Министерство Образования республики Беларусь

Учреждение Образования

«Брестский Государственный технический университет»

Кафедра ИИТ

# **Лабораторная работа №1**

По дисциплине «Проектирование программ в ИС»

Тема: «Типовой каркас оконного windows-приложения (ТКП). Обработка сообщений. Организация вывода в клиентскую область окна»

**Выполнила:**

Студент группы ИИ-23

Макаревич Н.Р.

**Проверил**:

Монтик Н.С.

Брест 2024

**ЧАСТЬ 1**

ЦЕЛЬ РАБОТЫ: 1. Ознакомиться со структурой и особенностями программирования оконных windows-приложений. 2. Изучить каркасы windows-приложений. 3. Изучить структуру типового каркаса оконных приложений (ТКП), использование ТКП для создания пользовательских приложений.

ХОД РАБОТЫ:

*1. Описание каркасов windows-приложений (Empty, Simple или аналогичных): - характеристика интерфейса, предоставляемых возможностей;*

**Характеристика интерфейса, предоставляемых возможностей:**

**Empty**: Этот каркас обычно не предоставляет никакого интерфейса или функциональности. Он служит основой для создания приложения с нуля.

**Simple**: Этот каркас предоставляет базовый интерфейс с минимальной функциональностью. Он может включать в себя окно приложения и основные элементы управления.

*- файловый состав (структура проекта, дерево папок, состав, назначение файлов и их соподчиненность по включению);*

**Файловый состав (структура проекта, дерево папок, состав, назначение файлов и их соподчиненность по включению):**

Оба каркаса имеют структуру проекта, которая включает в себя файлы исходного кода, ресурсы и конфигурационные файлы. Важные файлы включают main (или аналогичный файл), который содержит основной код приложения, и project.sln (или аналогичный файл), который содержит информацию о конфигурации проекта.

*- функциональный состав (схема иерархии функций приложения, назначение и прототипы функций).*

**Функциональный состав (схема иерархии функций приложения, назначение и прототипы функций):**

В каркасе Empty обычно нет функций, так как он служит основой для создания приложения с нуля.

В каркасе Simple могут быть базовые функции, такие как:

Init(), Update(), и Render(), которые служат для инициализации, обновления и отрисовки приложения соответственно.

*2. Аналогичное описание ТКП. Диаграммы прецедентов, состояний, компонентов.*

место для диаграмм

*3. Выводы:*

*а) о последовательности инициализации ТКП;*

Последовательность инициализации типового каркаса приложения может быть описана следующим образом:

1. WinMain: Это основная точка входа в приложение. Здесь происходит регистрация класса окна и создание главного окна приложения.
2. WM\_CREATE: Это сообщение обрабатывается при создании окна. Здесь создаются все необходимые элементы управления, такие как кнопки.
3. WM\_COMMAND: Это сообщение обрабатывается при взаимодействии пользователя с элементами управления. Здесь определяется поведение приложения в ответ на действия пользователя, такие как нажатие кнопок.
4. WM\_PAINT: Это сообщение обрабатывается при необходимости перерисовки окна. Здесь определяется, как будет выглядеть содержимое окна.
5. WM\_SIZE: Это сообщение обрабатывается при изменении размера окна. Здесь определяется, как будут изменяться размеры и положение элементов управления в ответ на изменение размера окна.
6. WM\_CLOSE и WM\_DESTROY: Эти сообщения обрабатываются при закрытии окна и завершении приложения. Здесь освобождаются все ресурсы, занятые приложением.

*б) о причинах и условиях посылки сообщений ТКП, рассмотренных в работе;*

1. WM\_CREATE: [причина]: создании окна.
2. WM\_COMMAND: [причина]: взаимодействие пользователя с элементами управления, такими как кнопки.
3. WM\_PAINT: [причина]: окно нуждается в перерисовке.
4. WM\_SIZE: [причина]: изменение размера окна.
5. WM\_CLOSE и WM\_DESTROY: [причина]: закрытие окна и завершение приложения.

*в) перечень ситуаций “перерисовки”;*

Перерисовка была необходима при изменении размеров окна(растягивании или нажатии на кнопку “развернуть”. Сделать это получилось не сразу( т.к. размеры окна корректно изменялись только при нажатии “Clear Screen”.

*г) выводы по способу реализации завершения работы приложения.*

Завершение работы реализовано с помощью стандартных функций WM\_CLOSE и WM\_DESTROY.

**ЧАСТЬ 2**

ЦЕЛЬ РАБОТЫ: 1. Ознакомиться с особенностями организации вывода в оконных windows-приложениях. 2. Ознакомиться с особенностями управления сообщениями

ХОД РАБОТЫ:

*1.Описание приложений :- состав интерфейса и его возможности; .*

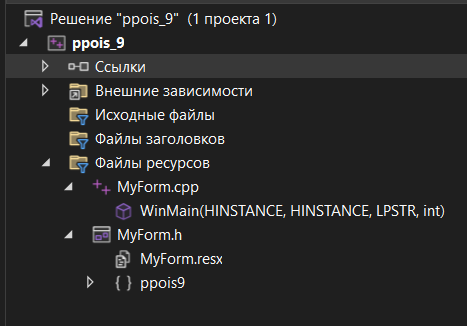
Интерфейс:

1. Окно приложения: Это основное окно, в котором отображается весь контент и элементы управления.
2. Кнопка “Numbers”: При нажатии на эту кнопку на экране отображаются числа и их квадраты.
3. Кнопка “Draw”: При нажатии на эту кнопку на экране отображается графика.
4. Кнопка “Clear”: При нажатии на эту кнопку содержимое экрана очищается.

Возможности:

1. Отображение чисел: Приложение может отображать числа и их квадраты на экране при нажатии на кнопку “Numbers”.
2. Отображение графики: Приложение может отображать графику на экране при нажатии на кнопку “Draw”.
3. Очистка экрана: Приложение может очищать экран при нажатии на кнопку “Clear”.
4. Вывод координат курсора при нажатии правой кнопки мыши.
5. Изменение размера окна: Приложение корректно обрабатывает изменение размера окна, перемещая и изменяя размер кнопок в соответствии с новыми размерами окна.
6. Завершение работы: Приложение корректно завершает работу при закрытии окна, освобождая все занятые ресурсы.

*2.файловый состав (структура проекта, дерево папок, состав, назначение файлов и их соподчиненность по включению);*



*3.функциональный состав (привести схему иерархии функций приложения, описать назначение и прототипы функций).*

В этой иерархии:

* WinMain - это основная точка входа в приложение.
* InitializeComponent используются для создания окна приложения.
* button1\_Click используется для обработки нажатия кнопки «Clean»
* button2\_Click используется для обработки нажатия кнопки «Draw»
* button3\_Click используется для обработки нажатия кнопки «Clean»
* MyForm\_MouseClick используется для обработки нажатия правой кнопки мыши.