1.ColumnStyles.Add(new System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 20F));  this.tableLayoutPanel1.Controls.Add(this.MCButton, 0, 0);  this.tableLayoutPanel1.Controls.Add(this.MRButton, 1, 0);  this.tableLayoutPanel1.Controls.Add(this.MSButton, 2, 0);  this.tableLayoutPanel1.Controls.Add(this.fracButton, 4, 3);  this.tableLayoutPanel1.Controls.Add(this.plusButton, 3, 5);  this.tableLayoutPanel1.Controls.Add(this.minusButton, 3, 4);  this.tableLayoutPanel1.Controls.Add(this.decimalButton, 2, 5);  this.tableLayoutPanel1.Controls.Add(this.sqrtButton, 4, 1);  this.tableLayoutPanel1.Controls.Add(this.threeButton, 2, 4);  this.tableLayoutPanel1.Controls.Add(this.percentButton, 4, 2);  this.tableLayoutPanel1.Controls.Add(this.twoButton, 1, 4);  this.tableLayoutPanel1.Controls.Add(this.multiplyButton, 3, 3);  this.tableLayoutPanel1.Controls.Add(this.oneButton, 0, 4);  this.tableLayoutPanel1.Controls.Add(this.MMinusButton, 4, 0);  this.tableLayoutPanel1.Controls.Add(this.sixButton, 2, 3);  this.tableLayoutPanel1.Controls.Add(this.MPlusButton, 3, 0);  this.tableLayoutPanel1.Controls.Add(this.fiveButton, 1, 3);  this.tableLayoutPanel1.Controls.Add(this.divideButton, 3, 2);  this.tableLayoutPanel1.Controls.Add(this.fourButton, 0, 3);  this.tableLayoutPanel1.Controls.Add(this.eraseButton, 0, 1);  this.tableLayoutPanel1.Controls.Add(this.nineButton, 2, 2);  this.tableLayoutPanel1.Controls.Add(this.PlusMinusButton, 3, 1);  this.tableLayoutPanel1.Controls.Add(this.eightButton, 1, 2);  this.tableLayoutPanel1.Controls.Add(this.ceButton, 1, 1);  this.tableLayoutPanel1.Controls.Add(this.sevenButton, 0, 2);  this.tableLayoutPanel1.Controls.Add(this.cButton, 2, 1);  this.tableLayoutPanel1.Font = new System.Drawing.Font("Microsoft Sans Serif", 21.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.tableLayoutPanel1.Location = new System.Drawing.Point(9, 99);  this.tableLayoutPanel1.Name = "tableLayoutPanel1";  this.tableLayoutPanel1.RowCount = 6;  this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 16.66667F));  this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 16.66667F));  this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 16.66667F));  this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 16.66667F));  this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 16.66667F));  this.tableLayoutPanel1.RowStyles.Add(new System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 16.66667F));  this.tableLayoutPanel1.Size = new System.Drawing.Size(245, 274);  this.tableLayoutPanel1.TabIndex = 31;  //  // checkBox1  //  this.checkBox1.AutoSize = true;  this.checkBox1.ForeColor = System.Drawing.Color.Black;  this.checkBox1.Location = new System.Drawing.Point(9, 5);  this.checkBox1.Name = "checkBox1";  this.checkBox1.Size = new System.Drawing.Size(119, 19);  this.checkBox1.TabIndex = 32;  this.checkBox1.Text = "Дополнительно";  this.checkBox1.UseVisualStyleBackColor = true;  this.checkBox1.CheckedChanged += new System.EventHandler(this.checkBox1\_CheckedChanged);  //  // calculatorScreen  //  this.calculatorScreen.AcceptsTab = true;  this.calculatorScreen.BackColor = System.Drawing.Color.White;  this.calculatorScreen.Font = new System.Drawing.Font("Microsoft Sans Serif", 26.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.calculatorScreen.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(0)))), ((int)(((byte)(0)))), ((int)(((byte)(0)))), ((int)(((byte)(128)))));  this.calculatorScreen.Location = new System.Drawing.Point(13, 31);  this.calculatorScreen.Margin = new System.Windows.Forms.Padding(4);  this.calculatorScreen.MaximumSize = new System.Drawing.Size(245, 60);  this.calculatorScreen.MinimumSize = new System.Drawing.Size(230, 60);  this.calculatorScreen.Name = "calculatorScreen";  this.calculatorScreen.ReadOnly = true;  this.calculatorScreen.Size = new System.Drawing.Size(237, 47);  this.calculatorScreen.TabIndex = 1;  this.calculatorScreen.Text = "0";  this.calculatorScreen.TextAlign = System.Windows.Forms.HorizontalAlignment.Right;  this.calculatorScreen.WordWrap = false;  this.calculatorScreen.TextChanged += new System.EventHandler(this.calculatorScreen\_TextChanged);  //  // panel1  //  this.panel1.Controls.Add(this.checkBox1);  this.panel1.Controls.Add(this.zeroButton);  this.panel1.Controls.Add(this.equalsButton);  this.panel1.Controls.Add(this.calculatorScreen);  this.panel1.Controls.Add(this.tableLayoutPanel1);  this.panel1.Location = new System.Drawing.Point(8, 9);  this.panel1.Name = "panel1";  this.panel1.Size = new System.Drawing.Size(265, 391);  this.panel1.TabIndex = 33;  //  // panel2  //  this.panel2.BackColor = System.Drawing.Color.White;  this.panel2.Controls.Add(this.label3);  this.panel2.Controls.Add(this.label2);  this.panel2.Controls.Add(this.label1);  this.panel2.Controls.Add(this.textBox2);  this.panel2.Controls.Add(this.textBox1);  this.panel2.Controls.Add(this.comboBox3);  this.panel2.Controls.Add(this.comboBox2);  this.panel2.Controls.Add(this.comboBox1);  this.panel2.Location = new System.Drawing.Point(279, 40);  this.panel2.Name = "panel2";  this.panel2.Size = new System.Drawing.Size(389, 332);  this.panel2.TabIndex = 34;  //  // label3  //  this.label3.AutoSize = true;  this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.label3.ForeColor = System.Drawing.Color.Navy;  this.label3.Location = new System.Drawing.Point(20, 217);  this.label3.Name = "label3";  this.label3.Size = new System.Drawing.Size(17, 16);  this.label3.TabIndex = 7;  this.label3.Text = "В";  this.label3.Click += new System.EventHandler(this.label3\_Click);  //  // label2  //  this.label2.AutoSize = true;  this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.label2.ForeColor = System.Drawing.Color.Navy;  this.label2.Location = new System.Drawing.Point(20, 116);  this.label2.Name = "label2";  this.label2.Size = new System.Drawing.Size(27, 16);  this.label2.TabIndex = 6;  this.label2.Text = "Из";  this.label2.Click += new System.EventHandler(this.label2\_Click);  //  // label1  //  this.label1.AutoSize = true;  this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.label1.ForeColor = System.Drawing.Color.Navy;  this.label1.Location = new System.Drawing.Point(20, 33);  this.label1.Name = "label1";  this.label1.Size = new System.Drawing.Size(300, 16);  this.label1.TabIndex = 5;  this.label1.Text = "Выберите тип преобразуемой единицы";  //  // textBox2  //  this.textBox2.BackColor = System.Drawing.SystemColors.ScrollBar;  this.textBox2.Location = new System.Drawing.Point(20, 147);  this.textBox2.Name = "textBox2";  this.textBox2.Size = new System.Drawing.Size(354, 21);  this.textBox2.TabIndex = 4;  //  // textBox1  //  this.textBox1.BackColor = System.Drawing.SystemColors.ScrollBar;  this.textBox1.Location = new System.Drawing.Point(20, 252);  this.textBox1.Name = "textBox1";  this.textBox1.ReadOnly = true;  this.textBox1.Size = new System.Drawing.Size(354, 21);  this.textBox1.TabIndex = 3;  //  // comboBox3  //  this.comboBox3.BackColor = System.Drawing.SystemColors.ScrollBar;  this.comboBox3.FormattingEnabled = true;  this.comboBox3.Items.AddRange(new object[] {  "inches",  "centimeters"});  this.comboBox3.Location = new System.Drawing.Point(20, 282);  this.comboBox3.Name = "comboBox3";  this.comboBox3.Size = new System.Drawing.Size(354, 23);  this.comboBox3.TabIndex = 2;  this.comboBox3.SelectedIndexChanged += new System.EventHandler(this.ComboBox3\_SelectedIndexChanged);  //  // comboBox2  //  this.comboBox2.BackColor = System.Drawing.SystemColors.ScrollBar;  this.comboBox2.FormattingEnabled = true;  this.comboBox2.Items.AddRange(new object[] {  "inches",  "centimeters"});  this.comboBox2.Location = new System.Drawing.Point(20, 174);  this.comboBox2.Name = "comboBox2";  this.comboBox2.Size = new System.Drawing.Size(354, 23);  this.comboBox2.TabIndex = 1;  this.comboBox2.SelectedIndexChanged += new System.EventHandler(this.ComboBox2\_SelectedIndexChanged);  //  // comboBox1  //  this.comboBox1.BackColor = System.Drawing.SystemColors.ScrollBar;  this.comboBox1.FormattingEnabled = true;  this.comboBox1.Items.AddRange(new object[] {  "length"});  this.comboBox1.Location = new System.Drawing.Point(20, 72);  this.comboBox1.Name = "comboBox1";  this.comboBox1.Size = new System.Drawing.Size(354, 23);  this.comboBox1.TabIndex = 0;  this.comboBox1.SelectedIndexChanged += new System.EventHandler(this.ComboBox1\_SelectedIndexChanged);  //  // button1  //  this.button1.ForeColor = System.Drawing.Color.Black;  this.button1.Location = new System.Drawing.Point(537, 377);  this.button1.Name = "button1";  this.button1.Size = new System.Drawing.Size(116, 23);  this.button1.TabIndex = 35;  this.button1.Text = "Конвертировать";  this.button1.UseVisualStyleBackColor = true;  this.button1.Click += new System.EventHandler(this.button1\_Click\_1);  //  // Form1  //  this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);  this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;  this.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(230)))), ((int)(((byte)(238)))), ((int)(((byte)(248)))));  this.ClientSize = new System.Drawing.Size(280, 407);  this.Controls.Add(this.button1);  this.Controls.Add(this.panel2);  this.Controls.Add(this.panel1);  this.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));  this.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(230)))), ((int)(((byte)(238)))), ((int)(((byte)(248)))));  this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.Fixed3D;  this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));  this.Margin = new System.Windows.Forms.Padding(4);  this.MaximizeBox = false;  this.Name = "Form1";  this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;  this.Text = "Калькулятор";  this.Load += new System.EventHandler(this.CalculatorUI\_Load);  this.tableLayoutPanel1.ResumeLayout(false);  this.panel1.ResumeLayout(false);  this.panel1.PerformLayout();  this.panel2.ResumeLayout(false);  this.panel2.PerformLayout();  this.ResumeLayout(false);  }    private System.Windows.Forms.Button zeroButton;  private System.Windows.Forms.Button decimalButton;  private System.Windows.Forms.Button plusButton;  private System.Windows.Forms.Button oneButton;  private System.Windows.Forms.Button twoButton;  private System.Windows.Forms.Button threeButton;  private System.Windows.Forms.Button minusButton;  private System.Windows.Forms.Button fourButton;  private System.Windows.Forms.Button fiveButton;  private System.Windows.Forms.Button sixButton;  private System.Windows.Forms.Button multiplyButton;  private System.Windows.Forms.Button sevenButton;  private System.Windows.Forms.Button eightButton;  private System.Windows.Forms.Button nineButton;  private System.Windows.Forms.Button divideButton;  private System.Windows.Forms.Button equalsButton;  private System.Windows.Forms.Button cButton;  private System.Windows.Forms.Button ceButton;  private System.Windows.Forms.Button eraseButton;  private System.Windows.Forms.Button PlusMinusButton;  private System.Windows.Forms.Button percentButton;  private System.Windows.Forms.Button fracButton;  private System.Windows.Forms.Button MCButton;  private System.Windows.Forms.Button sqrtButton;  private System.Windows.Forms.Button MRButton;  private System.Windows.Forms.Button MSButton;  private System.Windows.Forms.Button MPlusButton;  private System.Windows.Forms.Button MMinusButton;  private System.Windows.Forms.TableLayoutPanel tableLayoutPanel1;  private System.Windows.Forms.CheckBox checkBox1;  private System.Windows.Forms.TextBox calculatorScreen;  private System.Windows.Forms.Panel panel1;  private System.Windows.Forms.Panel panel2;  private System.Windows.Forms.ComboBox comboBox1;  private System.Windows.Forms.TextBox textBox2;  private System.Windows.Forms.TextBox textBox1;  private System.Windows.Forms.ComboBox comboBox3;  private System.Windows.Forms.ComboBox comboBox2;  private System.Windows.Forms.Label label1;  private System.Windows.Forms.Label label2;  private System.Windows.Forms.Label label3;  private System.Windows.Forms.Button button1;  }  } |

Контрольные вопросы:

1. Особенности построения приложений на C#?

Приложения на C# в Windows Forms имеют следующие особенности:

Графический интерфейс разработки с использованием Visual Studio

Поддержка объектно-ориентированного программирования

Интеграция с .NET Framework и библиотеками WinForms

Возможность создания пользовательских элементов управления

Поддержка обработки событий для взаимодействия с пользователем

Удобство разработки благодаря готовым компонентам и шаблонам

1. Для чего используется компонент Label;

Компонент Label используется для отображения текста или изображений на форме. Он позволяет:

Показывать информацию для пользователя

Отображать статус или состояние приложения

Использоваться как заголовок формы или раздела

Добавлять краткие инструкции или пояснения

1. Для чего используется компонент Button;

Компонент Button используется для создания кнопок, которые выполняют действия при нажатии. Он позволяет:

Выполнять различные функции приложения

Переключать между различными экранами или разделами

Отправлять данные на сервер или базу данных

Закрывать форму или приложение

1. Для чего используется компонент CheckBox;

Компонент CheckBox используется для создания переключательных элементов управления. Он позволяет:

Предоставить пользователю возможность выбрать один из нескольких вариантов

Сохранять состояние выбора между запусками приложения

Использоваться для простых "включено/выключено" опций

1. Для чего используется компонент ErrorProvider;

Компонент ErrorProvider используется для отображения ошибок или предупреждений. Он позволяет:

Показывать информацию об ошибках без прерывания потока ввода пользователя

Помочь пользователю исправить ошибки

Отобразить сообщения об ошибках рядом с соответствующими элементами управления

1. Для чего используется компонент ListBox;

Компонент ListBox используется для отображения списка элементов. Он позволяет:

Предоставить пользователю выбор из списка вариантов

Отобразить данные из базы данных или других источников

Реализовать функционал выбора одного или нескольких элементов

1. Для чего используется компонент TabControl;

Компонент TabControl используется для создания вкладочных интерфейсов. Он позволяет:

Разделить форму на несколько панелей

Переключаться между различными разделами информации

Организовать сложные формы с множеством элементов управления

1. Для чего используется компонент Panel;

Компонент Panel используется для группировки элементов управления. Он позволяет:

Организовать элементы управления в логических группах

Повторно использовать одинаковые группы элементов в разных частях формы

Улучшить читаемость и структуру формы

1. Для чего используется компонент RadioButton;

Компонент RadioButton используется для создания радиокнопок. Он позволяет:

Предоставить пользователю возможность выбрать один вариант из нескольких

Сохранять состояние выбора между запусками приложения

Использоваться для простых опций "выберите одно"

1. Какие компоненты используются для создания меню на форме;

Для создания меню на форме в Windows Forms используются следующие компоненты:

MenuStrip - основной компонент для создания верхнего меню

ToolStripMenuItem - компонент для создания пунктов меню

ContextMenuStrip - компонент для создания контекстного мен

**ЛАБОРАТОРНАЯ РАБОТА №3**

**Работа с массивами данных в C# с использованием WindowsForms**

**Цель**: Цель ознакомиться с компонентами и методами работы с массивами в С# c использованием Windows Form в простых Windows приложениях.

**Ход работы**

Моя задача: дана матрица размера M × N. Поменять местами столбец с номером N и последний из столбцов, содержащих только положительные элементы. Если требуемых столбцов нет, то вывести матрицу без изменений.

На рисунке 3.1 представлен интерфейс моей программной реализации.

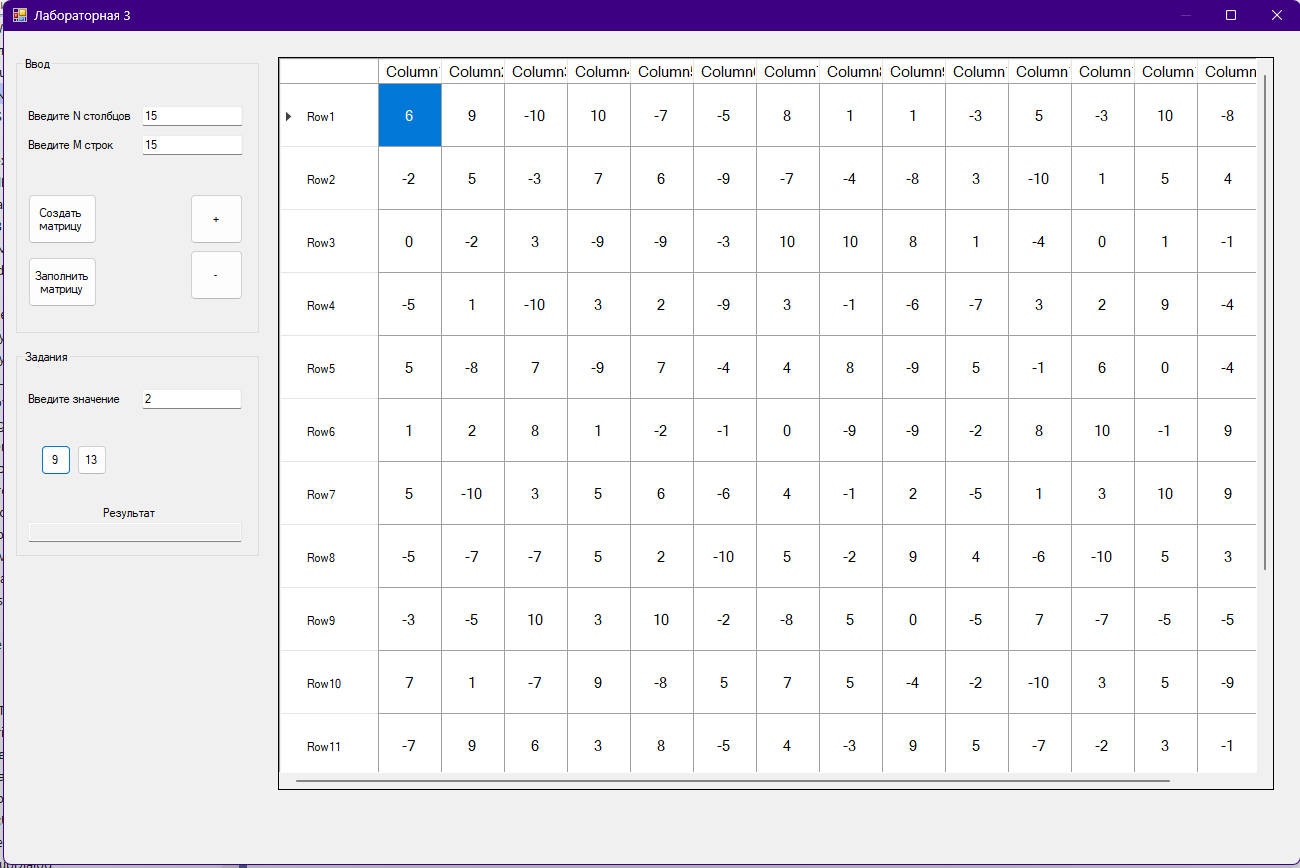


Рисунок 3.1 – Интерфейс программы

Дополнительно для себя я реализовал 13 вариант задания.

Ниже представлен листинг программы:

Form1.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.Common;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using static System.Windows.Forms.VisualStyles.VisualStyleElement;  namespace ThirdLaboratory  {  public partial class Form1 : Form  {  private int m;  private int n;  public Form1()  {  InitializeComponent();    this.ClientSize = new System.Drawing.Size(500, 500); // Устанавливаем стартовые размеры формы    // Добавляем обработчик события Resize для формы  this.Resize += new EventHandler(Form1\_Resize);  }  private void Form1\_Resize(object sender, EventArgs e)  {  dataGridView1.Size = new System.Drawing.Size(this.ClientSize.Width - 300, this.ClientSize.Height - 100); // Устанавливаем новые размеры dataGridView1 пропорционально размеру формы  }  private void createMatrixButton\_Click(object sender, EventArgs e)  {  if (!int.TryParse(mLinesBox.Text, out m) || !int.TryParse(nLinesBox.Text, out n))  {  MessageBox.Show("Пожалуйста, введите корректные значения для m и n.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  dataGridView1.Rows.Clear();  dataGridView1.Columns.Clear();  // Добавляем столбцы в dataGridView1  for (int j = 0; j < n; j++)  {  string columnName = $"Column{j + 1}";  dataGridView1.Columns.Add(columnName, columnName);  }  // Добавляем строки в dataGridView1  for (int i = 0; i < m; i++)  {  DataGridViewRow row = new DataGridViewRow();  row.CreateCells(dataGridView1);  // Устанавливаем название строки  row.HeaderCell.Value = $"Row{i + 1}";  // Устанавливаем высоту строки  row.Height = 60;  // Добавляем новую строку в DataGridView  dataGridView1.Rows.Add(row);  }  // Настройка размеров столбцов  foreach (DataGridViewColumn column in dataGridView1.Columns)  {  column.Width = 60;  }  // Устанавливаем ширину заголовков  dataGridView1.RowHeadersWidth = 100;  // Устанавливаем стиль ячеек для выравнивания текста по центру  dataGridView1.DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter;  // Разрешаем редактирование ячеек в dataGridView1  dataGridView1.AllowUserToAddRows = false;  dataGridView1.AllowUserToDeleteRows = false;  dataGridView1.ReadOnly = false;  }  private void randowButton\_Click(object sender, EventArgs e)  {  Random rand = new Random();  for (int i = 0; i < dataGridView1.Rows.Count; i++)  {  for (int j = 0; j < dataGridView1.Columns.Count; j++)  {  int randomNumber = rand.Next(-10, 11); // Генерирует случайное число от -10 до 10  dataGridView1.Rows[i].Cells[j].Value = randomNumber;  }  }  }  private void Form1\_Load(object sender, EventArgs e)  {  }  private void plusSizeButton\_Click(object sender, EventArgs e)  {  float currentSize = dataGridView1.DefaultCellStyle.Font.Size;  dataGridView1.DefaultCellStyle.Font = new Font(dataGridView1.DefaultCellStyle.Font.FontFamily, currentSize + 3);  dataGridView1.ColumnHeadersDefaultCellStyle.Font = new Font(dataGridView1.ColumnHeadersDefaultCellStyle.Font.FontFamily, currentSize + 3);  // Увеличиваем ширину и высоту ячеек  foreach (DataGridViewColumn column in dataGridView1.Columns)  {  column.Width += 3;  }  foreach (DataGridViewRow row in dataGridView1.Rows)  {  row.Height += 3;  }  }  private void minusSizeButton\_Click(object sender, EventArgs e)  {  float currentSize = dataGridView1.DefaultCellStyle.Font.Size;  dataGridView1.DefaultCellStyle.Font = new Font(dataGridView1.DefaultCellStyle.Font.FontFamily, currentSize - 3);  dataGridView1.ColumnHeadersDefaultCellStyle.Font = new Font(dataGridView1.ColumnHeadersDefaultCellStyle.Font.FontFamily, currentSize - 3);  // Увеличиваем ширину и высоту ячеек  foreach (DataGridViewColumn column in dataGridView1.Columns)  {  column.Width -= 3;  }  foreach (DataGridViewRow row in dataGridView1.Rows)  {  row.Height -= 3;  }  }  private void button1\_Click(object sender, EventArgs e)  {  int n = 0;  if (!int.TryParse(textBox1.Text, out n) || n <= 0 || n > dataGridView1.Columns.Count)  {  MessageBox.Show("Пожалуйста, введите корректный номер столбца (от 1 до " + dataGridView1.Columns.Count + ").", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  int lastPositiveColumnIndex = FindLastPositiveColumn();  if (lastPositiveColumnIndex != -1 && lastPositiveColumnIndex < dataGridView1.Columns.Count)  {  try  {  SwapColumns(n - 1, lastPositiveColumnIndex); // Уменьшаем индекс на 1, так как индексы в DataGridView начинаются с 0  }  catch (Exception ex)  {  MessageBox.Show($"Произошла ошибка при обмене столбцов: {ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  private int FindLastPositiveColumn()  {  for (int i = dataGridView1.Columns.Count - 1; i >= 0; i--)  {  bool allPositive = true;  foreach (DataGridViewRow row in dataGridView1.Rows)  {  DataGridViewCell cell = row.Cells[i];  if (cell.Value != null && Convert.ToDouble(cell.Value) <= 0)  {  allPositive = false;  break;  }  }  if (allPositive)  {  return i;  }  }  return -1; // Если нет столбцов с только положительными элементами  }  private void SwapColumns(int column1Index, int column2Index)  {  // Создаем списки для хранения значений  List<object> values1 = new List<object>();  List<object> values2 = new List<object>();  // Заполняем списки значениями из первых двух столбцов  for (int i = 0; i < dataGridView1.Rows.Count; i++)  {  values1.Add(dataGridView1.Rows[i].Cells[column1Index].Value);  values2.Add(dataGridView1.Rows[i].Cells[column2Index].Value);  }  // Обмениваем значения между списками    // Заменяем значения в ячейках  for (int i = 0; i < values1.Count; i++)  {  if (values1[i] != null)  {  dataGridView1.Rows[i].Cells[column1Index].Value = values2[i];  }  }  for (int i = 0; i < values2.Count; i++)  {  if (values2[i] != null)  {  dataGridView1.Rows[i].Cells[column2Index].Value = values1[i];  }  }  }  private void label3\_Click(object sender, EventArgs e)  {  }  private void task13Button\_Click(object sender, EventArgs e)  {  // Проверяем, что матрица создана  if (dataGridView1.Rows.Count == 0 || dataGridView1.Columns.Count == 0)  {  MessageBox.Show("Матрица не была создана. Пожалуйста, создайте матрицу с помощью кнопки \"Создать матрицу\".", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  // Получаем количество строк из DataGridView  int m = dataGridView1.Rows.Count;  // Проверяем ввод K  int k;  while (!int.TryParse(textBox1.Text, out k) || k < 1 || k > m)  {  MessageBox.Show("Пожалуйста, введите корректное значение K (от 1 до " + m + ").", "Ошибка ввода", MessageBoxButtons.OK, MessageBoxIcon.Error);    return;  }  // Получаем значения из K-й строки  string[] rowValues = new string[m];  for (int i = 0; i < m; i++)  {  rowValues[i] = dataGridView1.Rows[k - 1].Cells[i].Value?.ToString() ?? "";  }  // Выводим элементы K-й строки  string result = string.Join(", ", rowValues);  completeTextBox.Text = result ;    }  }  } |

Form1.designer

|  |
| --- |
| namespace ThirdLaboratory  {  partial class Form1  {  /// <summary>  /// Обязательная переменная конструктора.  /// </summary>  private System.ComponentModel.IContainer components = null;  /// <summary>  /// Освободить все используемые ресурсы.  /// </summary>  /// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</param>  protected override void Dispose(bool disposing)  {  if (disposing && (components != null))  {  components.Dispose();  }  base.Dispose(disposing);  }  #region Код, автоматически созданный конструктором форм Windows  /// <summary>  /// Требуемый метод для поддержки конструктора — не изменяйте  /// содержимое этого метода с помощью редактора кода.  /// </summary>  private void InitializeComponent()  {  this.groupBox1 = new System.Windows.Forms.GroupBox();  this.randowButton = new System.Windows.Forms.Button();  this.label2 = new System.Windows.Forms.Label();  this.label1 = new System.Windows.Forms.Label();  this.createMatrixButton = new System.Windows.Forms.Button();  this.nLinesBox = new System.Windows.Forms.TextBox();  this.mLinesBox = new System.Windows.Forms.TextBox();  this.dataGridView1 = new System.Windows.Forms.DataGridView();  this.plusSizeButton = new System.Windows.Forms.Button();  this.minusSizeButton = new System.Windows.Forms.Button();  this.groupBox2 = new System.Windows.Forms.GroupBox();  this.textBox1 = new System.Windows.Forms.TextBox();  this.task9Button = new System.Windows.Forms.Button();  this.label3 = new System.Windows.Forms.Label();  this.completeTextBox = new System.Windows.Forms.TextBox();  this.task13Button = new System.Windows.Forms.Button();  this.label4 = new System.Windows.Forms.Label();  this.groupBox1.SuspendLayout();  ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();  this.groupBox2.SuspendLayout();  this.SuspendLayout();  //  // groupBox1  //  this.groupBox1.Controls.Add(this.minusSizeButton);  this.groupBox1.Controls.Add(this.plusSizeButton);  this.groupBox1.Controls.Add(this.randowButton);  this.groupBox1.Controls.Add(this.label2);  this.groupBox1.Controls.Add(this.label1);  this.groupBox1.Controls.Add(this.createMatrixButton);  this.groupBox1.Controls.Add(this.nLinesBox);  this.groupBox1.Controls.Add(this.mLinesBox);  this.groupBox1.Location = new System.Drawing.Point(12, 26);  this.groupBox1.Name = "groupBox1";  this.groupBox1.Size = new System.Drawing.Size(243, 277);  this.groupBox1.TabIndex = 0;  this.groupBox1.TabStop = false;  this.groupBox1.Text = "Ввод";  //  // randowButton  //  this.randowButton.Location = new System.Drawing.Point(12, 200);  this.randowButton.Name = "randowButton";  this.randowButton.Size = new System.Drawing.Size(69, 50);  this.randowButton.TabIndex = 4;  this.randowButton.Text = "Заполнить матрицу";  this.randowButton.UseVisualStyleBackColor = true;  this.randowButton.Click += new System.EventHandler(this.randowButton\_Click);  //  // label2  //  this.label2.AutoSize = true;  this.label2.Location = new System.Drawing.Point(9, 81);  this.label2.Name = "label2";  this.label2.Size = new System.Drawing.Size(93, 13);  this.label2.TabIndex = 3;  this.label2.Text = "Введите M строк";  //  // label1  //  this.label1.AutoSize = true;  this.label1.Location = new System.Drawing.Point(9, 52);  this.label1.Name = "label1";  this.label1.Size = new System.Drawing.Size(110, 13);  this.label1.TabIndex = 3;  this.label1.Text = "Введите N столбцов";  //  // createMatrixButton  //  this.createMatrixButton.Location = new System.Drawing.Point(12, 137);  this.createMatrixButton.Name = "createMatrixButton";  this.createMatrixButton.Size = new System.Drawing.Size(69, 50);  this.createMatrixButton.TabIndex = 2;  this.createMatrixButton.Text = "Создать матрицу";  this.createMatrixButton.UseVisualStyleBackColor = true;  this.createMatrixButton.Click += new System.EventHandler(this.createMatrixButton\_Click);  //  // nLinesBox  //  this.nLinesBox.Location = new System.Drawing.Point(126, 78);  this.nLinesBox.Name = "nLinesBox";  this.nLinesBox.Size = new System.Drawing.Size(101, 20);  this.nLinesBox.TabIndex = 1;  this.nLinesBox.Text = "3";  //  // mLinesBox  //  this.mLinesBox.Location = new System.Drawing.Point(126, 49);  this.mLinesBox.Name = "mLinesBox";  this.mLinesBox.Size = new System.Drawing.Size(101, 20);  this.mLinesBox.TabIndex = 0;  this.mLinesBox.Text = "3";  //  // dataGridView1  //  this.dataGridView1.AllowUserToOrderColumns = true;  this.dataGridView1.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;  this.dataGridView1.Location = new System.Drawing.Point(274, 26);  this.dataGridView1.Name = "dataGridView1";  this.dataGridView1.Size = new System.Drawing.Size(500, 500);  this.dataGridView1.TabIndex = 1;  //  // plusSizeButton  //  this.plusSizeButton.Location = new System.Drawing.Point(174, 137);  this.plusSizeButton.Name = "plusSizeButton";  this.plusSizeButton.Size = new System.Drawing.Size(53, 50);  this.plusSizeButton.TabIndex = 5;  this.plusSizeButton.Text = "+";  this.plusSizeButton.UseVisualStyleBackColor = true;  this.plusSizeButton.Click += new System.EventHandler(this.plusSizeButton\_Click);  //  // minusSizeButton  //  this.minusSizeButton.Location = new System.Drawing.Point(174, 193);  this.minusSizeButton.Name = "minusSizeButton";  this.minusSizeButton.Size = new System.Drawing.Size(53, 50);  this.minusSizeButton.TabIndex = 5;  this.minusSizeButton.Text = "-";  this.minusSizeButton.UseVisualStyleBackColor = true;  this.minusSizeButton.Click += new System.EventHandler(this.minusSizeButton\_Click);  //  // groupBox2  //  this.groupBox2.Controls.Add(this.label4);  this.groupBox2.Controls.Add(this.task13Button);  this.groupBox2.Controls.Add(this.completeTextBox);  this.groupBox2.Controls.Add(this.task9Button);  this.groupBox2.Controls.Add(this.textBox1);  this.groupBox2.Controls.Add(this.label3);  this.groupBox2.Location = new System.Drawing.Point(12, 319);  this.groupBox2.Name = "groupBox2";  this.groupBox2.Size = new System.Drawing.Size(243, 207);  this.groupBox2.TabIndex = 2;  this.groupBox2.TabStop = false;  this.groupBox2.Text = "Задания";  //  // textBox1  //  this.textBox1.Location = new System.Drawing.Point(126, 39);  this.textBox1.Name = "textBox1";  this.textBox1.Size = new System.Drawing.Size(100, 20);  this.textBox1.TabIndex = 0;  //  // task9Button  //  this.task9Button.Location = new System.Drawing.Point(25, 95);  this.task9Button.Name = "task9Button";  this.task9Button.Size = new System.Drawing.Size(30, 30);  this.task9Button.TabIndex = 1;  this.task9Button.Text = "9";  this.task9Button.UseVisualStyleBackColor = true;  this.task9Button.Click += new System.EventHandler(this.button1\_Click);  //  // label3  //  this.label3.AutoSize = true;  this.label3.Location = new System.Drawing.Point(9, 42);  this.label3.Name = "label3";  this.label3.Size = new System.Drawing.Size(99, 13);  this.label3.TabIndex = 3;  this.label3.Text = "Введите значение";  this.label3.Click += new System.EventHandler(this.label3\_Click);  //  // completeTextBox  //  this.completeTextBox.Location = new System.Drawing.Point(12, 172);  this.completeTextBox.Name = "completeTextBox";  this.completeTextBox.ReadOnly = true;  this.completeTextBox.Size = new System.Drawing.Size(214, 20);  this.completeTextBox.TabIndex = 4;  //  // task13Button  //  this.task13Button.Location = new System.Drawing.Point(61, 95);  this.task13Button.Name = "task13Button";  this.task13Button.Size = new System.Drawing.Size(30, 30);  this.task13Button.TabIndex = 5;  this.task13Button.Text = "13";  this.task13Button.UseVisualStyleBackColor = true;  this.task13Button.Click += new System.EventHandler(this.task13Button\_Click);  //  // label4  //  this.label4.AutoSize = true;  this.label4.Location = new System.Drawing.Point(84, 156);  this.label4.Name = "label4";  this.label4.Size = new System.Drawing.Size(59, 13);  this.label4.TabIndex = 6;  this.label4.Text = "Результат";  //  // Form1  //  this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);  this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;  this.ClientSize = new System.Drawing.Size(784, 561);  this.Controls.Add(this.groupBox2);  this.Controls.Add(this.dataGridView1);  this.Controls.Add(this.groupBox1);  this.MinimizeBox = false;  this.MinimumSize = new System.Drawing.Size(800, 600);  this.Name = "Form1";  this.Text = "Лабораторная 3";  this.Load += new System.EventHandler(this.Form1\_Load);  this.groupBox1.ResumeLayout(false);  this.groupBox1.PerformLayout();  ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();  this.groupBox2.ResumeLayout(false);  this.groupBox2.PerformLayout();  this.ResumeLayout(false);  }  #endregion  private System.Windows.Forms.GroupBox groupBox1;  private System.Windows.Forms.Button createMatrixButton;  private System.Windows.Forms.TextBox nLinesBox;  private System.Windows.Forms.TextBox mLinesBox;  private System.Windows.Forms.DataGridView dataGridView1;  private System.Windows.Forms.Label label2;  private System.Windows.Forms.Label label1;  private System.Windows.Forms.Button randowButton;  private System.Windows.Forms.Button minusSizeButton;  private System.Windows.Forms.Button plusSizeButton;  private System.Windows.Forms.GroupBox groupBox2;  private System.Windows.Forms.Button task9Button;  private System.Windows.Forms.TextBox textBox1;  private System.Windows.Forms.Label label3;  private System.Windows.Forms.TextBox completeTextBox;  private System.Windows.Forms.Button task13Button;  private System.Windows.Forms.Label label4;  }  } |

Контрольные вопросы:  
1. Что такое массивы?

Массив - это последовательность элементов одного типа данных, хранящихся под одним именем и занимающих непрерывное место в памяти компьютера [1].

Ключевые моменты:

* Массивы представляют собой контейнер для хранения коллекции элементов
* Все элементы массива имеют одинаковую типизацию
* Индексация массива начинается с нуля

2. Особенности работы с массивами в C#.

В C# массивы являются частью языка программирования и предоставляют удобный способ работы с наборами данных [2].

Особенности:

* Динамическое выделение памяти при создании массива
* Возможность изменения размера массива
* Поддержка многомерных массивов
* Гибкие типы (dynamic) для работы с неявно типизированными массивами

3. Компоненты работы с массивами в Windows Form

В Windows Forms есть несколько компонентов для работы с массивами:

* TextBox: для ввода и отображения строковых данных
* ListBox: для отображения списка элементов
* ComboBox: для выбора из предопределенного списка
* DataGridView: для отображения и редактирования таблицы данных

4. Методы заполнения двумерных массивов в Windows Form

Для заполнения двумерного массива в Windows Forms можно использовать следующие методы:

* Заполнение через цикл:

int[,] array = new int[3, 3];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

array[i, j] = i \* j;

}

}

* Заполнение через LINQ:

int[,] array = Enumerable.Range(0, 9).Select(i => new { X = i % 3, Y = i / 3 })

.SelectMany(x => new[] { x.X, x.Y },

(x, y) => new { X = x, Y = y })

.Select(x => x.X \* x.Y)

.ToArray();

5. Опишите принципы работы ListBox в Windows Form

ListBox - это элемент управления, который позволяет пользователю выбрать один или несколько элементов из списка [3].

Принципы работы:

* Отображение списка элементов
* Возможность выделения одного или нескольких элементов
* Поддержка прокрутки списка
* Возможность добавления и удаления элементов динамически
* Связь с массивом через SelectedItems свойство

6. Опишите принципы работы CheckedListBox в Windows Form

CheckedListBox - это расширенная версия ListBox, которая позволяет отмечать элементы как выбранные или невыбранные [4].

Принципы работы:

* Отображение списка элементов с галочками
* Возможность выбора и отметки нескольких элементов
* Поддержка прокрутки списка
* Возможность добавления и удаления элементов динамически
* Связь с массивом через SelectedIndices и CheckedItems свойства

7. Опишите принципы работы ComboBox в Windows Form

ComboBox - это элемент управления, который представляет собой выпадающее меню со списком элементов [5].

Принципы работы:

* Отображение списка элементов в выпадающем меню
* Возможность выбора одного элемента из списка
* Поддержка ввода текста для поиска элементов
* Возможность добавления и удаления элементов динамически
* Связь с массивом через SelectedIndex и Items свойства

1. Опишите принципы работы DataGridView в Windows Form

DataGridView - это мощный компонент для отображения и редактирования таблицы данных [6].

Принципы работы:

Отображение таблицы данных в виде ячеистой структуры

Возможность сортировки и фильтрации данных

Поддержка редактирования ячеек

Возможность добавления и удаления строк и столбцов

Связь с массивами или коллекциями через DataSource свойство

**ЛАБОРАТОРНАЯ РАБОТА №4**

**Основы работы с графикой в C# с использованием Windows Forms. Класс Graphics.**

**Цель**: изучить методы создания приложений с использованием Windows Form. Также изучить стандартные компоненты и научиться их применять на практике.

**Ход работы**

Задание 1. На главной форме построить график функции, соответствующие варианту (9) в трех вариациях, каждый график должен отображаться другим цветом и изображаться на одной координатной сетке.

Для этого всего я создал отдельные три класса, которые содержат в себе модели. Пример одного из них представлен на рисунке 4.1

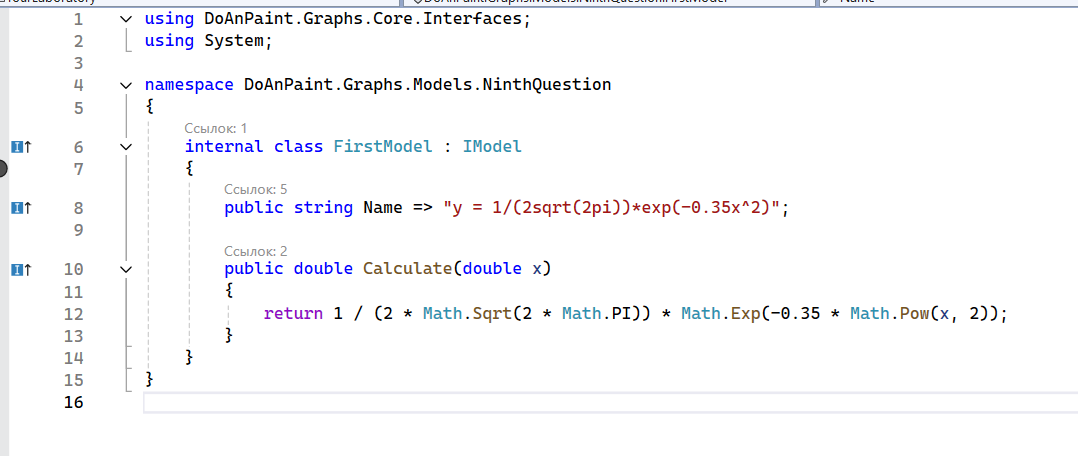


Рисунок 4.1 – Пример кода первой модели

На рисунке 4.2 представлен пример окна с выводом графиков. Каждый имеет свой цвет.

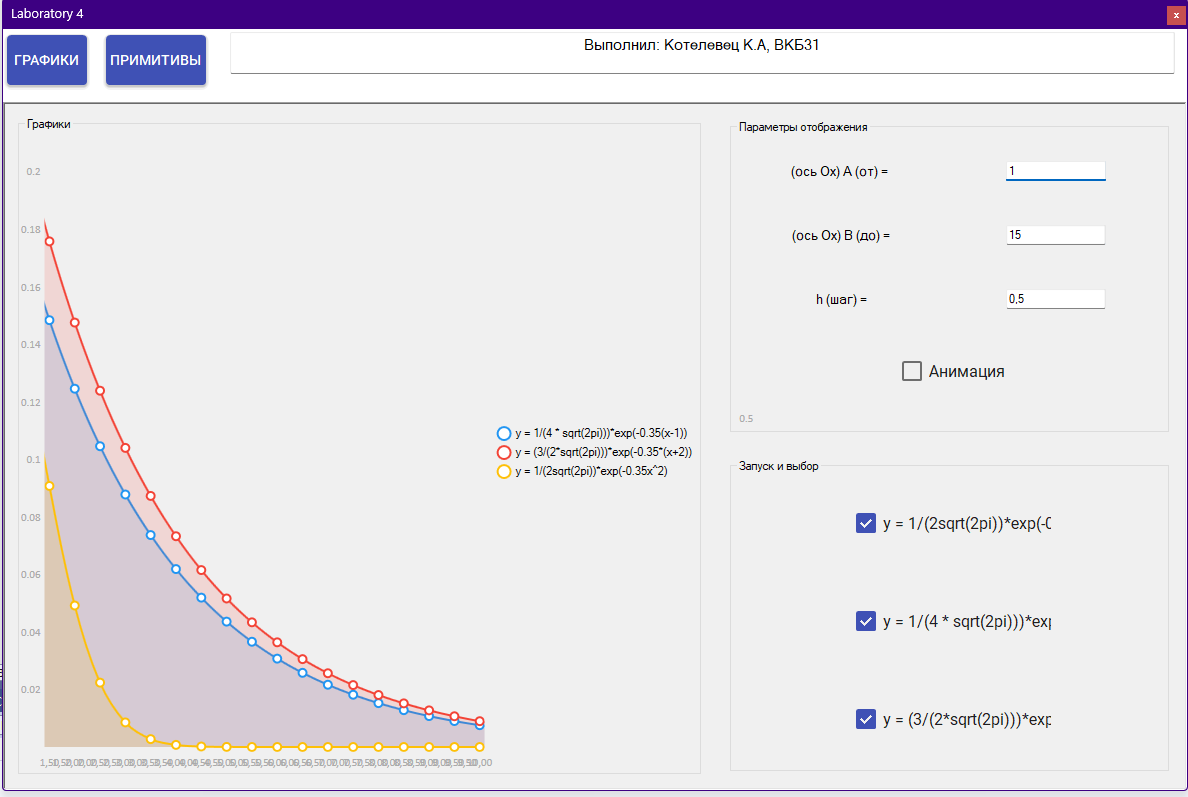


Рисунок 4.2 – Графики

Задание 2. Создать меню на главной форме для вызова дополнительной формы, на которой создать анимацию для рисования графика функции по варианту.

Для этого на той же форме (рисунок 4.2) я сделал CheckBox для анимации.

Анимация проходит просто плавающим каскадом снизу вверх – анимация по умолчанию для компонента cartesianChart.

Задание 3. В созданном меню создать обработчик вызова еще одной формы, где нарисовать на форме минимум 5 примитивов, разного размера, цвета и заливки. Создать кнопки загрузки и сохранения изображения.

На рисунке 4.3 представлен пример данного окна.

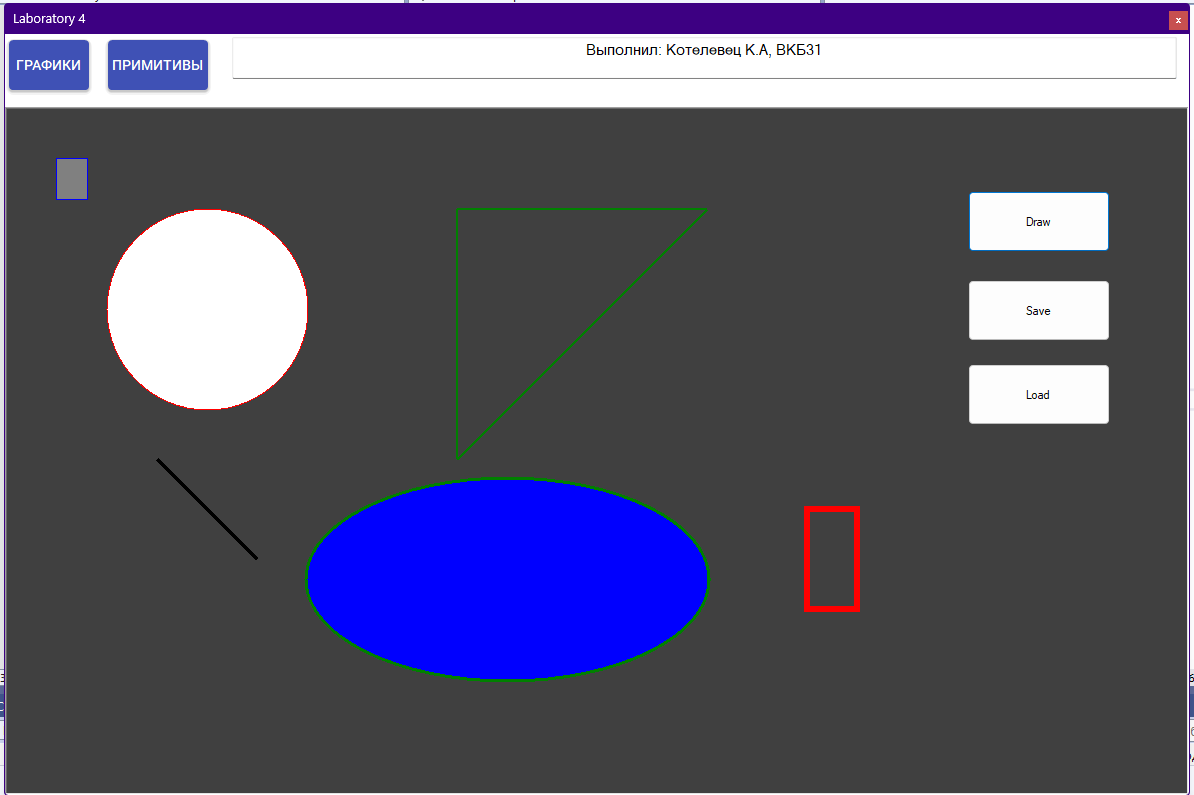


Рисунок 4.3 – Примитивы

На рисунке 4.4 показана возможность сохранения данного изображения и на рисунке 4.5 результат

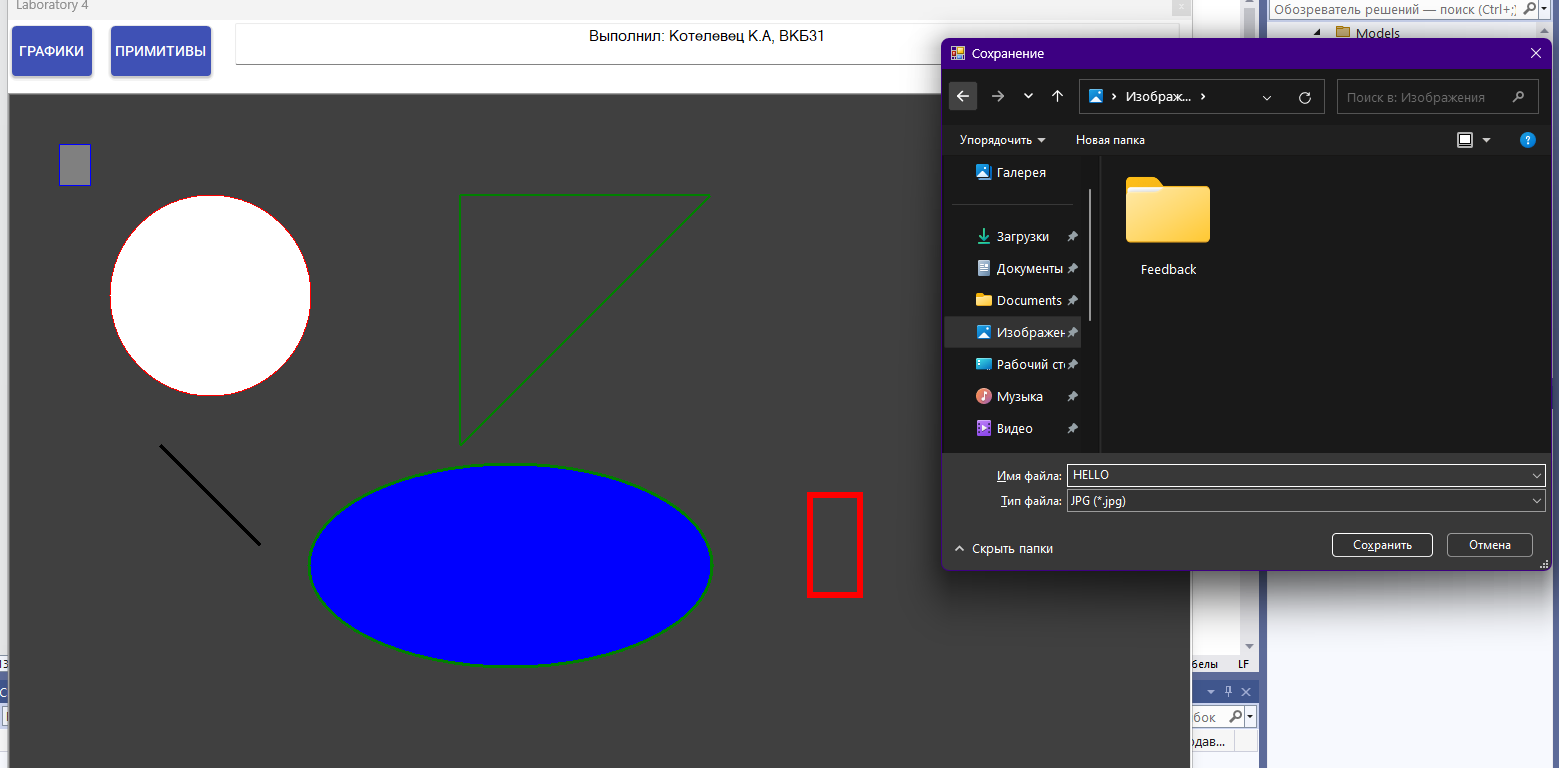


Рисунок 4.4 – Сохранение файла

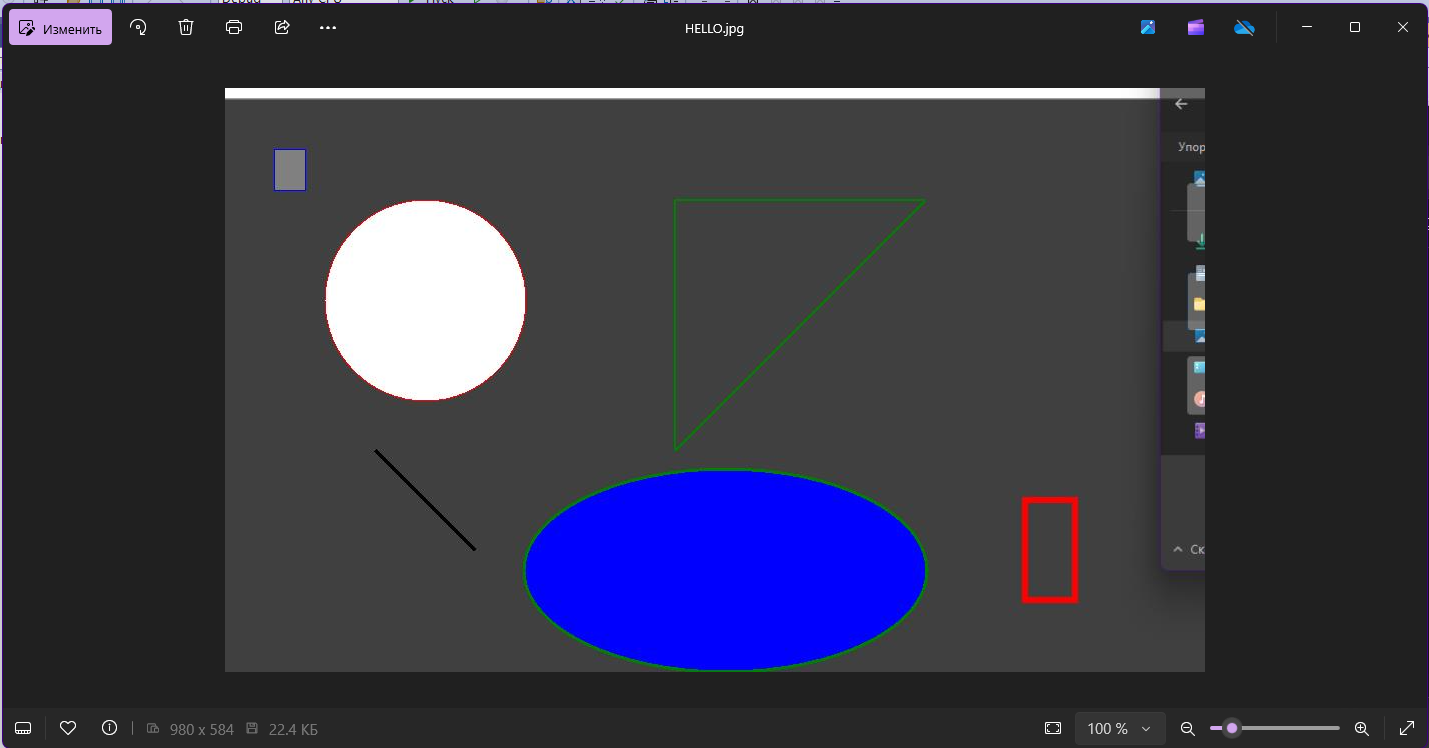


Рисунок 4.5 – Результат

Ниже представлен листинг программы для моделей и пары основных классов

MainForm.cs

|  |
| --- |
| using DoAnPaint.Graphs.Core;  using DoAnPaint.Graphs.Core.Helpers;  using DoAnPaint.Graphs.Core.Interfaces;  using DoAnPaint.Graphs.Presenters;  using System;  using System.Drawing;  using System.Linq;  using System.Windows.Forms;  namespace DoAnPaint  {  public partial class MainForm : Form, IMainView  {  private readonly IMainPresenter \_presenter;  public MainForm()  {  \_presenter = new MainPresenter(this, new FormFactory(this));  InitializeComponent();  this.SetBevel(false);  Controls.OfType<MdiClient>().FirstOrDefault().BackColor = Color.FromArgb(232, 234, 237);  }  public void QuestionButton\_Click(object sender, EventArgs e)  {  \_presenter.OnButtonClick(sender, e);  }  /// <summary>  /// После нажатия определенной кнопки задания открывается View, который был привязан.  /// Здесь идут визуальные изменения, поэтому делается на стороне View, а не Presenter, как я понимаю  /// </summary>  /// <param name="form">класс формы, который хотим показывать пользователю</param>  public void ShowForm(Form form)  {  form.Show();  }  /// <summary>  /// Если форма была загружена в память, то открывается заново (мое предположение, это взял из гайда в README.md).  /// Опять-таки визуальное изменение, но тут на стороне View это, что тоже смущает...  /// </summary>  /// <param name="form">класс формы, который хотим показывать пользователю</param>  public void ActivateForm(Form form)  {  form.Activate();  }  /// <summary>  /// MDI приложение знает свои дочерние формы, которые я использую.  /// Здесь можно получить каждую форму чисто по имени, опять-таки идет поиск тегов с кнопки.  /// </summary>  /// <param name="formName">имя формы, который мы получили с тэга кнопки</param>  /// <returns>возвращает форму, если она была созана до этого, в ином случае null</returns>  public Form GetOpenFormByName(string formName)  {  return MdiChildren.FirstOrDefault(f => f.GetType().Name == formName);  }  private void flowLayoutPanel1\_Paint(object sender, PaintEventArgs e)  {  }  private void textBox1\_TextChanged(object sender, EventArgs e)  {  }  private void MainForm\_Load(object sender, EventArgs e)  {  }  }  } |

FirstModel.cs

|  |
| --- |
| using DoAnPaint.Graphs.Core.Interfaces;  using System;  namespace DoAnPaint.Graphs.Models.NinthQuestion  {  internal class FirstModel : IModel  {  public string Name => "y = 1/(2sqrt(2pi))\*exp(-0.35x^2)";  public double Calculate(double x)  {  return 1 / (2 \* Math.Sqrt(2 \* Math.PI)) \* Math.Exp(-0.35 \* Math.Pow(x, 2));  }  }  } |

SecondModel

|  |
| --- |
| using DoAnPaint.Graphs.Core.Interfaces;  using System;  namespace DoAnPaint.Graphs.Models.NinthQuestion  {  internal class SecondModel : IModel  {  public string Name => "y = 1/(4 \* sqrt(2pi)))\*exp(-0.35(x-1))";  public double Calculate(double x)  {  return 1 / (4 \* Math.Sqrt(2)) \* Math.Exp(-0.35 \* (x - 1));  }  }  } |

ThirdModel

|  |
| --- |
| using DoAnPaint.Graphs.Core.Interfaces;  using System;  namespace DoAnPaint.Graphs.Models.NinthQuestion  {  internal class ThirdModel : IModel  {  public string Name => "y = (3/(2\*sqrt(2pi)))\*exp(-0.35\*(x+2))";  public double Calculate(double x)  {  return 3 / (2 \* Math.Sqrt(2 \* Math.PI)) \* Math.Exp(-0.35 \* (x + 2));  }  }  } |

BaseForm

|  |
| --- |
| using DoAnPaint.Graphs.Core.Interfaces;  using DoAnPaint.Graphs.Presenters;  using LiveCharts;  using LiveCharts.WinForms;  using System;  using System.Collections.Generic;  using System.Windows.Forms;  namespace DoAnPaint.Graphs.Core.Abstract  {  public partial class BaseForm : Form, IView  {  public IPresenter \_presenter;  public List<IModel> \_models;  public double Step { get; set; }  public double Start { get; set; }  public double End { get; set; }  public bool IsAnimationEnabled => animationCheckBox.Checked;  public CartesianChart CartesianChart => cartesianChart;  public SeriesCollection ChartData  {  get => cartesianChart.Series;  set  {  cartesianChart.Series.Clear();  foreach (var series in value)  {  cartesianChart.Series.Add(series);  }  }  }  /// <summary>  /// В Visual Studio под капотом для отображения Form используется рефлексия, поэтому этот пустой конструктор нужен  /// !!!НЕ ИСПОЛЬЗОВАТЬ ЕГО!!!!  /// </summary>  public BaseForm()  {  InitializeComponent();  }  /// <summary>  /// Пытался изначально написать BaseForm конструктор с List<IModel>, чтобы потом у дочерних классов вызывать родительский конструктор, передавая список.  /// Но проблема была в том, что не отображался экран. Было вот решено запихнуть в такой отдельный метод.  /// Каждый дочерний класс должен вызвать этот метод, передавая ему список моделей, в ином случае все поломается.  /// К сожалению, его нельзя сделать абстрактным, так как класс должен иметь модификатор abstract, который в свою очередь не может быть вместе с partitial  /// </summary>  /// <param name="models"></param>  public void InitializeModels(List<IModel> models)  {  \_models = models;  \_presenter = new BasePresenter(this, \_models);  firstChartCheckBox.Text = \_models[0].Name;  secondChartCheckBox.Text = \_models[1].Name;  thirdChartCheckBox.Text = \_models[2].Name;  }  /// <summary>  /// Обработчик событий на вписывание значений в textBox, где написано "(ось Ox) А (от) ="  /// </summary>  protected void GraphStartTextBox\_Validating(object sender, System.ComponentModel.CancelEventArgs e)  {  ValidateGraphInput(graphStartTextBox, value => Start = value, "Введите корректное значение для начала графика.");  }  /// <summary>  /// Обработчик событий на вписывание значений в textBox, где написано "(ось Ox) B (до) ="  /// </summary>  protected void GraphEndTextBox\_Validating(object sender, System.ComponentModel.CancelEventArgs e)  {  ValidateGraphInput(graphEndTextBox, value => End = value, "Введите корректное значение для конца графика.");  }  /// <summary>  /// Обработчик событий на вписывание значений в textBox, где написано "h (шаг) = "  /// </summary>  protected void GraphStepTextBox\_Validating(object sender, System.ComponentModel.CancelEventArgs e)  {  ValidateGraphInput(graphStepTextBox, value => Step = value, "Введите корректное значение для шага графика.");  }  /// <summary>  /// Обработчик событий на выбор у 1 comboBox  /// </summary>  protected void FirstChartCheckBox\_CheckedChanged(object sender, EventArgs e)  {  ChartCheckBox\_CheckedChanged(firstChartCheckBox, 0);  }  /// <summary>  /// Обработчик событий на выбор у 2 comboBox  /// </summary>  protected void SecondChartCheckBox\_CheckedChanged(object sender, EventArgs e)  {  ChartCheckBox\_CheckedChanged(secondChartCheckBox, 1);  }  /// <summary>  /// Обработчик событий на выбор у 3 comboBox  /// Честно говоря, понимаю, что тут можно было пойти через рефлексию и вписывание функции в тегах, но у меня времени нет, чтобы это писать.  /// Изляшняя масштабируемость - тоже плохо, когда мало времени.  /// </summary>  protected void ThirdChartCheckBox\_CheckedChanged(object sender, EventArgs e)  {  ChartCheckBox\_CheckedChanged(thirdChartCheckBox, 2);  }  /// <summary>  /// Вспомогательный метод, который фиксиурет изменения в ComboBox и добавляет в Presenter необходимые функции для отображения и построения  /// </summary>  /// <param name="checkBox">checkBox, где пользователь выбрал галочку</param>  /// <param name="modelIndex">Индекс, под которым находится нужный график</param>  private void ChartCheckBox\_CheckedChanged(CheckBox checkBox, int modelIndex)  {  if (  string.IsNullOrWhiteSpace(graphStartTextBox.Text) ||  string.IsNullOrWhiteSpace(graphEndTextBox.Text) ||  string.IsNullOrWhiteSpace(graphStepTextBox.Text)  )  {  MessageBox.Show("Заполните поля перед тем, как выбирать функцию");  checkBox.Checked = false;  }  else  {  if (checkBox.Checked)  {  \_presenter.SelectModel(\_models[modelIndex]);  }  else  {  \_presenter.DeselectModel(\_models[modelIndex]);  }  \_presenter.Draw();  }  }  /// <summary>  /// Валидация ввода от пользователя  /// </summary>  /// <param name="textBox">textBox, в котором пользователь ввел значение</param>  /// <param name="setValueAction">что происходит, когда пользователь ввел значение, здесь я сделал лямба выражение такое</param>  /// <param name="errorMessage">ошибка, которая выводится в MessageBox</param>  private void ValidateGraphInput(TextBox textBox, Action<double> setValueAction, string errorMessage)  {  if (double.TryParse(textBox.Text, out double value))  {  setValueAction(value);  if (  !string.IsNullOrWhiteSpace(graphStartTextBox.Text) &&  !string.IsNullOrWhiteSpace(graphEndTextBox.Text) &&  !string.IsNullOrWhiteSpace(graphStepTextBox.Text)  )  {  \_presenter.Draw();  }  }  else  {  MessageBox.Show(errorMessage);  textBox.Clear();  }  }  private void graphStartTextBox\_TextChanged(object sender, EventArgs e)  {  }  private void animationCheckBox\_CheckedChanged(object sender, EventArgs e)  {  }  private void cartesianChart\_ChildChanged(object sender, System.Windows.Forms.Integration.ChildChangedEventArgs e)  {  }  private void panel2\_Paint(object sender, PaintEventArgs e)  {  }  }  } |

Все остальные коды слишком большие для отчёта

Контрольные вопросы:

1. Опишите класс Graphics.  
Класс Graphics - это базовый класс для работы с графическими объектами в Windows Forms и WPF. Он предоставляет методы для рисования, отображения текста, работы с изображениями и другими графическими операциями.

2. Свойства класса Graphics.  
Основные свойства класса Graphics включают:

* Width - ширина графика
* Height - высота графика
* SmoothingMode - режим размытия линий
* InterpolationMode - режим интерполяции
* PixelOffsetMode - смещение пикселей
* CompositingQuality - качество композиции

3. Методы класса Graphics.  
Основные методы класса Graphics включают:

* DrawLine() - рисование линии
* DrawRectangle() - рисование прямоугольника
* DrawString() - отображение текста
* FillRectangle() - заливка прямоугольника
* DrawImage() - отображение изображения
* Clear() - очистка графика

4. Создание простой анимации.

Для создания простой анимации можно использовать следующие шаги:

1. Создать объект Graphics
2. Использовать методы DrawLine() или DrawRectangle() для рисования фигур
3. Изменять координаты и параметры рисования в цикле
4. Обновлять содержимое графика через timer или таймер задержки

|  |
| --- |
| private void timer1\_Tick(object sender, EventArgs e)  {  Graphics g = this.CreateGraphics();  int x = 50;  int y = 50;  int dx = 2;  int dy = 2;  x += dx;  y += dy;  if (x > ClientSize.Width || x < 0)  dx \*= -1;  if (y > ClientSize.Height || y < 0)  dy \*= -1;  g.DrawEllipse(new Pen(Color.Red), x, y, 20, 20);  g.Dispose();  } |

5. Создание графика функций.  
Для создания графика функций можно:

1. Определить функцию
2. Создать массив точек
3. Преобразовать точки в координаты экрана
4. Отобразить точки на графике

Пример кода:

|  |
| --- |
| private void drawGraph()  {  Graphics g = this.CreateGraphics();  int width = ClientSize.Width;  int height = ClientSize.Height;  float min = float.MaxValue;  float max = float.MinValue;  for (float x = -10; x <= 10; x += 0.1f)  {  float y = Math.Sin(x);  min = Math.Min(min, y);  max = Math.Max(max, y);  }  float scaleX = width / (max - min);  float scaleY = height / (max - min);  for (float x = -10; x <= 10; x += 0.1f)  {  float y = Math.Sin(x);  int xScreen = (int)((x + 10) \* scaleX);  int yScreen = height - (int)(y \* scaleY);  g.DrawPoint(new Point(xScreen, yScreen), Color.Red);  }  g.Dispose();  } |

6. Типы отображения линий.

Класс Graphics предоставляет несколько типов отображения линий через свойство Pen:

* SolidLine - непрерывная линия
* Dash - пунктирная линия
* DashDot - чередующиеся точки и линии
* DashDotDot - чередующиеся точки и короткие линии
* CustomDashCap - пользовательский вид линии

7. Для чего используется библиотека System.Threading

Библиотека System.Threading используется для работы с потоками выполнения в многопоточной среде. Основные классы включают:

* Thread - основной класс для создания новых потоков
* ThreadPool - пул потоков для выполнения задач
* Semaphore - механизм синхронизации доступа к ресурсам
* Mutex - семафор для синхронизации между процессами

8. Опишите принципы работы pictureBox в Windows Form  
pictureBox - это графический элемент управления в Windows Forms, который позволяет отображать изображения или рисовать на них. Основные принципы работы:

1. Добавление pictureBox на форму
2. Установка свойства Image для загрузки изображения
3. Рисование на pictureBox через метод CreateGraphics()
4. Очистка графика после рисования с помощью Dispose()

9. Опишите принципы работы saveFileDialog1 в Windows Form

saveFileDialog1 - это диалоговое окно для выбора места сохранения файла. Основные принципы работы:

1. Создание экземпляра SaveFileDialog
2. Настройка фильтра файлов через свойство Filter
3. Открытие диалога с помощью метода ShowDialog()
4. Получение выбранного пути через свойство FileName при успешном закрытии диалога

10. Опишите принципы работы openFileDialog1в Windows Form

openFileDialog1 - это диалоговое окно для выбора файла для открытия. Основные принципы работы:

1. Создание экземпляра OpenFileDialog
2. Настройка фильтра файлов через свойство Filter
3. Открытие диалога с помощью метода ShowDialog()
4. Проверка результата через свойство DialogResult
5. Получение пути к открытому файлу через свойство FileName

**ЛАБОРАТОРНАЯ РАБОТА №5**

**Постановка задачи на проектирование информационной системы** (**ИС).**

**Анализ программных продуктов.**

**Задание 1.** Выбрать предметную область, для которой надо спроектировать и реализовать ИС. Если для выбранной предметной области нет описания, составить описание. Если необходимо, откорректировать существующее описание, если оно есть для выбранной темы.

У меня вариант 9, поэтому у меня следующая предметная область:

**Предметная область № 9: ресторан.**

Ресторан предоставляет для своих клиентов услугу питания. На каждый день составляется меню, которое включает в себя список блюд для питания. На основе этого меню составляется список для закупки необходимых продуктов питания, входящих в состав блюд. Клиент, приехав в ресторан, выбирает из меню блюда, которые он хотел бы заказать, их готовят, если они заранее не были готовы, и приносят клиенту. В качестве дополнительной услуги ресторан может организовать развлекательные программы в своем помещении. По результатам своей деятельности ресторан производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.

**Информационная система** (**ИС**) — система обработки информации и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию (ISO/IEC 2382-1:1993).

**Задание 2.** Осуществить сравнительный анализ **программных продуктов** - информационных систем (ИС), уже существующих, работающих в выбранной Вами предметной области и выполняющих информационные функции, похожие на информационные функции проектируемой Вами ИС. Анализ представить в текстовой форме - описание характеристик и особенностей работы. Это может быть информация из Интернета с указанием источников, из которых взята информация. Добавить скриншоты с обзорных сайтов, чтобы продемонстрировать внешний вид и интерфейс анализируемых ИС.

На рынке общественного питания высокая конкуренция. Поэтому качество обслуживания играет основополагающую роль в ресторанном бизнесе. Профессиональное программное обеспечение упрощает работу персонала, работники заведения смогут уделить больше внимания гостю. Автоматизация позволит сэкономить время и деньги, а работа станет более эффективной.

Основные преимущества автоматизации в ресторанном бизнесе:

* Повышение эффективности работы персонала
* Увеличение времени, которое можно уделять клиентам
* Экономия средств на различных операционных расходах
* Быстрое обучение новых сотрудников
* Возможность контролировать оплату, меню, склад и маркетинг через единую систему

Ключевые моменты:

* Автоматизация позволяет предоставлять лучший сервис клиентам
* Снижение ошибок при оформлении заказов
* Улучшение безопасности рабочего места
* Повышение конкурентоспособности ресторана
* Возможность адаптироваться к изменениям в бизнесе

Преимущества для клиентов:

* Быстрая доставка блюд
* Доступ к онлайн-заказам и контактным платежам
* Возможность бронирования столиков онлайн
* Инновационный диннерский опыт

В таблице 1 представлен сравнительный анализ нескольких таких систем

Таблица 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факторы | iiko | Restik | YUMA | Quick resto |
| Функциональность | * Comprehensive POS-система * Менеджмент блюд с возможностью создания кастомных меню * Онлайн-бронирование столиков * Аналитические инструменты для отслеживания продаж и финансов * Управление запасами и поставщиками * Интеграция с системами доставки еды | * Интегрированная система POS и менеджмента блюд * Онлайн-бронирование столиков * Мобильное приложение для клиентов * Аналитические инструменты для оптимизации продаж * Управление персоналом и расписанием * Интеграция с платежными системами | * Comprehensive POS-система * Менеджмент блюд с возможностью создания кастомных меню * Онлайн-бронирование столиков * Аналитические инструменты для отслеживания бизнеса * Управление запасами и поставщиками * Интеграция с системами доставки еды | * Интегрированная система POS и менеджмента блюд * Онлайн-бронирование столиков * Мобильное приложение для клиентов * Аналитические инструменты для оптимизации бизнеса * Управление персоналом и расписанием * Интеграция с платежными системами |
| Интерфейс | * Профессиональный интерфейс с возможностью настройки * Интуитивно понятные панели управления * Возможность создания кастомных отчетов и графиков | * Современный и интуитивно понятный интерфейс * Простое управление заказами и оплатой * Легкое бронирование через мобильное приложение | * Интуитивно понятный интерфейс для персонала * Простое управление меню и ценниками * Легкое создание кастомных отчетов | * Современный и интуитивно понятный интерфейс * Простое управление заказами и оплатой * Легкое бронирование через мобильное приложение |
| Интеграции | * Интеграция с платежными системами * Интеграция с системами доставки еды * API для интеграции с другими системами * Поддержка различных POS-систем | * Интеграция с платежными системами * Интеграция с системами доставки еды * API для интеграции с другими системами * Поддержка различных POS-систем | * Интеграция с платежными системами * Интеграция с системами доставки еды * API для интеграции с другими системами * Поддержка различных POS-систем | * Интеграция с платежными системами * Интеграция с системами доставки еды * API для интеграции с другими системами * Поддержка различных POS-систем |

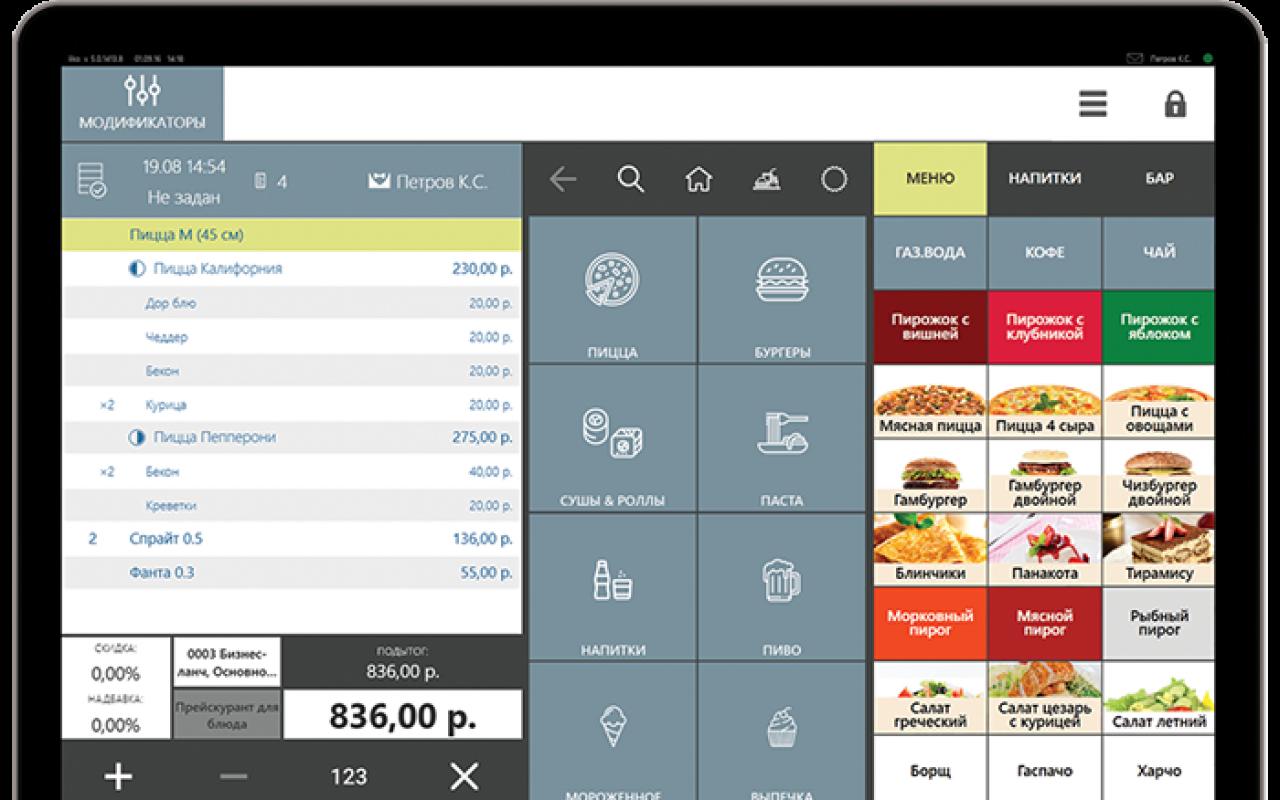


Рисунок 5.1 – Интерфейс Iiko

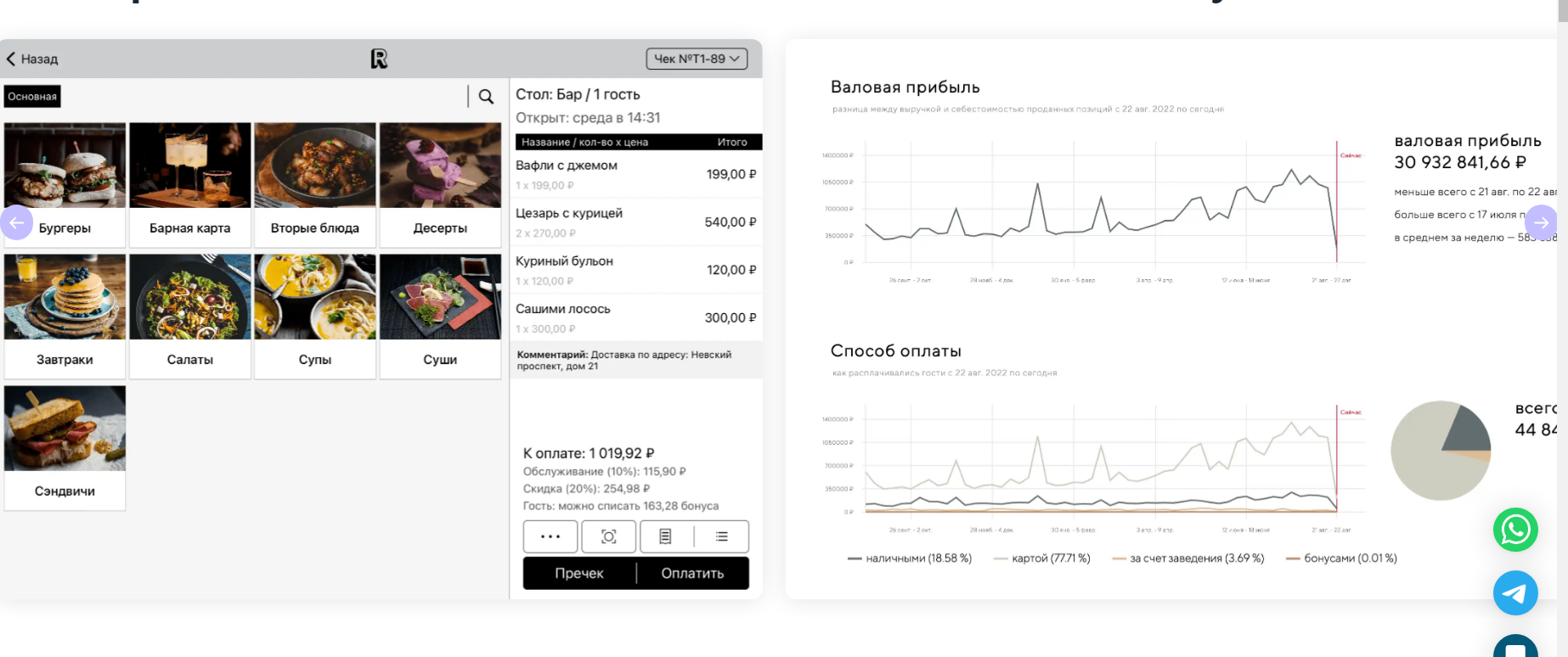


Рисунок 5.2 – Интерфейс Ristik

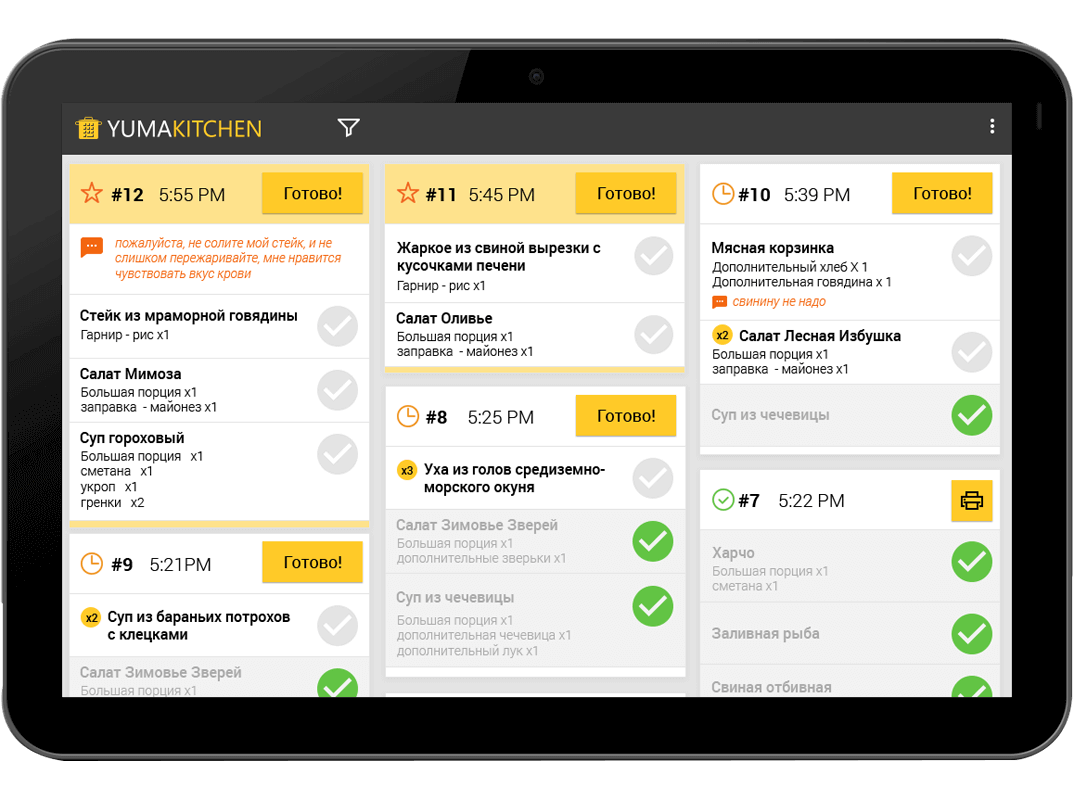


Рисунок 5.3 – YUMA

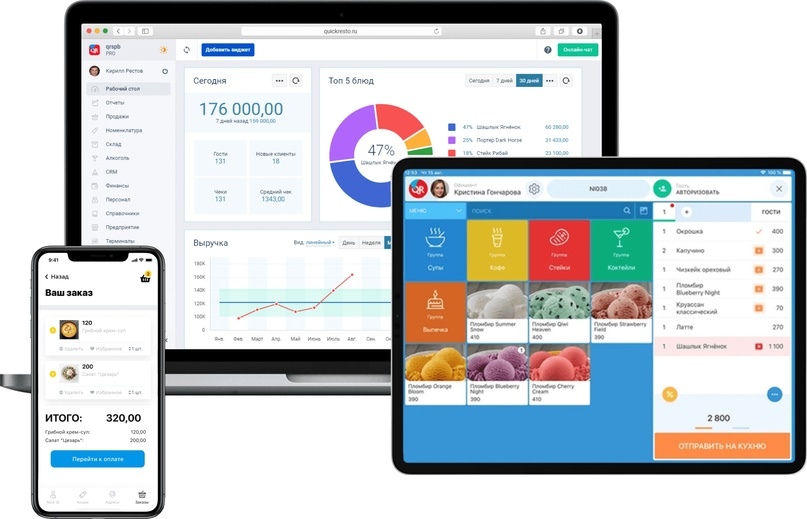


Рисунок 5.4 - Quick resto

**Задание 3.** На основе анализа существующих программных продуктов (примеров проектируемой ИС) сделать выводы о программном продукте, который Вы будете проектировать: сформулировать назначение и цели проектируемой ИС.

Всё это представлено в таблице 2.

Таблица 2

|  |  |
| --- | --- |
| Характеристики проекта | Значения характеристик |
| Назначение системы | |
| вид деятельности системы (управление, проектирование и т. п.) | * Mu Управление рестораном * Mu Проектирование меню и ценовой политики * Mu Анализ продаж и оптимизация бизнеса * Co Управление персоналом и расписанием * W Управление запасами и логистикой * Co Управление бронированием столиков * S Управление онлайн-заказами * Co Создание отчетов и аналитики |
| перечень объектов информатизации (объектов), на которых предполагается ее использовать | * Mu Кассовые аппараты * Mu Терминалы для обработки заказов * Mu Компьютеры для административных задач * Co Мобильные устройства для персонала * Co Мониторы для отображения информации о бронировании * S Планшеты для создания меню и управления ценами * S Серверы для хранения данных и обработки транзакций |
| Цели создания системы | |
| наименования и требуемые значения технических показателей объекта информатизации, которые должны быть достигнуты в результате создания ИС | - Mu Кассовый терминал, чековый аппарат |
| наименования и требуемые значения технологических показателей объекта информатизации, которые должны быть достигнуты в результате создания ИС | * Mu Установка необходимого ПО, создание информационной системы и базы данных |
| наименования и требуемые значения производственно-экономических показателей объекта информатизации, которые должны быть достигнуты в результате создания ИС | * Mu Снижение времени на оформление заказа * Mu Увеличение точности подсчета и оплаты благодаря интеграции с кассовыми аппаратами * Mu Снижение времени бронирования столиков благодаря интеграции с онлайн-бронированием * Mu Увеличение среднего чека благодаря персонализированным рекомендациям * Mu Снижение затрат на персонал благодаря оптимизации расписания и управления рабочим временем * Mu Повышение лояльности клиентов благодаря персонализированной коммуникации и лояльности программе |
| критерии оценки достижения целей создания системы | * Информационная система с необходимым функционалом, база данных |

**Назначение и цели создания системы**

**Назначение системы**

Информационная система управления рестораном предназначена для автоматизации функций управления, сбора, обработки и передачи информации на всех уровнях ресторанного бизнеса:

* Уровень ресторана (основной ресторан)
* Уровень филиалов (если применимо)
* Уровень головного офиса (централизованное управление)

**Цели создания системы**

ИС управления рестораном создается для достижения следующих целей:

* Повышение эффективности работы ресторана и оптимизация бизнес-процессов
* Увеличение скорости обработки заказов и бронирований
* Улучшение качества обслуживания клиентов
* Повышение лояльности клиентов и увеличение среднего чека
* Оптимизация управленческих решений благодаря аналитическим данным
* Снижение затрат на персонал и оптимизация расписания
* Автоматизация ручных операций по формированию и выдаче документов
* Создание единого центра управления всеми аспектами ресторанного бизнеса
* Экономия ресурсов (трудовые, материальные, временные) через оптимизацию процессов
* Формирование банков данных о клиентах, продуктах и внутренних процессах
* Увеличение скорости принятия решений благодаря доступу к актуальной информации
* Улучшение репутации ресторана через систему отзывов и рейтингов

**ЛАБОРАТОРНАЯ РАБОТА №6**

**Обследование объекта автоматизации**

**Задание 1.** Составить примерную организационную схему объекта, на котором планируется создание ИС (Рисунок 6.1)

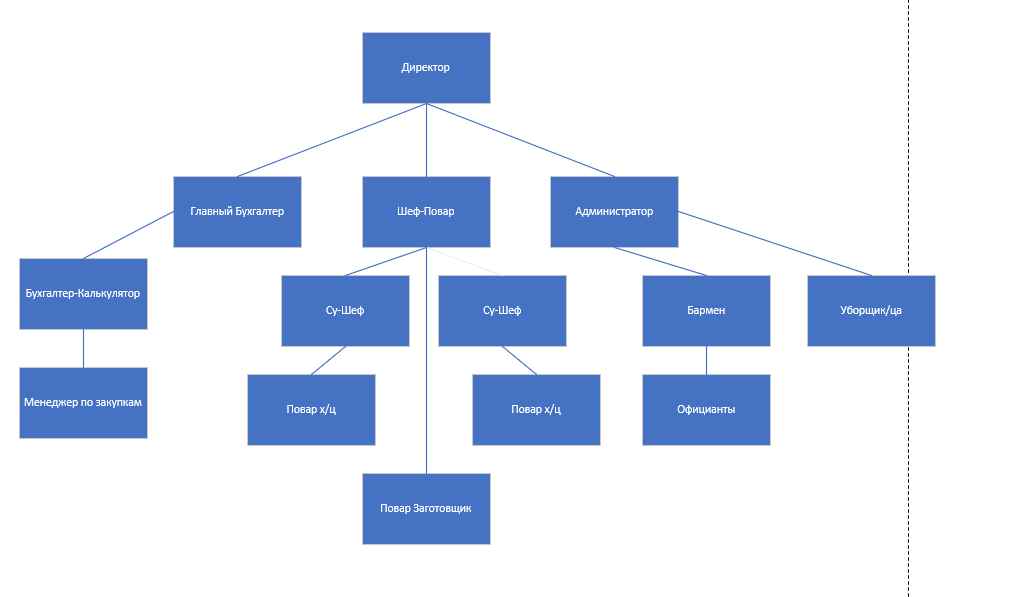
****

Рисунок 6.1 – Организационная схема объекта автоматизации

**Обследование объекта автоматизации** представляет собой комплекс научно-исследовательских работ и организационно-технических мероприятий, направленных на изучение и диагностический анализ его организационной структуры, деятельности и существующих технологий обработки информации.

**Задание 2.** Исходя из выбранной предметной области и организационной схемы объекта, выполнить следующие проектные решения, заполнив соответствующие таблицы и оформив их в Word. Они представлены в таблицах 6.1-6.4

Таблица 6.1 - Методика проведения обследования

|  |  |  |
| --- | --- | --- |
| **Параметры** | **Решение** | **Обоснование** |
| По цели обследования | Системное | Разработка проекта решения комплекса задач |
| По количеству участников | Индивидуальное | Один проектировщик |
| По степени охвата предметной области | Сплошное |  |

Таблица 6.2 - Методы сбора материалов обследования

|  |  |  |
| --- | --- | --- |
| Характеристика выбранного метода | Методы сбора, выполняемого силами проектировщиков | Методы сбора, выполняемые силами специалистов предметной области |
| Проведение бесед по вопросам, которые носят глобальный характер и относятся к определению стратегии развития информационной системы | Проведение бесед, опросов и семинаров с руководителями объекта, и его подразделений | Ведение индивидуальной тетради-дневника |
| Применяется для определения трудоемкости и стоимости работ, подлежащих автоматизации, а также для установления объемов работ по отдельным операциям | Расчетный метод | Метод документальной инвентаризации управленческих работ |
| Заключается в декомпозиции рассматриваемого делового процесса или работы на ее составные части: задачи, расчеты, операции и элементы операций. Проводится анализ каждой части в отдельности, выявляется повторяемость отдельных операций, многократное обращение к одной и той же операции, степень их зависимости друг от друга | Метод анализа операций | Метод документальной инвентаризации управленческих работ |
| Заключается в извлечении необходимой информации из предоставленных материалов: чаще всего бумажной документации, а также электронных таблиц и иных файлов | Метод анализа предоставленного материала | Метод документальной инвентаризации управленческих работ |
| Сравнение и анализ информации | Интернет-опрос | Интернет-опрос |
| Исследование предметной области | Проведение бесед, опросов и семинаров. Интервью и анкетирование исполнителей | Ведение индивидуальной тетради-дневника Самофотографии рабочего дня |
| Анализ бизнес-процессов | Метод анализа операций Расчетный метод | Метод документальной инвентаризации управленческих работ |
| Формализация потоков данных | Метод выборочного хронометража отдельных работ Личное наблюдение | Самофотографии рабочего дня |
| Поиск аналогий | Метод аналогии | Анализ предоставленного материала |

Таблица 6.3 - Программа обследования

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование вопроса | Источники информации | Получатель информации |
| 1 | Цель функционирования объекта | Руководитель предприятия | Руководитель проекта |
| 2 | Основные параметры объекта | Методы сбора | Руководитель проекта |
| 3 | Функции подразделения | Метод анализа операций | Руководители отделов |
| 4 | Технологические процессы | Метод анализа предоставленного материала | Руководитель проекта |
| 5 | Материальных потоки и процессы их обработки | Расчетный метод | Руководитель предприятия |

Таблица 6.4 - План-график работ по проведению обследования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование работы | Исполнитель | Дата начала | Длительность | Дата окончания |
| 1 | Проведение беседы с руководителем предприятия | Руководитель проекта | 3.09.2024 | 2 часа | 3.09.2024 |
| 2 | Проведение опроса и семинара с руководителями отделов | Руководитель проекта | 4.09.2024 | 4.5 часа | 4.09.2024 |
| 3 | Анализ кассовых операций | Руководитель проекта | 30.11.2024 | 1 день | 1.12.2024 |
| 4 | Расчет стоимости ПО | Руководитель проекта | 11.09.2024 | 2 дня | 13.09.2024 |
| 5 | Анализ предоставленного материала от руководителя | Руководитель проекта | 6.09.2024 | 3 дня | 9.09.2024 |
| 6 | Изучение работы других ИС | Руководитель проекта | 1.12.2024 | 3 дня | 5.12.2024 |
| 7 | Ведение дневника официанта | Руководитель проекта | 1.11.2024 | 1 месяц | 2.12.2024 |
| 8 | Ведение дневника бармена | Руководитель проекта | 1.11.2024 | 1 месяц | 2.12.2024 |
| 9 | Ведение дневника повара | Руководитель проекта | 1.11.2024 | 1 месяц | 2.12.2024 |
| 10 | Ведение дневника шеф-повара | Руководитель проекта | 1.11.2024 | 1 месяц | 2.12.2024 |

**ЛАБОРАТОРНАЯ РАБОТА №7**

**Применение структурного подхода к проектированию ИС.**

**Реализация фазы структурного анализа: анализ поведения системы (построение DFD диаграммы верхнего уровня)**

**Задание 1.** Построить контекстную диаграмму верхнего уровня иерархии (начальную) для проектируемой ИС на бумаге (рисунок 7.1) и оформить ее в электронном виде с помощью программного средства MS Visio**.** (рисунок 7.2)

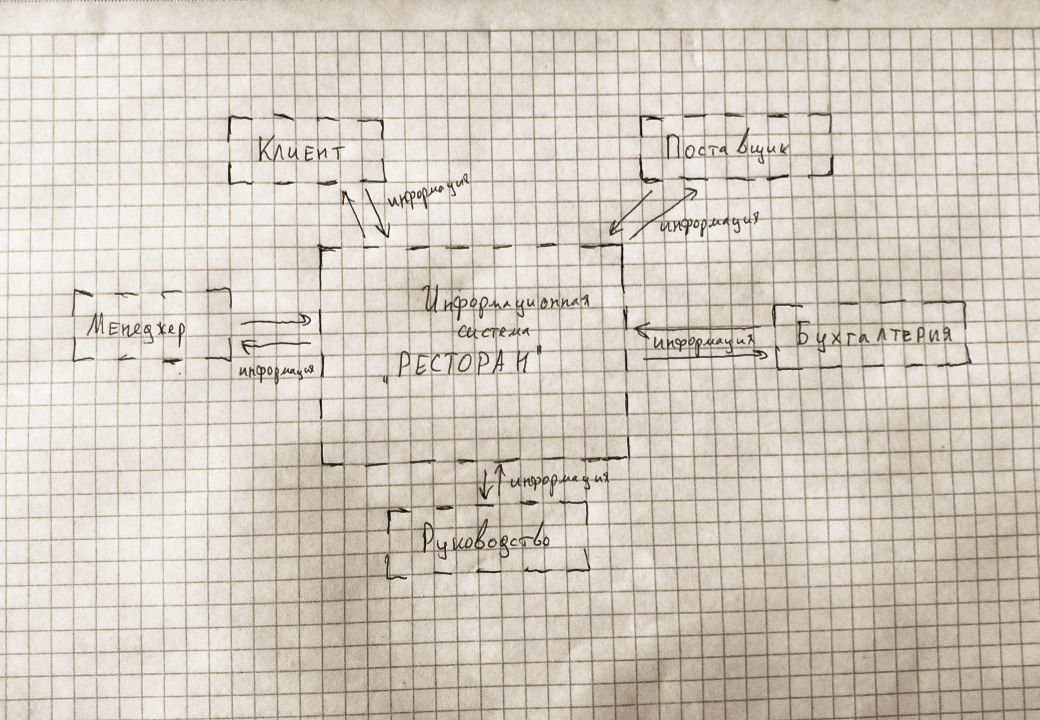


Рисунок 7.1 – Контекстная диаграмма

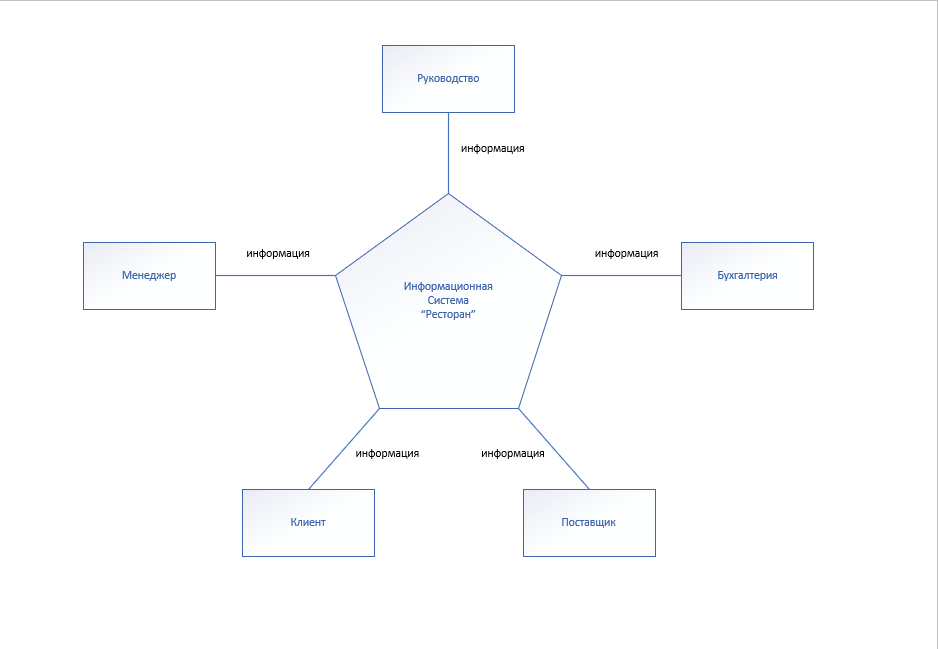


Рисунок 7.2 - Контекстная диаграмма в Visio

**Задание 2.** Построить в файле Word матрицу списка событий.

Список событий строится в виде матрицы (ELM) и описывает различные действия внешних сущностей и реакцию ИС на них. (таблица 7.1)

Таблица 7.1 – Матрица списка событий

|  |  |  |
| --- | --- | --- |
| Описание | Тип | Реакция |
| Клиент желает совершить заказ (покупку) | ND | Регистрация заказа (покупки) |
| Обслуживание клиента | ND | Продажа товара |
| Руководство предоставляет полномочия новому поставщику | ND | Регистрация поставщика |
| Поставщик сообщает об изменении адреса | ND | Регистрация измененного адреса поставщика |
| Поставщик направляет продукты | ND | Получение продуктов |
| Руководство запрашивает новый отчет | ND | Формирование требуемого отчета для руководства |

**Нормальные данные** – это такие данные, которые обеспечивают корректную и быструю обработку запросов системой. Нормальность данных обеспечивается в дальнейшем (при построении логической модели данных) путем проверки их на нормальность и если необходимо, приведением отношений к 1-ой, 2-ой и 3-ей нормальной форме.

**Задание 3.**

Для дальнейшей конкретизации контекстной диаграммы верхнего уровня необходимо определить процессы, которые будут учтены в ИС: дать им имя (например, зарегистрировать, оформить заявку – начинаются с конкретного глагола) и пронумеровать их. Процессы обозначаются окружностью в нотации Йодана., далее надо расщепить абстрактные потоки диаграммы верхнего уровня на реальные, конкретизировав информацию (циркулирующую в этих потоках), которая будет направляться в ИС через процессы к сущностям и от них. Все потоки также поименовать так, чтобы было понятно какая информация с ними связана. Для этого необходимо построить список потоков.

Существующие "абстрактные" потоки данных между терминаторами и процессами трансформируются в потоки, представляющие обмен данными на более конкретном уровне. Список событий показывает, какие потоки существуют на этом уровне: каждое событие из списка должно формировать некоторый поток (событие формирует входной поток, реакция - выходной поток). Один "абстрактный" поток может быть разделен на более чем один "конкретный" поток. Если входной и выходной потоки связаны с одним набором данных, то они могут иметь одно и тоже имя. Такие потоки описываются в дальнейшем один раз.

Список потоков представлен в таблице 7.2

Таблица 7.2 – Список потоков

|  |  |
| --- | --- |
| Потоки на диаграмме верхнего уровня | Потоки на диаграмме нулевого уровня |
| Информация от клиента | Запрос о заказе |
| Информация для клиента | Выдача чека |
| Информация от руководства | Запрос отчета о заказах, запрос отчета о поставщиках |
| Информация для руководства | Отчет о заказах, отчет о поставщиках |
| Информация для поставщика | Данные о продуктах |
| Информация от поставщика | Поставка продуктов, информация для подписания договора |
| Информация для кухни | Запрос на выполнение заказа, поставка продуктов |
| Информация от кухни | Меню, заказ на поставку, информация о продуктах |
| Информация для менеджера | Выполнение заказа, заказ |
| Информация от менеджера | Сохранение информации о заказе, запрос на выполнение заказа |

**ЛАБОРАТОРНАЯ РАБОТА №8**

**Применение структурного подхода к проектированию ИС.**

**Реализация фазы структурного анализа: анализ поведения системы (иерархия DFD диаграммы)**

**Задание 1.**

После конкретизации процессов и потоков построить DFD диаграмму нулевого уровня иерархии. (Рисунок 8.1) Она строится на основе диаграммы верхнего уровня путем анализа и определения процессов и расщепления потоков.

Групповой узел. Предназначен для расщепления и объединения потоков. В некоторых случаях может отсутствовать (т.е. фактически вырождаться в точку слияния/расщепления потоков на диаграмме).

Узел-предок. Позволяет увязывать входящие и выходящие потоки между детализируемым процессом и детализирующей DFD.

Неиспользуемый узел. Применяется в ситуации, когда декомпозиция данных производится в групповом узле, при этом требуются не все элементы входящего в узел потока.

Узел изменения имени. Позволяет неоднозначно именовать потоки, при этом их содержимое эквивалентно. Например, если при проектировании разных частей системы один и тот же фрагмент данных получил различные имена, то эквивалентность соответствующих потоков данных обеспечивается узлом изменения имени. При этом один из потоков данных является входным для данного узла, а другой - выходным.

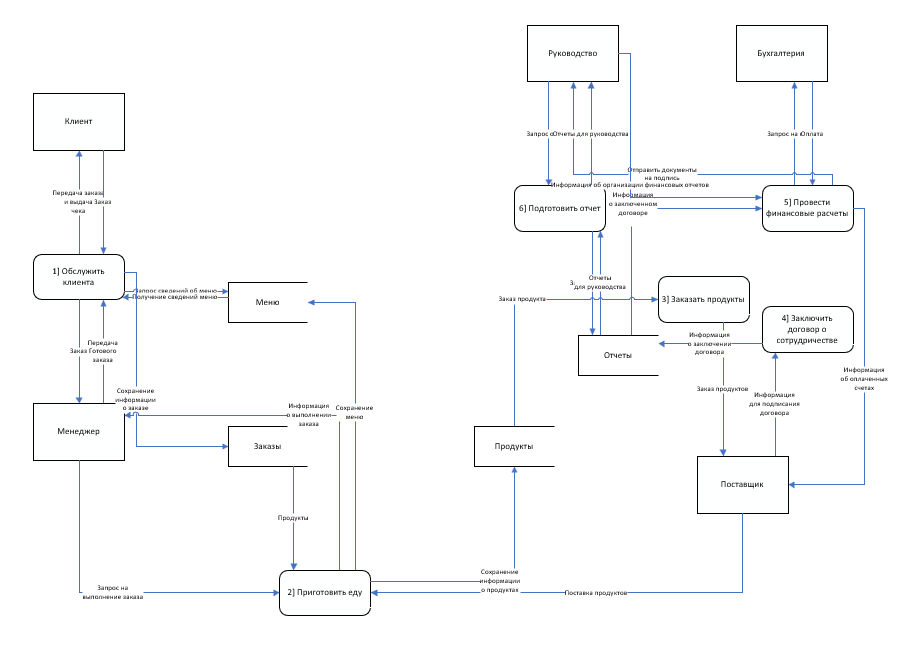


Рисунок 8.1 DFD (контекстная) диаграмма нулевого уровня иерархии

**Задание 2.**

Проанализировать созданные диаграммы верхнего и нулевого уровня иерархии и привести их в строгое соответствие с нотацией Гейна-Сарсона. (рисунок 8.2)

Основными компонентами диаграмм потоков данных являются:

* внешние сущности;
* системы/подсистемы;
* процессы;
* накопители данных;
* потоки данных.

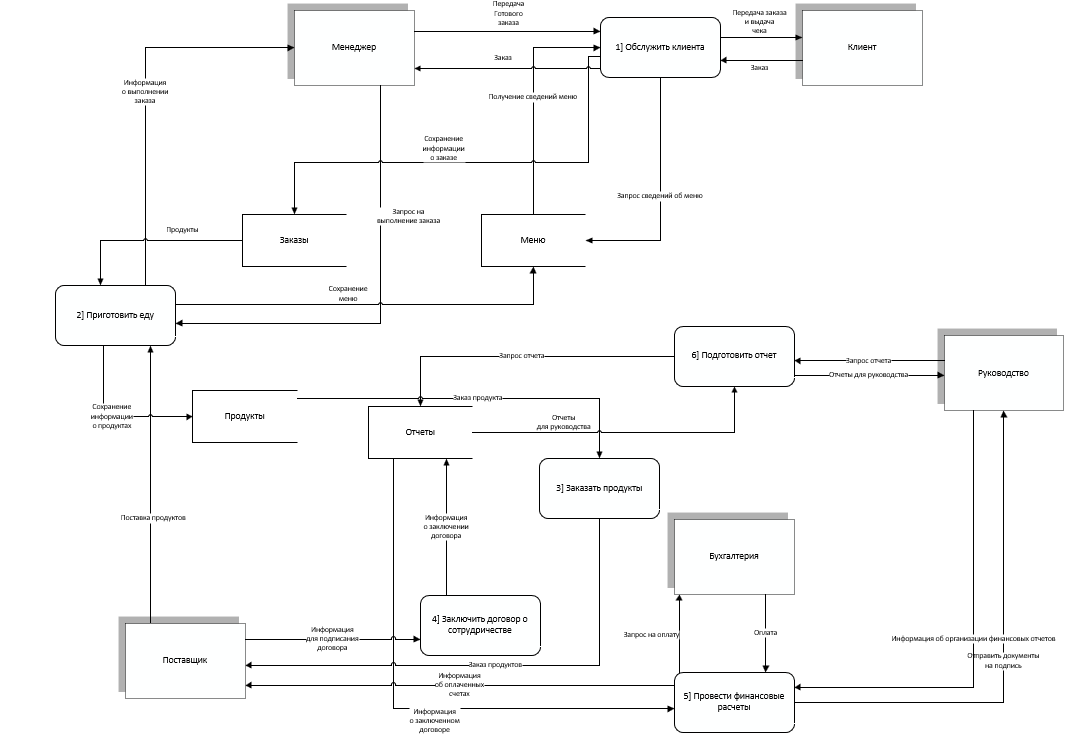


Рисунок 8.2 – DFD диаграмма нулевого уровня иерархии (приведенная)

**ЛАБОРАТОРНАЯ РАБОТА №9**

**Основы работы в редакторе деловой графики Microsoft Visio.**

**Изучение возможностей и настройка режимов работы**

**Задачи исследований**

1. Изучить операции по запуску MS Visio на персональном компьютере.

2. Познакомиться с элементами заставки, выдаваемой на экран системой, и элементами рабочего экрана MS Visio.

3. Рассмотреть основные возможности работы с векторной графикой в программе MS Visio.

Ход работы:

Запустить редактор MS Visio 2010. Выбрать категорию шаблонов «Общие», шаблон «Простая блок-схема», нажать на кнопку «Создать». (рисунок 9.1)

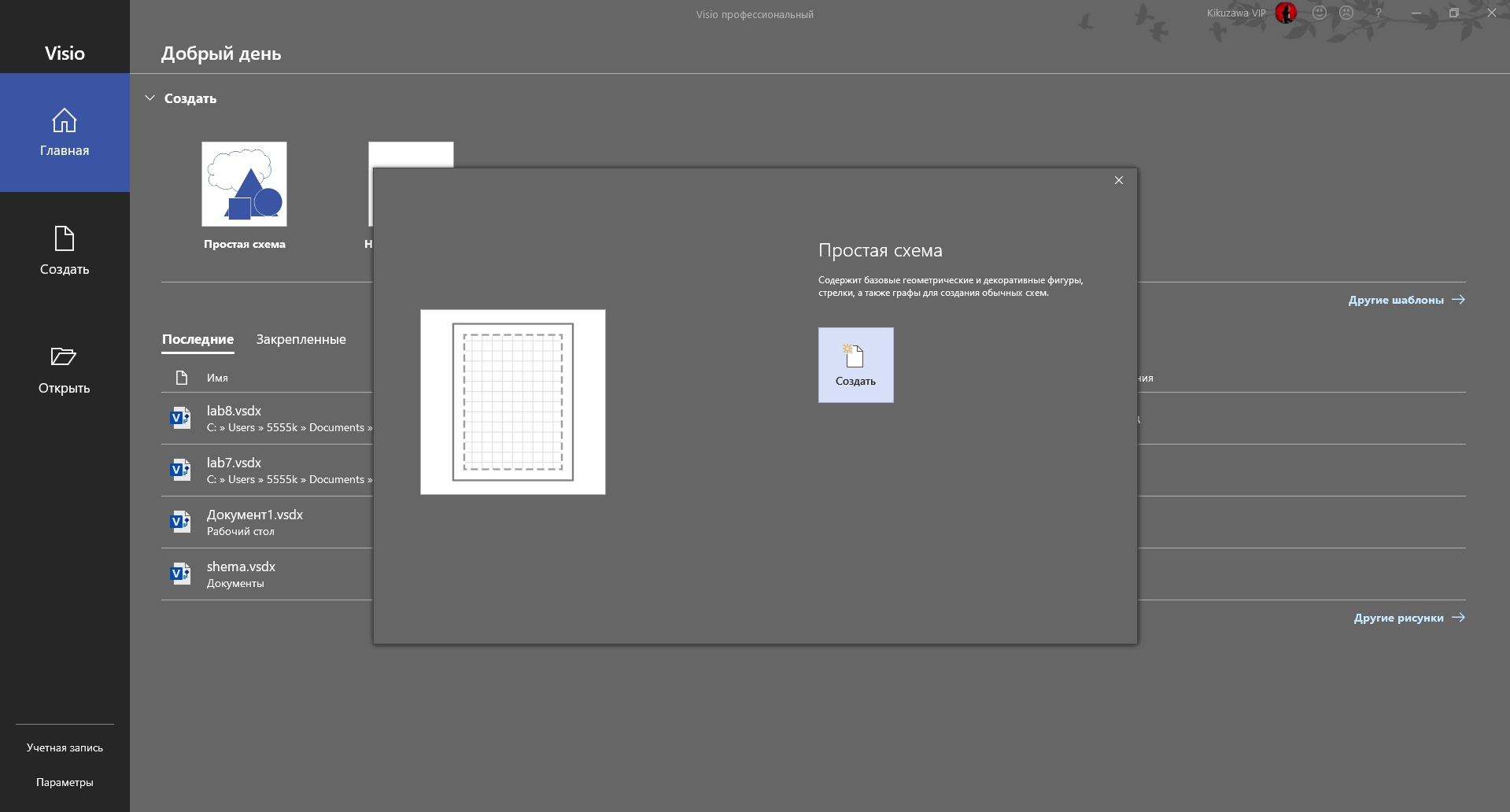


Рисунок 9.1 – Выбор шаблона

Изучить элементы рабочего экрана MS Visio. Окно MS Visio состоит из нескольких основных стандартизированных частей (всё это присутствует на рисунке 9.2):

* Команды меню.
* Панель инструментов.
* Рабочая область (в центре экрана).
* Окно фигур – как правило, слева от рабочей области.
* Строка состояния, на которой чаще всего отображаются параметры выделенного объекта. (
* Полосы прокрутки, ярлычки страницы и кнопки навигации (перемещения) между ними.
* Стартовая панель задач - как правило, справа от рабочей области и другие элементы, которые могут выводиться на экран и убираться с него пользователем.

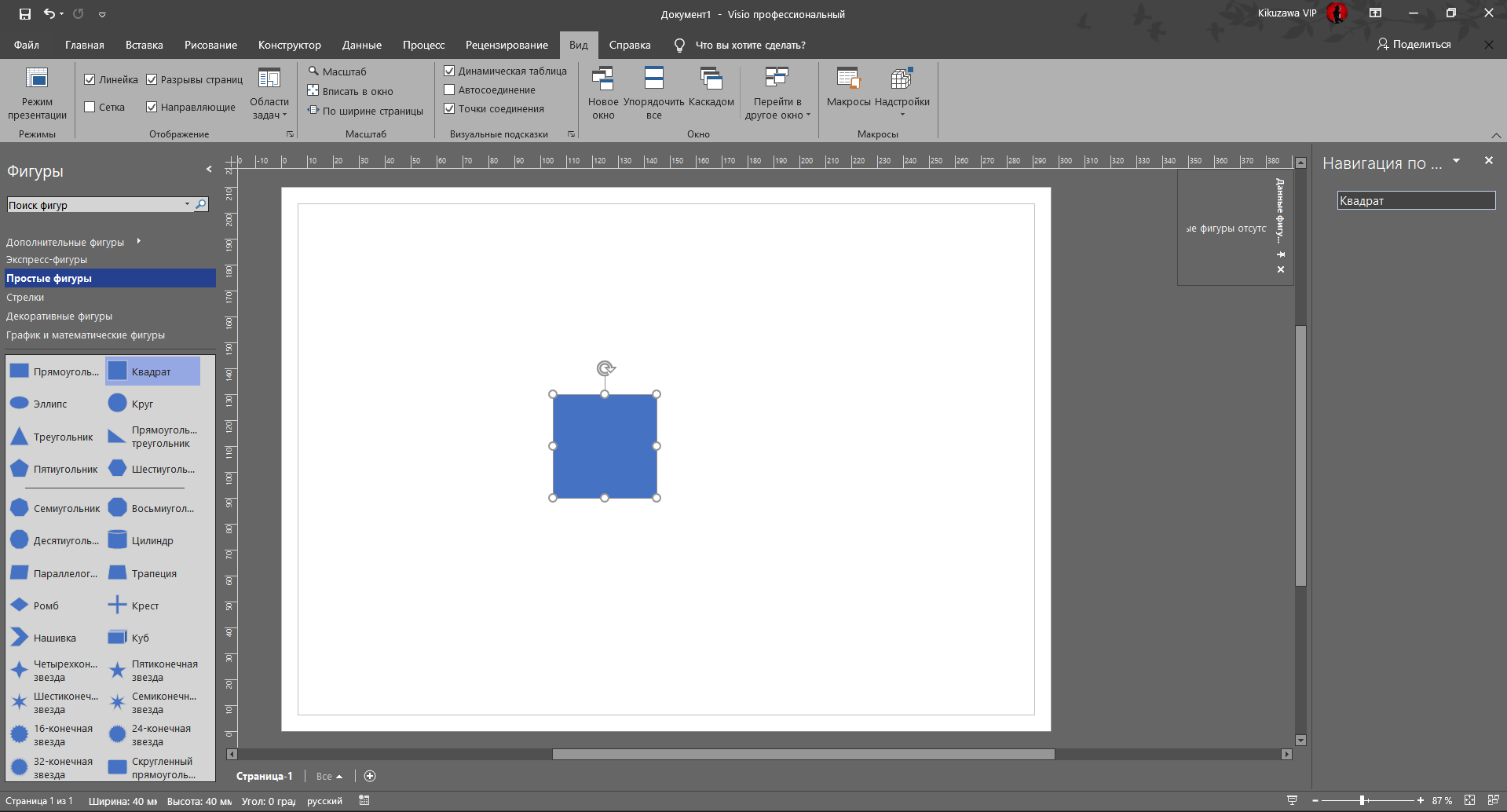


Рисунок 9.2 – Рабочий экран Visio

Чтобы работать с шаблонами (наборами элементов), обычно настройки можно найти в Фигуры -> Дополнительные фигуры -> Мои фигуры и там уже с ними работать

Для работы с многостраничным документов ниже страницы документа расположены вкладки страниц, облегчающие перемещение между страницами многостраничного документа. Также здесь можно переименовывать страницу, создавать или удалять (рисунок 9.3)

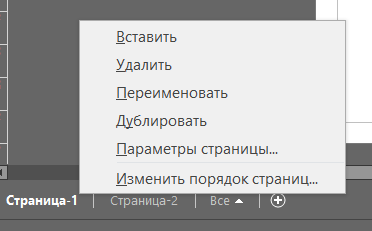


Рисунок 9.3 – Работа с многостраничным документом

Есть несколько способов выделения элементов. Самый простой и выделить объекты на страничке с помощью мыши или простым нажатий с нажатой кнопкой shift или ctrl. Также их можно выделить с помощью специальный инструментов, например, лассо (рисунок 9.4)

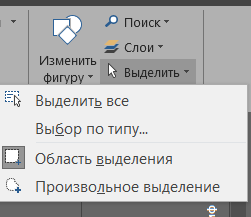


Рисунок 9.4 – Способы выделения

Для форматирования элементов все инструменты находятся на главной вкладке инструментов. Там можно найти форматирование текста, линий, заливки, теней. (рисунок 9.5)

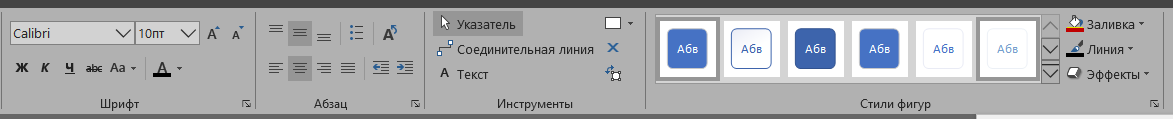


Рисунок 9.5 – Инструменты форматирования

Для включений функций защиты фигур и их поведения, необходимо запустить режим разработчика (рисунок 9.6) и затем все эти функции отобразятся в новой вкладке инструментов разработчика

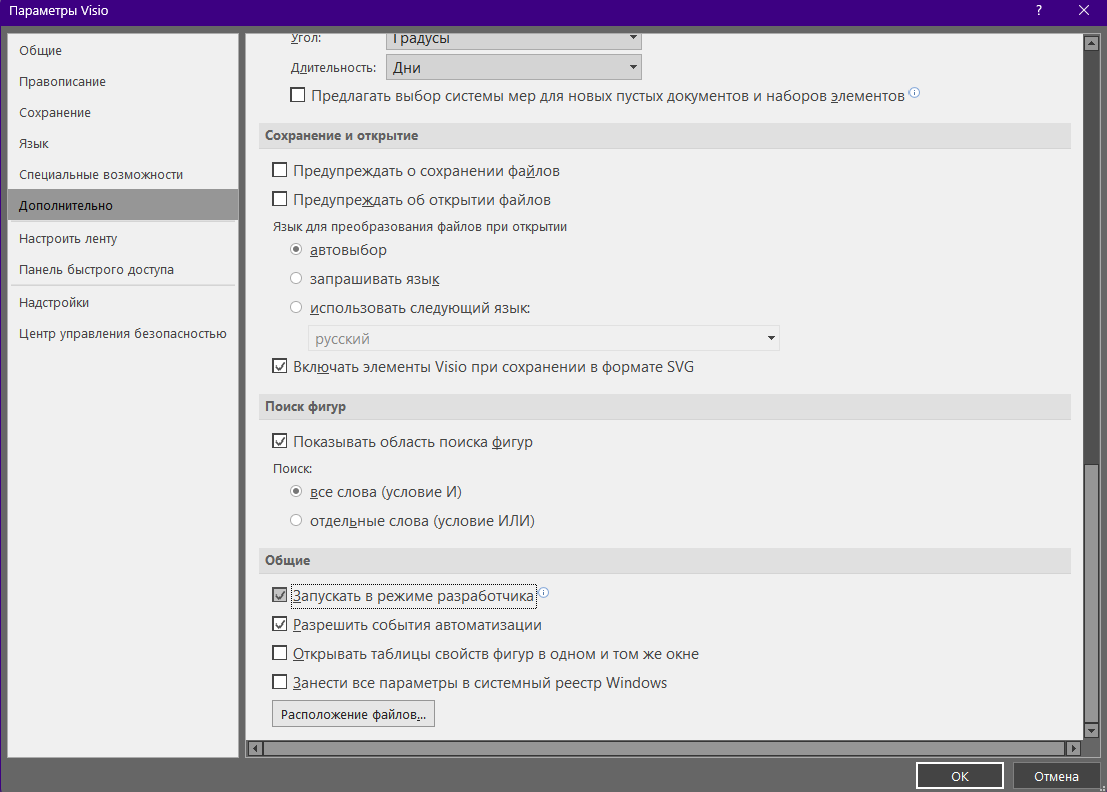


Рисунок 9.6 – Включение режима разработчика

Также есть различные взаимодействия с фигурами, например, соединять фигуры. Все фигуры (символы) на схеме должны быть соединены между собой линиями. Эти линии показывают направление потока информации. Visio позволяет легко соединять фигуры (символы) диаграмм и блок-схем с помощью коннекторов (соединительных линий) – одномерных фигур. Эти фигуры представляют собой линии или стрелки. При перемещении соединенных фигур (символов) коннекторы остаются присоединенными к перемещаемым фигурам. Чтобы разбить эту связь, нужно переместить концевую точку коннектора или удалить его совсем. В блок-схемах можно поместить двумерную фигуру между двумя соединенными фигурами, и Visio перестроит линии соединения и свяжет последовательно все три фигуры.

При изменении соединений между фигурами и перемещении соединенных фигур используется несколько специальных инструментов, которые помогают правильно расположить и выровнять фигуры. В документе также можно изменить ориентацию всех соединенных фигур – например, можно изменить макет блок-схемы, повернув вверх ногами или слева направо.

В Visio можно создать два типа соединений: соединения между фигурами и соединения между точками фигур. Соединение между фигурами выполняется по двум ближайшим точкам фигур. Соединение между точками фигур связывает их по конкретным точкам соединения.

При рисовании коннектора его концевые точки «приклеиваются» к фигурам. В результате создается связь, которая не разорвется до тех пор, пока не будет перемещена концевая точка коннектора или он не будет удален. При выделении коннектора, «приклеенного» к фигуре, его концевая точка становится красной, указывая, что при перемещении соединенных фигур коннектор изменит свое расположение. Для соединения фигур существует несколько способов.

Первый способ

Сначала поместить все необходимые фигуры (символы) блок-схемы на страницу документа. Затем на стандартной панели инструментов щелкнуть по кнопке Соединительная линия (Connector Tool) . Указатель измениться на пиктограмму коннектора. Подвести его к Точке соединения фигуры и, не отпуская левую кнопку мыши, протянуть к Точке соединения следующей фигуры. Отпустить кнопку мыши в тот момент, когда место склеивания на экране выделится красным цветом.

Второй способ

Если перед добавлением второй и последующих фигур на стандартной панели инструментов щелкнуть (включить команду) по кнопке Соединительная линия (Connector Tool), то фигуры автоматически соединются друг с другом линиями, показывающими последовательность выполняемых действий пользователем.

Для добавлений подписи в фигуре достаточно нажать по ЛКМ двойным щелчком. Для разворота текста на соединительной линий достаточно нажать кнопку переворота текста на главной вкладке в поле абзац

Для группировки есть специальное поле во главной вкладке (рисунок 9.7)

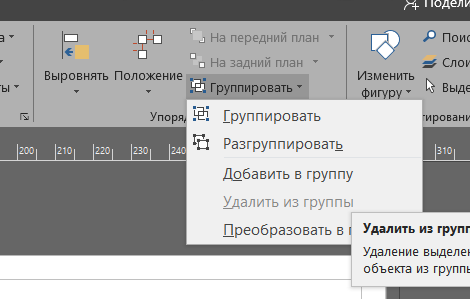


Рисунок 9.7 – Работа с группировкой

Там же есть инструменты выравнивания и порядка расположения фигур, также зеркалирование.

Для копирования фигур можно использовать типичное сочетание клавиш или контекстное меню ПКМ

Операции с фигурами также находятся на главной вкладке, как и работа с текстом.

Работа со слоями также есть на главной вкладке в поле редактирование. Там можно как раз ими и управлять.

Для работы с документом есть инструменты во вкладке конструктор

Для сохранения или копирования схем и отдельных частей есть функционал на главной вкладке в поле Буфер Обмена

Проверка в орфографии находится во вкладке рецензирование

Сохранение аналогично происходит, как в обычных программах, например, MS Word

Защита документа происходит во вкладке разработчик в проводнике по рисункам, там ПКМ по документу и в контекстном меню

Контрольные вопросы:

1. Отличительные особенности векторной графики

Векторная графика имеет следующие ключевые особенности:

* Шаблонность: объекты можно масштабировать без потери качества
* Гибкость: легко изменять размеры и формы элементов
* Компактность: файлы занимают меньше места по сравнению с растровой графикой
* Простота редактирования: можно легко добавлять или удалять элементы

2. Основные части окна Microsoft Visio

Окно Visio состоит из следующих основных частей:

* Панель инструментов (Ribbon) с различными вкладками
* Плоскость рисования (canvas) для размещения элементов
* Панель Shapes для выбора и добавления фигур
* Рукоятка быстрого доступа к часто используемым командам
* Стартовая страница с шаблонами и недавними документами

3. Организация многостраничных документов в MS Visio

В Visio многостраничные документы организуются следующим образом:

* Каждая страница представляет собой отдельный лист
* Листы можно переключать между ними
* Возможна навигация по листам через панель слайдов или кнопки в верхней части окна
* Доступны возможности для гиперссылок между листами

4. Основные команды меню и панели инструментов MS Visio

Основные категории команд в Visio включают:

* Группировка и разгруппировка объектов
* Форматирование текста и стилей
* Создание и редактирование связей между объектами
* Настройка макета и печати
* Инструменты для работы со слоями
* Тools для управления стилями и форматированием

5. Шаблоны MS Visio и их возможности

Шаблоны в Visio предоставляют следующие возможности:

* Библиотека готовых шаблонов различных типов диаграмм
* Возможность создания собственных шаблонов
* Экспорт шаблонов в другие проекты
* Настройка параметров шаблонов (цвета, размеры, стили)
* Использование шаблонов как отправной точки для создания новых диаграмм

6. Возможности выделения элементов в MS Visio

Visio предлагает следующие способы выделения элементов:

* Селекция мышью
* Команды выделения (Shift+Click, Ctrl+A)
* Условное выделение (Ctrl+8)
* Выделение по группам
* Автоматическое выделение связанных элементов

7. Группировка и разгруппировка элементов в MS Visio

Группировка и разгруппировка выполняется следующим образом:

* Группировка: Ctrl+G
* Разгруппировка: Ctrl+Shift+G
* Возможность группировки по типу, цвету, размеру
* Переименование групп
* Изменение свойств группы (визуальное отображение, защита)

8. Соединение элементов в MS Visio

Соединение элементов в Visio осуществляется:

* Через специальные инструменты соединения
* При помощи автоматических соединений (AutoConnect)
* С помощью ручного рисования линий
* Через свойства объектов (например, коннекторные точки)
* С использованием мастера связи

9. Форматирование элементов в MS Visio

Форматирование элементов включает:

* Изменение цвета и заливки
* Настраивание границ и тени
* Применение стилей и шаблонов
* Изменение размеров и масштаба
* Применение эффектов (тени, прозрачность)
* Настройка текстового содержимого

10. Распределение и выравнивание фигур в MS Visio

Распределение и выравнивание выполняются:

* Через инструменты распределения на панели инструментов
* С использованием направляющих и сетки
* При помощи мастеров выравнивания (Align)
* Через свойства объектов (Snap и Glue)
* С применением макетов и ограничителей

11. Работа с текстом в MS Visio

Visio предоставляет следующие возможности для работы с текстом:

* Вставка текстовых блоков
* Форматирование текста (цвет, шрифт, размер)
* Выравнивание текста
* Создание многострочных текстовых областей
* Использование таблиц для организации текста
* Вставка изображений и символов

12. Роль слоев в MS Visio

Слои в Visio используются для:

* Организации элементов в разных уровнях
* Изоляции изменений
* Управления видимостью элементов
* Обеспечения совместной работы над документом
* Сохранения истории версий
* Улучшения производительности при работе с большими проектами

13. Настройка параметров изображения в MS Visio

Настройка параметров изображения включает:

* Установку разрешения
* Выбор типа растровой графики (RGB, CMYK)
* Настройку качества компрессии
* Оптимизацию для печати или онлайн-просмотра
* Управление цветопередачей
* Настройка размера и масштаба изображения

14. Сохранение документа и копирование изображения в MS Visio

Сохранение и копирование выполняются:

* Сохранение как Visio файл (\*.vsdx): File > Save As
* Экспорт в другие форматы (PDF, PNG, JPEG): File > Export
* Копирование изображения: Ctrl+C, затем вставка в другое приложение
* Отправка эскиза по email: File > Share > Send Email Attachment

15. Изменение масштаба изображения и перемещение по документу в MS Visio

Изменение масштаба и перемещение выполняются:

* Масштабирование: Ctrl++ (увеличение), Ctrl+- (ублюдение)
* Перемещение: Shift+Arrow keys
* Навигация по страницам: Page Up, Page Down
* Зум: Wheel mouse или Zoom tool
* Настройка зума: View > Zoom Level

16. Проверка орфографии в документе MS Visio

Проверка орфографии в Visio:

* Используйте встроенный проверщик орфографии
* Настройте язык и правила правописания
* Проверьте весь документ или выделенную область
* Используйте автозагрузку правописания

17. Защита документа в MS Visio

Защита документа в Visio включает:

* Защиту файла с паролем
* Ограничение прав доступа для пользователей
* Использование информационного права управления (IRM)
* Защита слоев и групп

**ЛАБОРАТОРНАЯ РАБОТА №10**

**Создание DFD –диаграмм в MS Visio**

**Цель работы:** изучение основных характеристик и основ работы с DFD-моделями в графическом редакторе Microsoft Visio 201

**Задачи исследований**

1. Изучить операции по созданию DFD-модели в редакторе MS Visio.

2. Изучить состав диаграмм DFD, назначение элементов каждого вида и способы их размещения на диаграмме в MS Visio.

3. Изучить возможности изменения внешнего вида элементов DFD-диаграмм и управления внешним видом соединений.

Ход работы:

Запустим редактор Visio. Выберем категорию шаблонов Программы и базы данных, шаблон Схема модели потоков данных (рисунок 10.1)

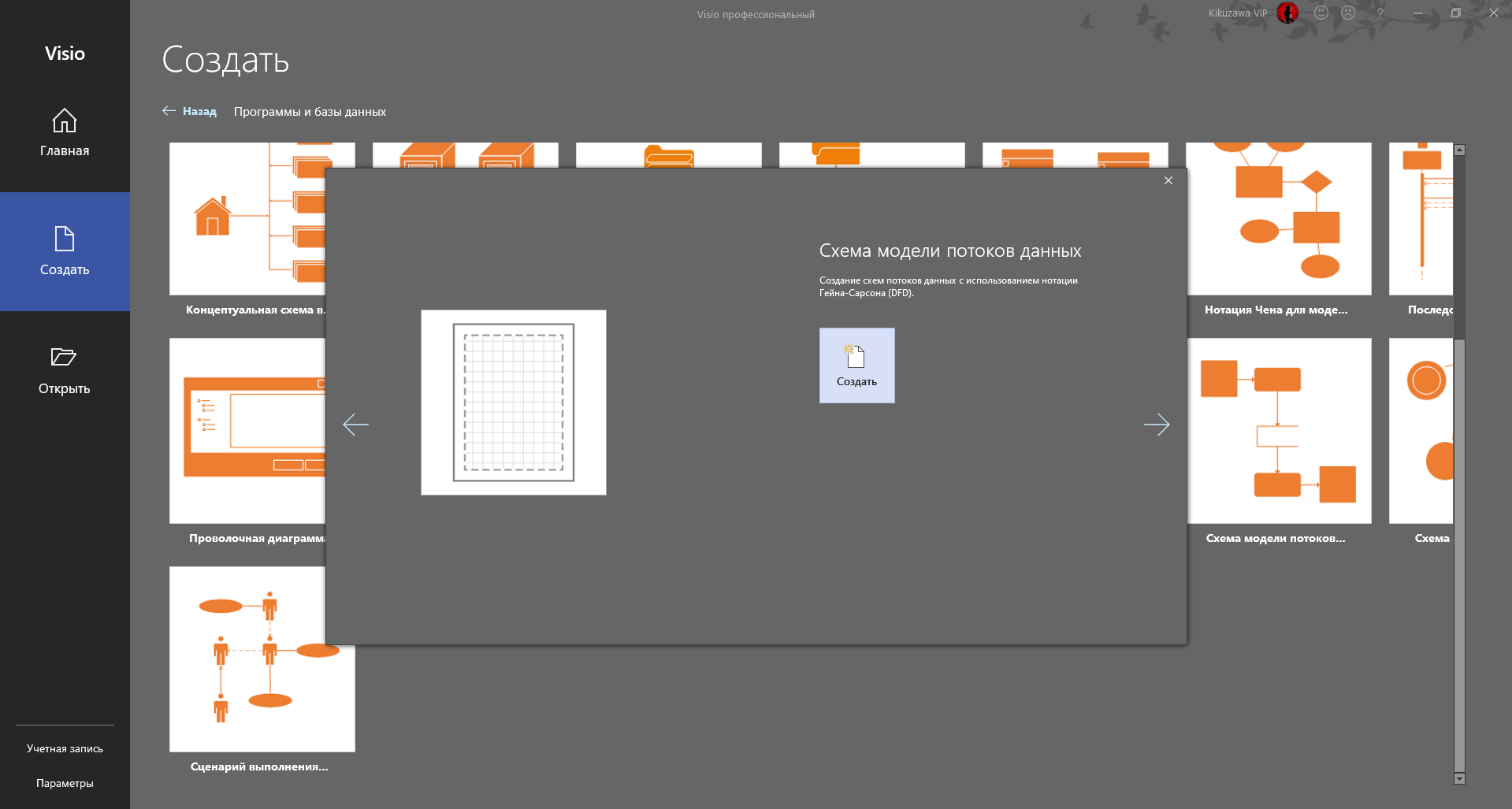


Рисунок 10.1 – Выбираем шаблон

Странице нового документа дадим название «DFD контекст». (Рисунок 10.2)

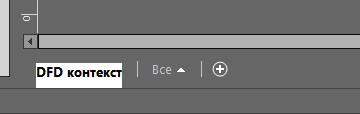


Рисунок 10.3 – Даем название DFD контекст

Создаем новую модель в стандарте DFD и разместим на данной странице элементы DFD-диаграммы в соответствии с обозначениями по нотации   
Гейна-Сарсона. Модель представлена на рисунке 10.4

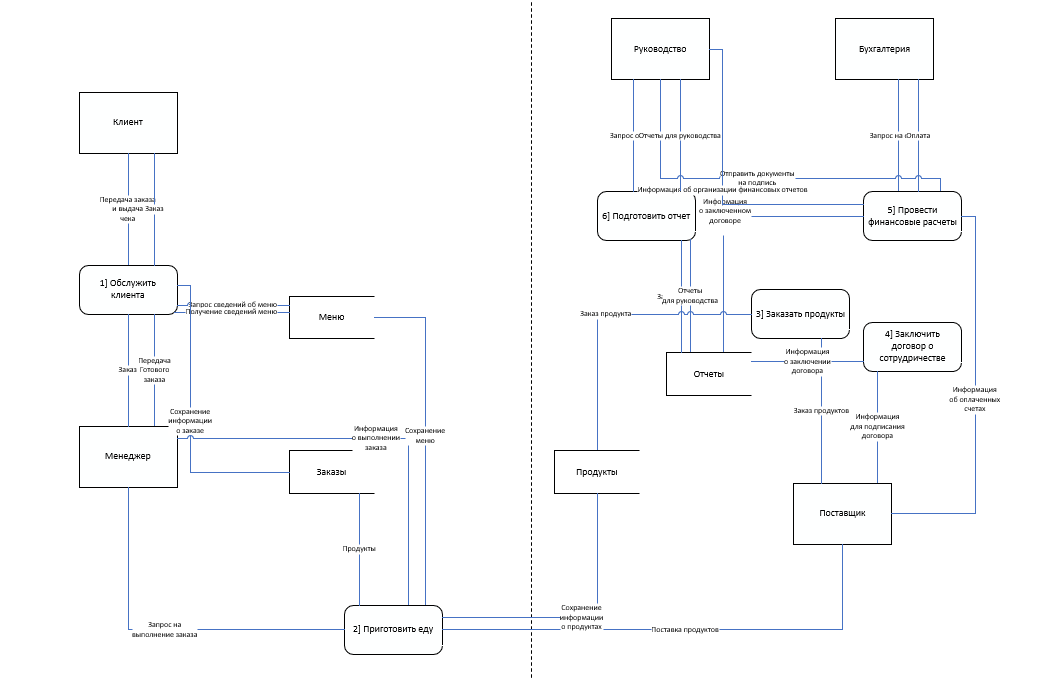


Рисунок 10.4 – Модель DFD (черновая)

Для элементов «Внешняя сущность» («Интерфейс») рекомендуется задать тень, так как это сильнее будет подчеркивать их визуальное отличие от функциональных блоков, и более будет приближено к стандарту DFD. Для задания тени нужно выделить элемент «Внешняя сущность», нажать клавишу F3 (или выбрать команду меню Формат – Заливка), в появившемся окне «Заливка» в категории «Тень» выбрать стиль «05: Смещение, вверх влево». Причем цвет тени не обязательно делать абсолютно черным, чтобы тень не сливалась со стрелкой. Напомним, что для корректного отображения стрелок на DFD-диаграмме в нотации Гейна-Сарсона на соединяемые фигуры нужно добавить точки соединения, а для смещения подписей использовать изменение полей или элемент «Подписи» из шаблона «Фигуры схемы IDEF0». Стрелки подписей в виде молний можно не отображать если задать им цвет белый или прозрачный. В результате проделанных действий контекстная диаграмма будет выглядеть примерно так, как показано на 10.5

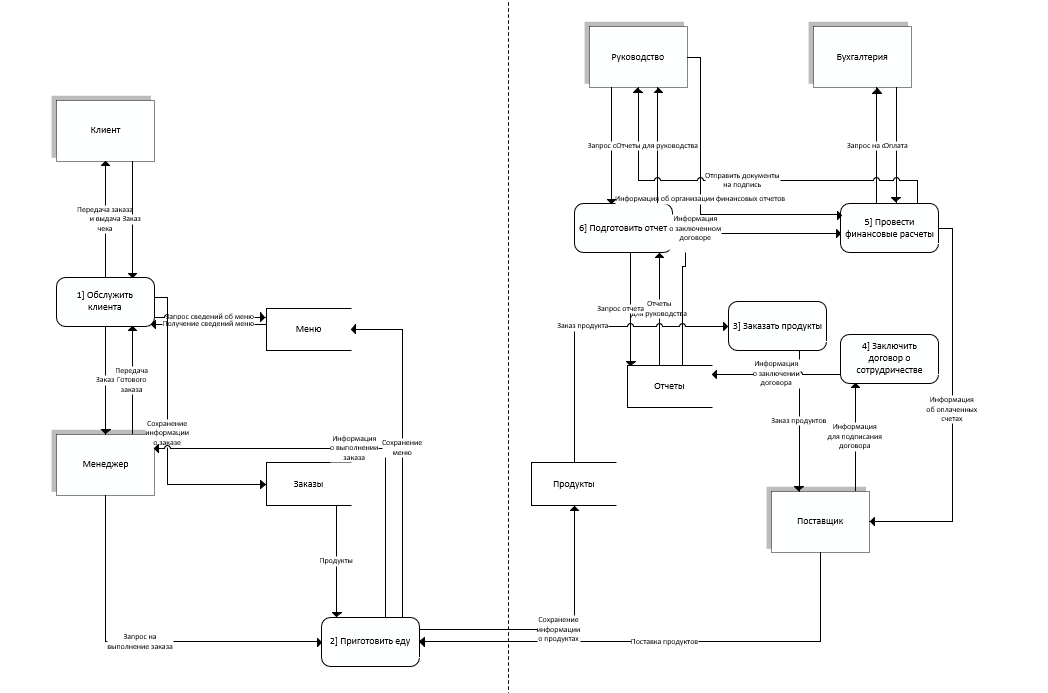


Рисунок 10.5 – Контекстная диаграмма DFD

Теперь произведем анализ недостатков существующей модели AS-IS:

1. Наличие лишних функциональных блоков

* Неэффективное использование ресурсов
* Повышенная сложность системы
* Потенциальное снижение производительности
* Усложнение поддержки и обновления
* Возможное дублирование функций

2. Отсутствие необходимых функциональных блоков

* Недостаточная обработка данных
* Отсутствие необходимых бизнес-функций
* Неполнота системного обеспечения
* Сложности в интеграции с другими системами

3. Наличие лишних потоков данных

* Увеличение нагрузки на систему
* Потенциальное снижение производительности
* Повышенные затраты на обработку и хранение данных
* Усложнение мониторинга и аудита

4. Отсутствие необходимых потоков данных

* Недостаточная синхронизация данных между системами
* Отсутствие реального времени обновлений
* Сложности в принятии оперативных решений
* Неполная картина business intelligence

5. Дублирование функций

* Неэффективное использование ресурсов
* Повышенная сложность системы
* Увеличение вероятности ошибок
* Сложности в поддержке и обновлении
* Потенциальное снижение производительности

6. Отсутствие необходимых хранилищ данных

* Недостаточное хранение исторических данных
* Отсутствие возможности архивации
* Сложности в восстановлении данных
* Неполная картина business intelligence
* Ограниченные возможности анализа

7. Неучтенное поступление информации от внешних источников

* Несогласованность данных
* Сложности в интеграции
* Потенциальное несоответствие бизнес-правилам
* Увеличение рисков безопасности
* Сложности в управлении данными

8. Нехватка гибкости системы

* Сложности в адаптации к изменениям бизнес-требований
* Ограниченные возможности расширения
* Сложности в миграции к новым технологиям
* Неполная поддержка мобильных рабочих мест

9. Отсутствие единства в интерфейсе пользователя

* Сложности в обучении персонала
* Нежелательное влияние на продуктивность
* Отсутствие единообразия в работе с системой
* Сложности в поддержке пользователей

Теперь для каждого хранилища данных, представленного на DFD-диаграмме привести описание с указанием его типа (бумажный носитель, таблица Excel, таблица базы данных, созданной в определенной СУБД и др.), и роли в перемещении и хранении потоков данных.

**1. Хранилище данных "Меню"**

Тип: Бумажный носитель (принципально, но может быть электронным)

Описание:

* Это физическое хранилище, где находятся текущие блюда и напитки, предлагаемые рестораном.
* Содержит список всех продуктов, цен, специальных предложений и сезонных блюд.

Роль в перемещении и хранении потоков данных:

* Хранит постоянную информацию о меню, которую регулярно обновляют.
* Может служить отправной точкой для загрузки данных в электронные системы.
* Предоставляет актуальную информацию о продуктах для других процессов.

**2. Хранилище данных "Заказы"**

Тип: Таблица базы данных (например, MySQL или PostgreSQL)

Описание:

* Электронное хранилище, которое записывает все заказы клиентов.
* Содержит детали заказов, включая дату, время, номер столика, товары и количество.

Роль в перемещении и хранении потоков данных:

* Получает данные от POS-системы или кассового аппарата.
* Передает информацию о заказах в систему отчетов.
* Может экспортироваться для анализа продаж и статистики.

**3. Хранилище данных "Продукты"**

Тип: Таблица базы данных (аналогично "Заказы")

Описание:

* Хранилище, содержащее информацию о всеми продаваемых продуктах.
* Включает детали о каждом продукте, такие как название, описание, цена, состав.

Роль в перемещении и хранении потоков данных:

* Поддерживает обновление цен и описаний продуктов.
* Предоставляет актуальную информацию о продуктах для системы меню.
* Может использоваться для генерации отчетов о популярности продуктов.

**4. Хранилище данных "Отчеты"**

Тип: Таблица базы данных (аналогично "Заказы" и "Продукты")

Описание:

* Хранилище, содержащее различные виды отчетов, используемых рестораном.
* Может включать финансовые отчеты, отчеты о продажах, статистику посещений.

Роль в перемещении и хранении потоков данных:

* Получает данные из других хранилищ (Заказы, Продукты).
* Генерирует новые отчеты на основе имеющихся данных.
* Может экспортироваться в различные форматы для анализа и презентаций.

Дополнительные замечания:

1. Все эти хранилища могут взаимодействовать друг с другом через потоки данных, например, когда система отчетов использует данные из хранилищ Заказы и Продуктов.
2. Для повышения надежности и безопасности, рекомендуется использовать резервные копии всех баз данных.
3. Возможно использование дополнительных хранилищ, таких как:
   * Архивные данные о прошлых заказах (для длительного хранения)
   * Фото продуктов (для меню и маркетинговых материалов)
   * Информация о сотрудниках (для HR-процессов)
4. При необходимости можно добавить виртуальные таблицы в базу данных для агрегации данных из нескольких источников.
5. Для больших объемов данных может потребоваться использование распределенных хранилищ или облачных сервисов.

**Контрольные вопросы:**

1. Назначение стандарта DFD

Стандарт DFD (Data Flow Diagram) предназначен для:

* Показания потоков данных между процессами и хранилищами в системе
* Иллюстрации процесса обработки данных
* Анализа и проектирования информационных систем
* Объяснения сложных процессов обработки данных
* Коммуникации между разработчиками и пользователями системы

2. Создание схемы DFD в MS Visio

Создание схемы DDF в MS Visio осуществляется следующим образом:

* Выбрать шаблон DFD из библиотеки Visio
* Использовать инструменты для рисования процессов, хранилищ данных и потоков данных
* Применять нотации Гейна-Сарсона для обозначения элементов
* Настроить стили и цвета для улучшения читаемости диаграммы

Нотации, используемые в Visio:

* Процесс
* Интерфейс
* Хранилище данных
* Поток данных

3. Роль основных элементов в стандарте DFD

Основные элементы DFD играют следующие роли:

* Процессы: представляют операции над данными
* Хранилища данных: содержат информацию, используемую системой
* Потоки данных: показывают направление и содержание передачи информации между элементами

4. DFD-модель AS-IS

DFD-модель AS-IS создается для:

* Описания текущей ситуации в системе
* Идентификации проблем и недостатков существующего процесса
* Анализа потоков данных и процессов обработки
* Предоставления базы для проектирования улучшенной модели To-Be

Модель AS-IS показывает:

* Все процессы, хранилища данных и потоки данных в существующей системе
* Наличие лишних функций и отсутствие необходимых
* Несоответствие между процессами и хранилищами данных
* Потенциальные проблемы в обработке информации

5. Элементы DFD-модели и их назначение

Элементы DFD-модели включают:

* Процессы: представляют операции над данными
* Хранилища данных: содержат информацию, используемую системой
* Потоки данных: показывают направление передачи информации
* Детали процесса: расшифровывают действия внутри процесса

Назначение этих элементов:

* Процессы: обозначают конкретные действия с данными
* Хранилища данных: показывают источники и приемники информации
* Потоки данных: иллюстрируют поток информации через систему
* Детали процесса: предоставляют детальную информацию о работе процесса

6. Логическое развитие диаграммы потоков данных модели AS-IS

Логическим развитием DFD-модели AS-IS является модель To-Be. Она создается для:

* Представления желаемого состояния системы
* Идентификации возможных улучшений в процессе обработки данных
* Описания новых или измененных процессов и потоков данных
* Показания потенциальных преимуществ новой модели

Модель To-Be должна:

* Улучшать существующую структуру данных и процессов
* Обеспечивать более эффективную обработку информации
* Упрощать взаимодействие между различными частями системы
* Поддерживать будущий рост и развитие бизнеса

**ЛАБОРАТОРНАЯ РАБОТА №11**

**Применение структурного подхода к проектированию ИС.**

**Реализация фазы структурного анализа: анализ данных**

**(построение DSD диаграмм)**

Задание. Построить диаграммы структур данных.

Анализ функционального аспекта поведения системы дает представление об обмене и преобразовании данных в системе. Взаимосвязь между "абстрактными" потоками данных и "конкретными" потоками данных на диаграмме нулевого уровня выражается в диаграммах структур данных

На рисунке 11.1 представлена диаграмма структур данных.

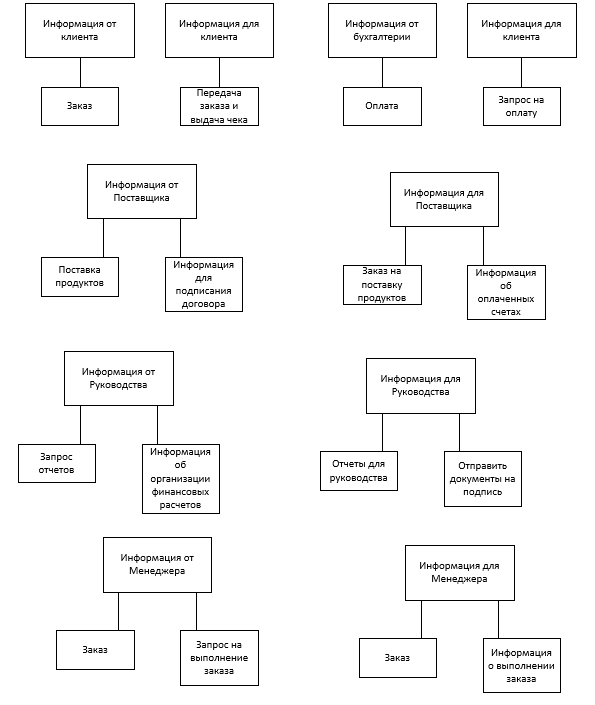


Рисунок 11.1 – Диаграмма структуры данных

**ЛАБОРАТОРНАЯ РАБОТА №12**

**Применение структурного подхода к проектированию ИС.**

**Реализация фазы структурного анализа: анализ поведения системы (построение словаря данных)**

Задание. Построить словарь данных. Сначала напечатать нумерованный список всех потоков. Далее последовательно описать все потоки, используя хотя бы такой минимум параметров:

@ИМЯ =

@ТИП =

@ЕДИНИЦЫ =

@БНФ =

@КОММЕНТАРИЙ =

Ниже представлен весь словарь данных и описание процессов

# Словарь данных

|  |
| --- |
| @ИМЯ = ЗАКАЗ  @ТИП = внешний поток  @ЕДИНИЦА = текстовая + числовая  @БНФ = Наименование + Количество  @КОММЕНТАРИЙ = указывает желаемый заказ клиента |
| @ИМЯ = ПЕРЕДАЧА ЗАКАЗА И ВЫДАЧА ЧЕКА  @ТИП = внутренний поток  @ЕДИНИЦА = текстовая + числовая  @БНФ = Наименование + Количество + Стоимость + Итого  @КОММЕНТАРИЙ = предоставление информации о чеке клиенту |
| @ИМЯ = ЗАПРОС СВЕДЕНИЙ О МЕНЮ  @ТИП = внутренний поток  @ЕДИНИЦА = текстовая + числовая  @БНФ = Наименование + Стоимость  @КОММЕНТАРИЙ = информация о меню |
| @ИМЯ = ПОЛУЧЕНИЕ СВЕДЕНИЙ О МЕНЮ  @ТИП = внутренний поток  @ЕДИНИЦА = текстовая + числовая  @БНФ = Наименование + Стоимость  @КОММЕНТАРИЙ = информация о меню |
| @ИМЯ = ИНФОРМАЦИЯ О ВЫПОЛНЕНИИ ЗАКАЗА  @ТИП = внутренний поток  @ЕДИНИЦА = Булево  @БНФ = Статус выполнения  @КОММЕНТАРИЙ = Отправка статуса выполнения заказа |
| @ИМЯ = ЗАПРОС НА ВЫПОЛНЕНИЕ ЗАКАЗА  @ТИП = внутренний поток  @ЕДИНИЦА = текст  @БНФ = Дата заказа + содержимое заказа  @КОММЕНТАРИЙ = Передача запроса на выполнение заказа |
| @ИМЯ = СОХРАНЕНИЕ ИНФОРМАЦИИ О ЗАКАЗЕ  @ТИП = внутренний поток  @ЕДИНИЦА = дата + текстовая  @БНФ = Дата заказа + содержимое заказа  @КОММЕНТАРИЙ = Сохранение в базу |
| @ИМЯ = СОХРАНЕНИЕ МЕНЮ  @ТИП = внутренний поток  @ЕДИНИЦА = текстовая + числовая  @БНФ = Наименование + Стоимость  @КОММЕНТАРИЙ = Сохранение меню в базу |
| @ИМЯ = ПОСТАВКА ПРОДУКТОВ  @ТИП = внешний поток  @ЕДИНИЦА = Текстовая + Числовая  @БНФ = Поставщик + Наименование + Количество  @КОММЕНТАРИЙ = поставка продуктов |
| @ИМЯ = ЗАКАЗ ПРОДУКТОВ  @ТИП = внутренний поток  @ЕДИНИЦА = Текстовая + Числовая  @БНФ = Поставщик + Наименование + Количество  @КОММЕНТАРИЙ = заказ на поставку |
| @ИМЯ = СОХРАНЕНИЕ ИНФОРМАЦИИ О ПРОДУКТАХ  @ТИП = внутренний поток  @ЕДИНИЦА = Текстовая + Числовая  @БНФ = Поставщик + Наименование + Количество  @КОММЕНТАРИЙ = сохранение информации о продуктах в базу |
| @ИМЯ = ИНФОРМАЦИЯ ДЛЯ ПОДПИСАНИЯ ДОГОВОРА  @ТИП = внешний поток  @ЕДИНИЦА = дата + текстовая  @БНФ = Дата договора + Содержание договора  @КОММЕНТАРИЙ = передача договора  @ИМЯ = Информация О ЗАКЛЮЧЕННОМ ДОГОВОРЕ  @ТИП = внутренний поток  @ЕДИНИЦА = дата + текстовая  @БНФ = Дата договора + Реквизиты + Содержание договора  @КОММЕНТАРИЙ = сохранение информации в базу о договоре |
| @ИМЯ = ОТПРАВИТЬ ДОКУМЕНТЫ НА ПОДПИСЬ  @ТИП = внешний поток  @ЕДИНИЦА = Текстовая  @БНФ = Дата отчета + Содержание отчета  @КОММЕНТАРИЙ = отправка документов на подпись руководству |
| @ИМЯ = ЗАПРОС ОТЧЕТА  @ТИП = внешний поток  @ЕДИНИЦА = текстовая  @БНФ = Отчет  @КОММЕНТАРИЙ = руководство запрашивает отчеты |
| @ИМЯ = ОТЧЕТЫ ДЛЯ РУКОВОДСТВА  @ТИП = внешний поток  @ЕДИНИЦА = текстовая  @БНФ = Дата отчета + Содержание отчета  @КОММЕНТАРИЙ = предоставление отчетов руководству |
| @ИМЯ = ИНФОРМАЦИЯ ОБ ОРГАНИЗАЦИИ ФИНАСОВЫХ РАСЧЕТОВ  @ТИП = внешний поток  @ЕДИНИЦА = текстовая  @БНФ = Дата документа + Содержание документа  @КОММЕНТАРИЙ = предоставление информации о финансовых расчетах |
| @ИМЯ = ОПЛАТА  @ТИП = внешний поток  @ЕДИНИЦА = Текстовая + Числовая  @БНФ = Наименование + Цена + Итого  @КОММЕНТАРИЙ = данные об оплате |
| @ИМЯ = ЗАПРОС НА ОПЛАТУ  @ТИП = внешний поток  @ЕДИНИЦА = текстовая + числовая  @БНФ = Наименование + Цена + Итого  @КОММЕНТАРИЙ = запрос на оплату поставщика |
| @ИМЯ = ИНФОРМАЦИЯ ОБ ОПЛАЧЕННЫХ СЧЕТАХ  @ТИП = внешний поток  @ЕДИНИЦА = дата + текстовая  @БНФ = Дата оплаты + Содержание отчета  @КОММЕНТАРИЙ = информация для поставщика о счетах |

**ЛАБОРАТОРНАЯ РАБОТА №13**

**Изучение методов спецификации процессов**

Задание. Подготовить спецификации процессов (основного процесса или группы связанных процессов) проектируемой системы с использованием всех рассмотренных методов. Применить разные методы для описания разных процессов: псевдокод, структурированный естественный язык, визуальный язык проектирования, таблицу решений.

# Спецификация процессов

|  |
| --- |
| * 1. Процесс 1: обслужить клиента |
| 2. Процесс 2: приготовить еду  ***@ВХОД*** = ЗАПРОС НА ВЫПОЛЕНИЕ ЗАКАЗА  ***@СПЕЦПРОЦ***  Выполнить заказ  ***@ВЫХОД =*** ИНФОРМАЦИЯ О ВЫПОЛНЕНИИ ЗАКАЗА  ***@ВХОД*** = ПОСТАВКА ПРОДУКТОВ  ***@СПЕЦПРОЦ***  Обработать информацию о продуктах  ***@ВЫХОД =*** СОХРАНЕНИЕ ИНФОРМАЦИИ О ПРОДУКТАХ  ***@ВХОД*** = ПРОДУКТЫ  ***@СПЕЦПРОЦ***  Обработать информацию о продуктах  ***@ВЫХОД =*** СОХРАНЕНИЕ МЕНЮ |
| 3. Процесс 3: Заказать продукты  ЕСЛИ Заказали продукты ТО ВЫПОЛНИТЬ Заказ продуктов |
| 4. Процесс 4: Заключить договор о сотрудничестве  Построим спецификацию процесса заключения договора о сотрудничестве. При выборе поставщика следует руководствоваться следующими правилами:  ) *если цены поставщика не устраивают, то обговорить скидки;*  2) *если не полный перечень товаров, то обговорить расширение ассортимента;*  3) *если документы не в порядке, то переподготовить документы;*  4) *если все документы в порядке, то приступить к подписанию договора.*     |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | Условия | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | C1 | Цены поставщика устраивают | Д | Д | Д | Д | Н | Н | Н | Н | | С2 | Полный перечень товаров | Д | Д | Н | Н | Д | Д | Н | Н | | С3 | Все документы в порядке | Д | Н | Д | Н | Д | Н | Д | Н | |  | Действия |  |  |  |  |  |  |  |  | | D1 | Обговорить скидки |  |  |  |  | 1 | 1 | 1 | 1 | | D2 | Обговорить расширение ассортимента |  |  | 1 | 1 |  |  | 1 | 1 | | D3 | Переподготовка документов |  | 1 |  | 1 |  | 1 |  | 1 | | D4 | Приступить к подписанию договора | 1 |  | 1 |  | 1 |  | 1 |  | |  |  |  |  |  |  |  |  |  |  | |
| Процесс 5: Провести финансовые расчеты  ***@ВХОД*** = ИНФОРМАЦИЯ О ЗАКЛЮЧЕННОМ ДОГОВОРЕ  ***@СПЕЦПРОЦ***  Обработать информацию  ***@ВЫХОД =*** ОТПРАВИТЬ ДОКУМЕНТЫ НА ПОДПИСЬ  ***@ВХОД*** = ОПЛАТА  ***@СПЕЦПРОЦ***  Оплатить счета  ***@ВЫХОД =*** ЗАПРОС НА ОПЛАТУ  ***@ВХОД*** = ИНФОРМАЦИЯ ОБ ОРГАНИЗАЦИИ ФИНАНСОВЫХ РАСЧЕТОВ  ***@СПЕЦПРОЦ***  Организовать финансовые расчеты  ***@ВЫХОД =*** ИНФОРМАЦИЯ ОБ ОПЛАЧЕННЫХ СЧЕТАХ |
| Процесс 6: Подготовить отчет  ЕСЛИ Запрос отчета ТО ВЫПОЛНИТЬ Отчеты для руководства |

**ЛАБОРАТОРНАЯ РАБОТА №14**

**Применение структурного подхода к проектированию ИС.**

**Построение ER-диаграммы**

**Задание.** Построить модель данных ИС, представленную в виде диаграммы "сущность-связь" (рисунок 14.1)

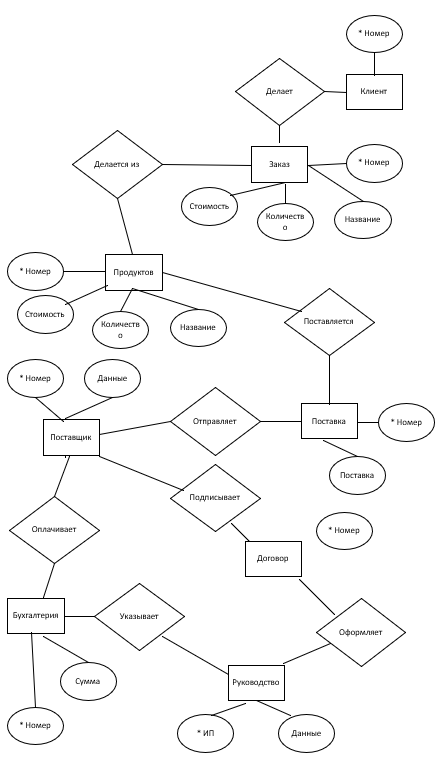


Рисунок 14.1 - Диаграмма сущность-связь (ERD)

Рассмотрим все три нормальные формы на примере Заказа. У Заказа ключом является реквизит Номер (№ заказа), к описательным реквизитам относятся: Наименование, Количество, Стоимость.

Отношение называется нормализованным или приведенным к первой нормальной форме, если все его атрибуты простые (неделимы). Во второй нормальной форме если есть уникальный однозначно атрибут, определяющий каждую запись, и в третьей нормальной форме если отсутствуют функциональные зависимости.

Отношение Заказ= (Номер, Наименование, Количество, Стоимость) находится в первой нормальной форме.

Отношение Заказ= (Номер, Наименование, Количество, Стоимость) находится во второй нормальной форме, так как есть уникальный атрибут номер.

Отношение Заказ= (Номер, Наименование, Количество, Стоимость) находится в третьей нормальной форме, так как возможная транзитивная зависимость с Наименованием устранена путем добавления таблицы Меню.

Ниже от себя представил диаграмму STD (рисунок 14.2)



Рисунок 14.2 - Диаграмма переходов состояний (STD)