

**СОЗДАНИЕ
GUI-ПРИЛОЖЕНИЙ
(Graphics User Interface)**

Создание GUI-приложений

GUI-приложение – это такой тип приложения, которое создается и выполняется на основе графического интерфейса.

Категории сообщений

1. Сообщения-запросы (используются для изменения или получения значений свойств адресата);
2. Сообщения-уведомления (используются для сообщения своих свойств).

Создание GUI-приложений

Каждому потоку, создавшему хотя бы одно окно, ОС выделяет свою очередь сообщений, используя структуру **THREADINFO**.

Создание GUI-приложений

Структура **THREADINFO** содержит:

- указатель на очередь асинхронных сообщений;
- указатель на очередь синхронных сообщений;
- указатель на очередь виртуального ввода;
- указатель на очередь ответных сообщений;
- флаги пробуждения;
- переменные, отражающие локальное состояние ввода;
- переменная nExitCode.

Создание GUI-приложений

Посылка асинхронных сообщений:

```
BOOL PostMessage (  
    HWND hwnd,          // дескриптор окна, которому  
                        // посылается сообщение  
    UINT  uMsg,         // код сообщения  
    WPARAM wParam,     // параметры сообщения  
    LPARAM lParam);    // параметры сообщения
```

Флаг пробуждения: QS_POSTMESSAGE

```
void PostQuitMessage (int nExitCode);
```

Флаг пробуждения: QS_QUIT

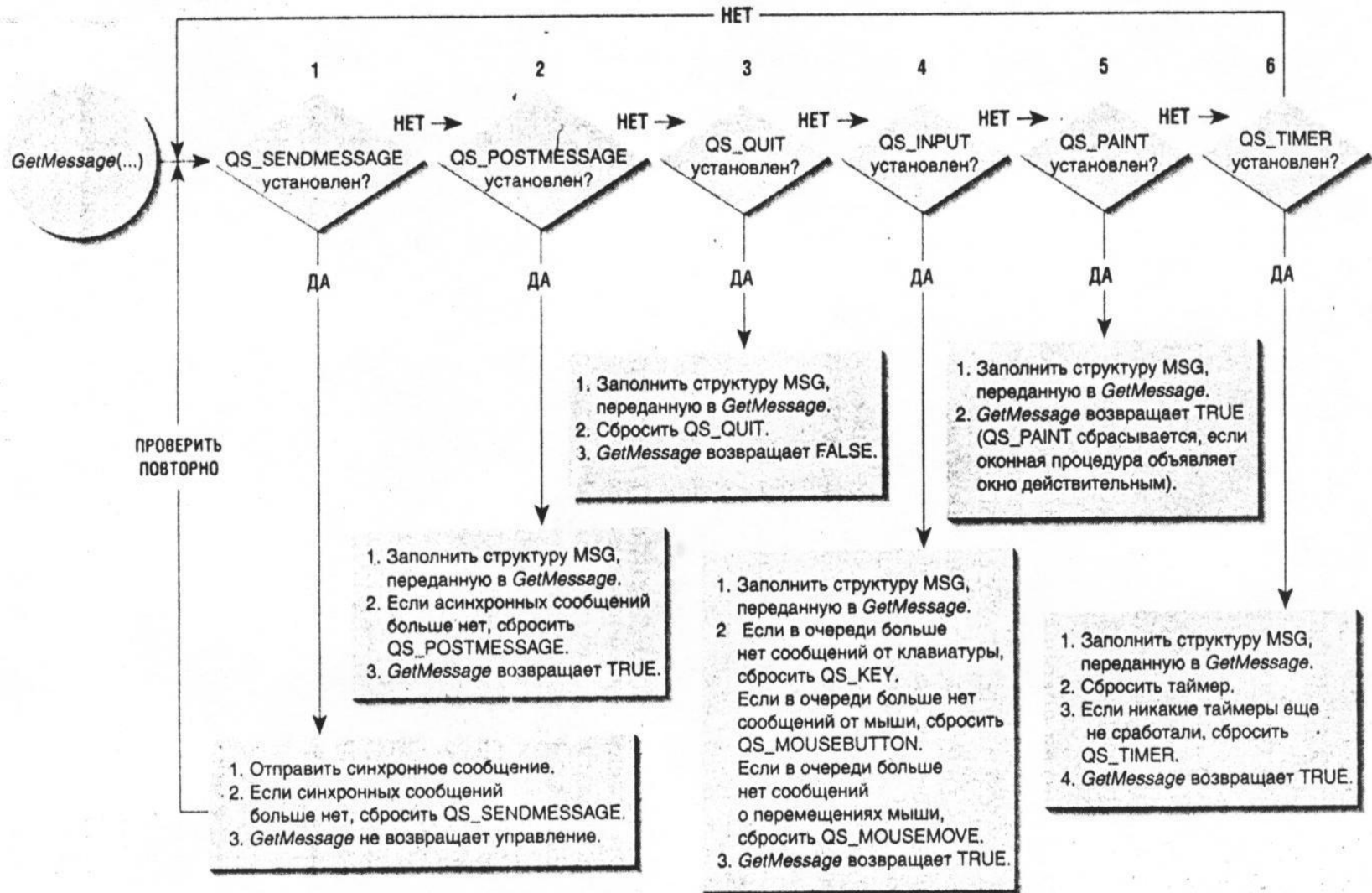
Создание GUI-приложений

Посылка синхронных сообщений:

```
LRESULT SendMessage (  
    HWND      hwnd,    // дескриптор окна, которому  
                    // посылается сообщение  
    UINT      uMsg,    // код сообщения  
    WPARAM    wParam, // параметры сообщения  
    LPARAM    lParam); //параметры сообщения
```

Флаг пробуждения: QS_SENDMESSAGE

Создание GUI-приложений



Режим функционирования окна

1. Приоритетное окно, foreground window
(созданное потоком, с которым в текущий момент времени работает пользователь);
2. Фоновое окно, background window
(неприоритетное окно, например рабочий стол).

Создание GUI-приложений

Расположение окон относительно друг друга

1. Окно-владелец всегда находится ниже (под) окнами, которыми владеет.
2. При уничтожении окна-владельца система автоматически уничтожает все окна, которыми владеет.
3. При сворачивании окна-владельца сворачиваются все окна, которыми он владеет.

Создание GUI-приложений

Области окна

1. Служебная область (область окна с заголовком, меню, рамками и т.д.).
2. Клиентская область (область окна для вывода изображений).

Создание GUI-приложений

Типы окон

1. Перекрывающиеся окна (overlapped window);
2. Всплывающие окна (pop-up window);
3. Дочерние окна (child window);
4. Слоистые окна (layered window);
5. Окна для сообщений (message-only window).

Создание GUI-приложений

Состояния окон

1. Скрытое окно
2. Свернутое окно
3. Развернутое окно
4. Активное окно
5. Заблокированное окно

Создание GUI-приложений

Алгоритм создания простого GUI-приложения

1. Описание функции WinMain(), которая содержит:

- описание класса окна приложения;
- регистрацию описанного класса;
- создание главного окна приложения;
- отображение главного окна приложения;
- организацию цикла обработки сообщений для окна.

2. Описание оконной процедуры, которая содержит обработку получаемых сообщений.

Создание GUI-приложений

Точка входа в программу:

```
int WINAPI WinMain(  
    HINSTANCE hIns,  
    HINSTANCE hPrevIns,  
    LPSTR     lpszCmpLine,  
    int      nCmdShow);
```

Создание GUI-приложений

Структура для описания класса окна:

```
typedef struct {  
    UINT        style;  
    WNDPROC     lpfnWndProc;  
    int         cbClsExtra;  
    int         cbWndExtra;  
    HANDLE      hInstance;  
    HICON       hIcon;  
    HCURSOR     hCursor;  
    HBRUSH      hbrbackground;  
    LPCTSTR     lpszMenuName;  
    LPCTSTR     lpszClassName;  
} WNDCLASS;
```