**Шаг 17.  
Среда программирования Visual C++.  
Основные принципы создания Windows-приложения**

    На этом шаге мы рассмотрим ***общую структуру* Windows-*приложения***.

    Опишем основные элементы архитектуры 32-разрядных **ОС Windows**.

**Процессы и потоки**

**Windows**-приложения состоят как минимум из одного процесса. ***Процесс*** - это экземпляр выполняющейся программы. Ему выделяются адресное пространство и ресурсы. В рамках одного процесса может быть запущено несколько потоков.

***Поток*** - это минимальная программная единица, для выполнения которой планировщик выделяет процессорное время. Поток запускается в адресном пространстве процесса и использует его ресурсы.

    Процесс имеет как минимум один исполняемый поток, называемый ***первичным потоком***. Для выполнения фоновых задач можно создавать вторичные потоки, используя таким образом преимущества многозадачности. Приложения, применяющие несколько потоков, называются ***многопоточными***.

**Запуск приложения**

    При запуске приложения операционная система создает процесс и начинает выполнение его первичного потока. Когда поток заканчиает свою работу, заканчивается и процесс. Информация о первичном потоке передается операционной системе в виде адреса функции. Поэтому все **Windows**-приложения содержат вызываемую при запуске функцию **WinMain()**, адрес которой и передается в качестве адреса первичного потока.

    Затем приложение создает окно. Но до его отображения на экране в операционной системе должны быть зарегистрированы ***оконные классы*** - шаблоны, содержащие информацию о свойствах окна. В процессе регистрации окно связывается с оконной процедурой, которая позволяет задать отображаемую этим окном информацию, а также его реакцию на действия пользователя посредством ***обработки сообщений*** системы.

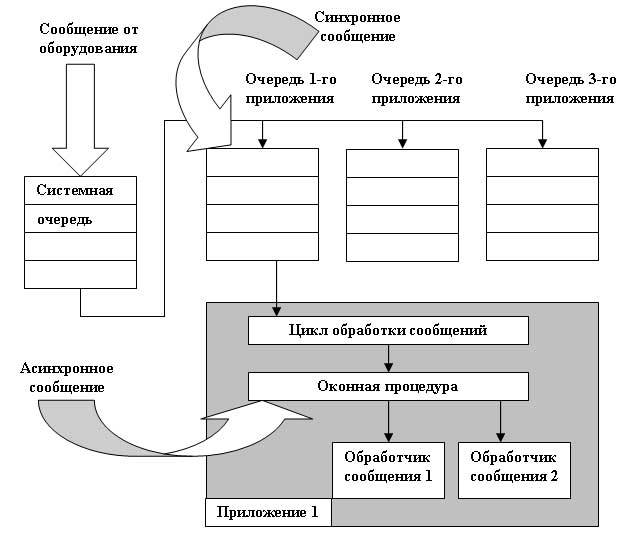
**Сообщения Windows**

    Окно служит для взаимодействия пользователя с приложением, а для взаимодействия операционной системы, приложения и его компонентов предназначены ***системные сообщения***. Например, при создании экземпляра приложения операционная система посылает ему серию сообщений, отвечающих за его инициализацию. Клавиатура и мышь тоже генерируют сообщения и отправляют их в соответствующее приложение. Таким образом, основная задача программы - ***обработать получаемые сообщения***, то есть передать их ожидающему окну и выполнить некоторые действия, зависящие от типа сообщения и его параметров. Разработчик должен связать сообщения с функциями, которые будут запускаться на них в ответ.

**Обработка сообщений приложением**

    Каждый поток, создающий окно, имеет ***очередь сообщений*** - структуру данных, в которой операционная система хранит сообщения. У всех **Windows**-приложений есть главное окно, имеющее в свою очередь ***цикл обработки сообщений*** - часть программы, извлекающую сообщения из очереди и посылающую их в соответствующую оконную процедуру. Эта процедура обрабатывает сообщение или передает его стандартной оконной процедуре, содержащей стандартный обработчик сообщений. Например, сообщение о свертывании окна будет обработано почти всеми приложениями одинаково, так как для этого подходит стандартная оконная процедура.

    На рисунке 1 изображено, как сообщения ставятся в очередь и обрабатываются приложением. Обратите внимание на то, как средствами функций **PostMessage()** и **SendMessage()** можно послать сообщение. Обратившись к этим функциям или их двойникам из библиотеки **MFC** - **CWnd::PostMessage()** и **CWnd::SendMessage()**, - приложение посылает или принимает сообщения.

  
Рис.1.Обработка сообщений

    Функция **PostMessage()** помещает сообщение в очередь, связанную с окном, и сразу же завершается, не дожидаясь его обработки (***асинхронное сообщение***). А функция **SendMessage()** после помещения сообщения в очередь ждет его обработки (***синхронное сообщение***).

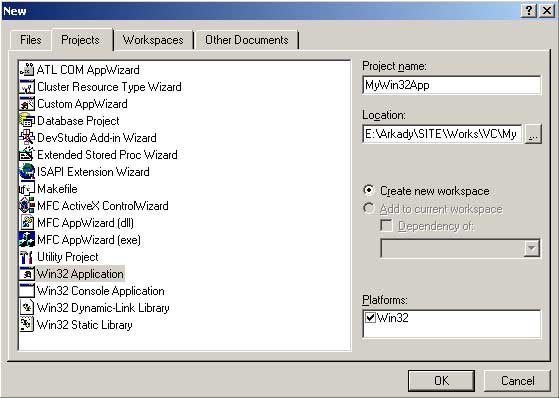
**Основные этапы построения Windows-приложения**

    Для создания **Win32**-приложения необходимо:

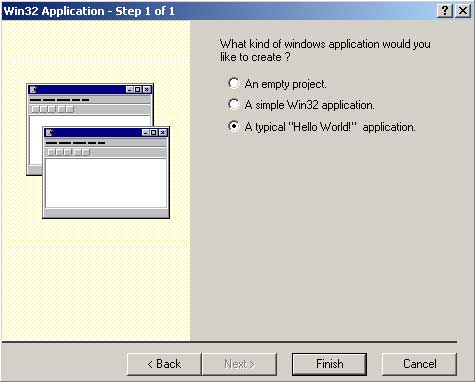
* создать функцию **WinMain()**, которая является точкой входа в программу;
* зарегистрировать оконные классы и объявить связанные с ними оконные процедуры;
* создать экземпляр главного окна приложения;
* создать цикл обработки сообщений, передающий их в соответствующую оконную процедуру;
* реализовать оконную процедуру, обрабатывающую сообщения.

    Создадим простое **Windows**-приложение без использования **MFC**.

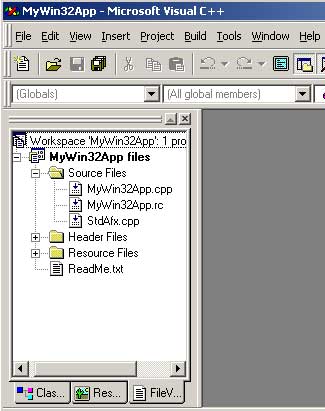
* Создание **Win32**-приложения.
  1. Запустите **Visual C++**.
  2. В меню **File** щелкните пункт **New**.
  3. Выберите из списка **Win32 Application**.
  4. В качестве названия проекта введите **MyWin32App**.

  
Рис.2. Выбор типа проекта

* 1. Щелкните кнопку **ОК**. Появится мастер **Win32 AppWizard**. Выберите **A typical "Hello World" application** и щелкните кнопку **Finish**.

  
Рис.3. Первый шаг ***Мастера***

* 1. Чтобы создать проект, в диалоговом окне **New Project Information** щелкните кнопку **ОК**.
  2. Когда проект будет готов, перейдите на вкладку **FileView** окна **Workspace**. Для просмотра файлов проекта **MyWin32App** щелкните знак "+". Таким же образом откройте узел **Source Files**.

  
Рис.4. Перечень исходных файлов

* 1. Дважды щелкните значок файла **MyWin32App.cpp**, чтобы просмотреть сгенерированный исходный код.

http://it.kgsu.ru/images/vopr.jpgНайдите в файле **MyWin32App.cpp**:

* функцию **WinMain()**, в которой происходит инициализация приложения и реализуется цикл обработки сообщений;
* функцию **MyRegisterClass()**, которая регистрирует оконный класс главного окна приложения;
* оконную процедуру **WndProc()**, связанную с главным окном в функции **MyRegisterClass()**. Эта процедура обрабатывает сообщения **WM\_COMMAND** (которые посылаются пунктами меню, элементами управления и "быстрыми" клавишами), **WM\_PAINT** (посылается при запросе на перерисовку некоторой области окна) и **WM\_DESTROY** (посылается при уничтожении окна). Остальные сообщения передаются в стандартную оконную процедуру;
* функцию **InitInstance()**, создающую и отображающую экземпляр главного окна приложения.

    Со следующего шага мы начнем рассматривать ***структуру* MFC-*приложения***.

[Предыдущий шаг](http://it.kgsu.ru/VC/vc_0016.html) http://it.kgsu.ru/images/redball.gif[Содержание](http://it.kgsu.ru/VC/oglav.html) http://it.kgsu.ru/images/redball.gif[Следующий шаг](http://it.kgsu.ru/VC/vc_0018.html)