Министерство Образования республики Беларусь

Учреждение Образования

«Брестский Государственный технический университет»

Кафедра ИИТ

# **Лабораторная работа №2**

По дисциплине «Проектирование программ в ИС»

# “Создание интерфейса оконного windows-приложения (окна, диалоговые окна). Автоматический каркас приложения”

**Выполнил:**

Студент 2 курса группы ИИ-23

Макаревич Н. Р.

**Проверил**:

Монтик Н.С.

Брест 2024

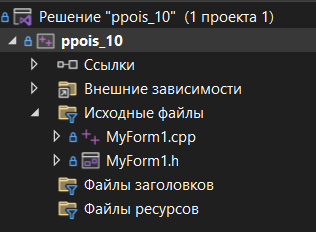
ЦЕЛЬ РАБОТЫ. 1. Изучить создание Windows-приложений с использований каркаса Hello (или аналогичного каркаса). 2. Изучить технологию работы с диалоговыми окнами и элементами управления (ЭУ). Применением диалогового окна в составе главного, в качестве главного окна. 3. Изучить использование кнопок, окон редактирования.

СОСТАВ ОТЧЕТА. 1. Описание каркаса Hello: - состав интерфейса; - файловый состав (структура проекта, папок, состав, назначение файлов и их соподчиненность по включению); - функциональный состав (схема иерархии функций приложения, назначение и прототипы функций). Диаграммы состояний, компонентов. 2. По заданиям п. 5, 6: - диаграммы состояний; листинги. 3. По заданиям п. 8, 11: - диаграммы состояний.

1. Описание каркаса:

Состав интерфейса: Каркас представляет собой простое оконное приложение с базовым интерфейсом, включающим меню и главное окно.

Файловый состав:



Функции обработки меню: функции, вызываемые при выборе определенных пунктов меню.

Функции для работы с ресурсами: функции для загрузки ресурсов приложения, таких как изображения и строки.

Диаграммы состояний и компонентов: Диаграммы состояний и компонентов обычно не используются в контексте каркаса Hello, так как это минимальное оконное приложение без сложной логики и состояний.

2. private: int clickCount = 0;

bool flag9 = false;

private: System::Void MyForm1\_Click(System::Object^ sender, System::EventArgs^ e) {

clickCount++;

if (!flag9) {

Graphics^ graphics = this->CreateGraphics();

int width = this->ClientSize.Width;

int height = this->ClientSize.Height;

SolidBrush^ brush = gcnew SolidBrush(Color::White);

graphics->FillRectangle(brush, 0, 0, width, height);

for each (Control ^ control in this->Controls){

if (dynamic\_cast<Label^>(control) != nullptr){

this->Controls->Remove(control);

}

}

delete brush;

String^ text = clickCount.ToString();

Label^ label = gcnew Label();

label->Text = text;

label->Location = System::Drawing::Point(50, 50);

label->AutoSize = true;

this->Controls->Add(label);

}

}

private: System::Void MyForm1\_MouseClick(System::Object^ sender, System::Windows::Forms::MouseEventArgs^ e) {

if (e->Button == System::Windows::Forms::MouseButtons::Right) {

int mouseX = e->X;

int mouseY = e->Y;

String^ text = mouseX.ToString() + "-" + mouseY.ToString();

Label^ label = gcnew Label();

label->Text = text;

label->Location = System::Drawing::Point(50, 75);

label->AutoSize = true;

this->Controls->Add(label);

String^ text1 = clickCount.ToString();

Label^ label1 = gcnew Label();

label1->Text = text1;

label1->Location = System::Drawing::Point(50, 50);

label1->AutoSize = true;

this->Controls->Add(label1);

}

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

flag9 = true;

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

flag9 = false;

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

array<String^>^ strings = gcnew array<String^> {

"Use",

"function prototype",

"GetTextExtentPoint( )"

};

Graphics^ graphics = this->CreateGraphics();

int x = 100; // начальная координата X

int y = 50; // начальная координата Y

int lineHeight = 20; // высота строки

for each (String ^ str in strings) {

graphics->DrawString(str, this->Font, Brushes::Black, x, y);

y += lineHeight; // переход на следующую строку

}

delete graphics; // освобождаем ресурсы

}

private: System::Void MyForm1\_Load(System::Object^ sender, System::EventArgs^ e) {

}

private:

System::Windows::Forms::Button^ okButton;

System::Windows::Forms::Button^ cancelButton;

System::Windows::Forms::Form^ dialogForm;

void OkButton\_Click(System::Object^ sender, System::EventArgs^ e) {

dialogForm->Close();

}

void CancelButton\_Click(System::Object^ sender, System::EventArgs^ e) {

dialogForm->Close();

}

System::Void MyForm1\_Shown(System::Object^ sender, System::EventArgs^ e) {

dialogForm = gcnew Form();

dialogForm->Text = "Диалоговое окно";

okButton = gcnew System::Windows::Forms::Button();

okButton->Text = "OK";

okButton->Click += gcnew System::EventHandler(this, &MyForm1::OkButton\_Click);

cancelButton = gcnew System::Windows::Forms::Button();

cancelButton->Text = "Cancel";

cancelButton->Click += gcnew System::EventHandler(this, &MyForm1::CancelButton\_Click);

okButton->Location = System::Drawing::Point(20, 20);

cancelButton->Location = System::Drawing::Point(100, 20);

dialogForm->Controls->Add(okButton);

dialogForm->Controls->Add(cancelButton);

dialogForm->Show();

Диаграммы состояний:

3. Диаграммы состояний: