Лекция №13 по дисциплине «Объектно-ориентированное программирование» на тему:

Работа с графическим интерфейсом

Графический интерфейс пользователя

Графический интерфейс пользователя (GUI)— разновидность пользовательского интерфейса, в котором элементы интерфейса исполнены в виде графических изображений.

- В GUI пользователь имеет произвольный доступ ко всем элементам интерфейса и осуществляет непосредственное манипулирование ими.
- Элементы интерфейса в GUI соответствуют их назначению и свойствам, что облегчает понимание и освоение программ неподготовленными пользователями.

Виды графического интерфейса пользователя

- простой: типовые экранные формы и стандартные элементы интерфейса, обеспечиваемые самой подсистемой GUI;

- истинно-графический, двумерный: нестандартные элементы интерфейса, реализованные собственными средствами приложения или сторонней библиотекой;

- трёхмерный: например, CompizFusion для Linux). Для его создания в C++ можно использовать библиотеку GLUT, а для более серьезных целей другие возможности OpenGL.

Разработка графического интерфейса пользователя

- Для облегчения создания GUI было создано большое количество библиотек: в частности: VCL для Builder C++, MFC и его облегченный вариант WTL для Visual C++.

- API Windows Forms. API упрощает доступ к элементам интерфейса Microsoft Windows.

- Управляемый код (классы, реализующие API для Windows Forms) не зависит от языка разработки. Одинаково можно использовать Windows Forms как при написании ПО на C#, C++, так и на VB.Net и др.

Каркас приложения

- Библиотека MFC облегчает работу с GUI путем создания каркаса приложения — «скелетной» программы, автоматически создаваемой по заданному макету интерфейса и берущей на себя действия по его обслуживанию.

- Программисту после генерации каркаса приложения необходимо только вписать код в места, где требуются специальные действия. Каркас должен иметь определенную структуру, поэтому для его генерации и изменения в Visual C++ предусмотрены мастера.

Добавление кода приложения

- Добавление кода приложения к каркасу реализовано двумя способами.

- Первый использует механизм наследования: основные программные структуры каркаса представлены в виде классов, наследуемых от библиотечных.

- В этих классах предусмотрено множество виртуальных функций, вызываемых в определенные моменты работы программы.

Добавление обработчиков оконных событий

- Второй способ используется для добавления обработчиков оконных событий.

- Мастер создает внутри каркасов классов, связанных с окнами, специальные массивы — карты оконных сообщений, содержащие пары «ИД сообщения — указатель на обработчик».

- При добавлении/удалении обработчика мастер вносит изменения в соответствующую карту сообщений.

Создание простейшей программы

- Для создания программы используется Visual Studio и мастер MFC AppWizard, который сгенерирует основную часть кода. Потом код нужно отредактировать.

- Процесс работы с MFC Wizard состоит из нескольких этапов.

- На первом этапе выбирается тип приложения «один документ» – это режим SDI (Single Document Interface), для того, чтобы программа работала в одном окне. Стиль проекта - стандарт MFC.

Второй этап

- На втором этапе нужно убрать поддержку составных документов, баз данных.

- Программа с именем Lab5Visual состоит из 4 классов: CLab5VisualApp, CLab5VisualView, CLab5VisualDoc, CMainlFrame, которые и составляют ее ядро, и дополнительных файлов: заголовочные файлы, PCH файлы, каталог ресурсов и др.

- Более подробное их описание находится в текстовом файле ReadMe, который создается вместе с проектом.

Структура графической программы

- Графическая программа состоит из четырёх основных частей.

- 1. Объект приложения- Windows запускает его при старте программы. Когда этот объект начинает работу, он размещает на экране главное окно. Он содержится в файлах Lab5Visual.cpp и Lab5Visual.h

- 2. Объект главного окна – в нём находится меню, заголовок окна и панель инструментов. Рабочая область программы называется клиентской областью окна.

Структура графической программы

- 3. Объект вида предназначен для работы с клиентской областью. Объект вида – это окно, которое накладывается поверх клиентской области.
- 4. Объект документа - хранит данные программы.
- Visual C++ облегчает задачу разработки интерфейса и позволяет сохранить все данные в объекте документа, а затем поручить объекту вида отобразить лишь те данные, которые попадают в клиентскую область объекта вида.

Разработка простейшей программы

Изменим класс Lab5VisualView , который предназначен для отображения данных. Этот класс содержит несколько методов для печати данных.

Воспользуемся методом OnDraw(). Этому методу передаётся указатель pDC, ссылающийся на контекст устройства.

Контекст устройства- это область памяти, используемая в Windows для выполнения графических операций. Для того чтобы вывести строку на экран, используется объект класса CString.

Также используется метод TextOut класса CDC. Объекты этого класса содержат встроенные методы, используемые в процессе рисования в контексте устройства.

Текст простейшей программы

```
void CLab5VisualView::OnDraw(CDC*pDC)
{
    CString my_cstring;
    my_cstring ="Привет всем !";
    pDC->TextOut(0,0,my_cstring);
    CLab5VisualDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
       return;
}
```

Объект приложения

Объект приложениявыполняет три задачи:

запускает программу, используя функцию WinMain();

создаёт главное окно;

организовывает передачу сообщений Windows в главное окно и из него.

Сообщения - это сигналы с минимальным объёмом служебных данных, посредством которых различные объекты в среде Windows общаются между собой.