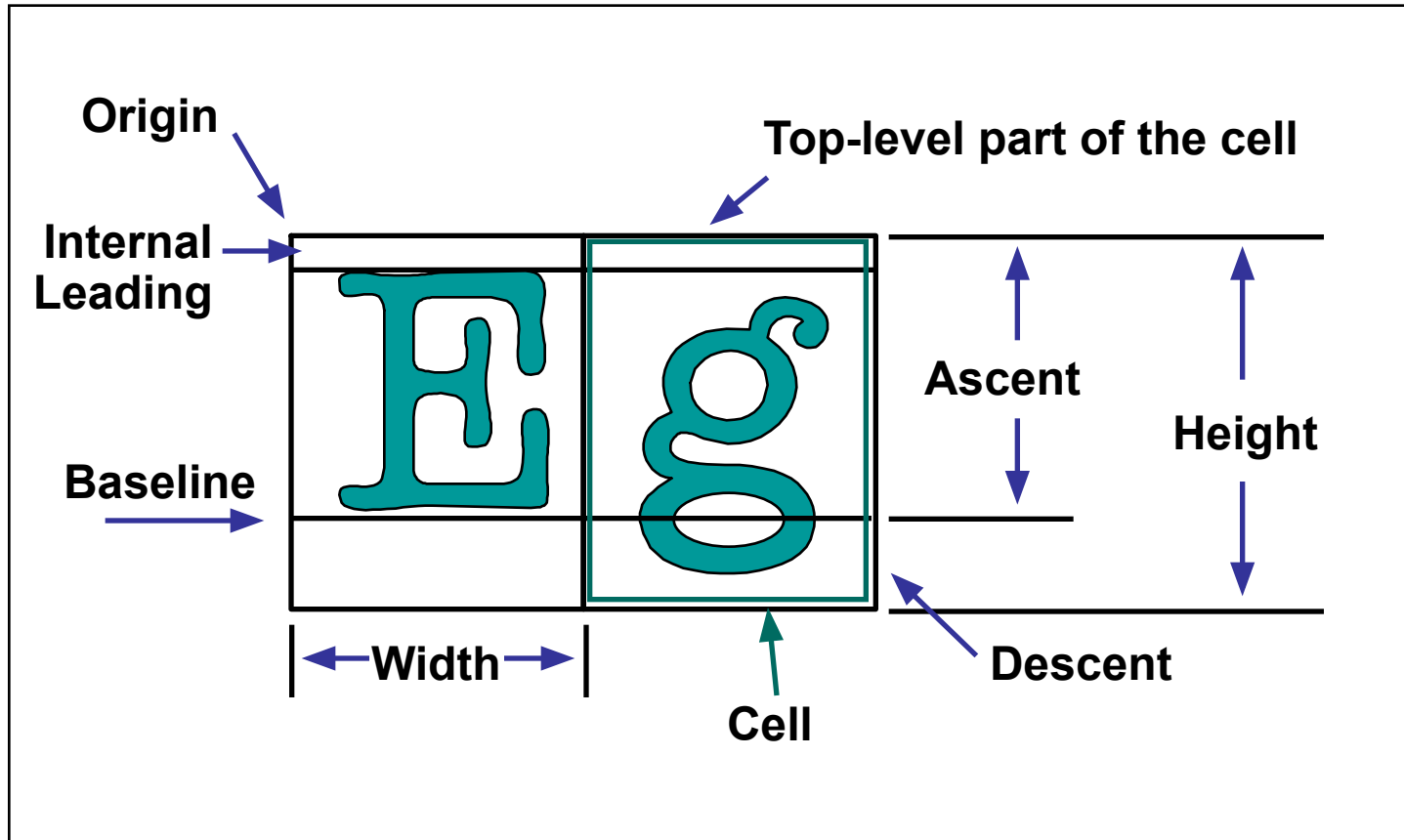


# **ГРАФИЧЕСКИЙ ИНТЕРФЕЙС УСТРОЙСТВА**

Дополнение к лекции СРС-1

# О шрифтах



# Описание шрифтов

- **TEXTMETRIC** - стандартный тип данных Windows для описания шрифтов
  - Это структура C: содержит 20 полей для описания шрифта
- **Логические шрифты:** Описывают “идеальные” шрифты
  - Могут быть недоступны в системе
  - Представлены типом данных **LOGFONT**
- *Физические шрифты:* Инсталлированы и доступны в системе
  - **TEXTMETRIC** или **OUTLINETEXTMETRIC**

# Класс CFont: создание и использование шрифтов

- Создать объект CFont (логическая структура шрифта) с набором желаемых атрибутов.
- Выбрать шрифт для контекста устройства
  - **CFont::CreateFont**
  - *Сопоставитель шрифтов Windows GDI* выбирает наиболее похожий на идеальный шрифт среди доступных
  - Процесс определения шрифта - *realization*

# Функции для вывода текста класса CDC

- Атрибуты вывода текста
  - CDC::SetTextColor, CDC::GetTextColor
  - CDC::SetBkMode, CDC::GetBkMode
  - CDC::SetBkColor, CDC::GetBkColor
  - CDC::SetTextAlign, CDC::GetTextAlign
- Вывод текста
  - CDC::TextOut
  - CDC::TabbedTextOut
  - CDC::DrawText

# Печать средствами MFC

- Программирование
  - Сообщить количество страниц
  - Указать параметры каждой страницы
  - Занимать/освободить ресурсы системы
- Каркас приложения
  - Отображение ОД Печать
  - Создание объекта класса CDC
  - Сообщение драйверу принтера о начале/завершении печати и начале/конце страницы, текущей печатаемой страницы
  - Вызов функций класса CView

# Последовательность вызова функций при печати

- **CView::OnPreparePrinting – Од Печать, кол-во страниц, номера страниц**
- **CView::OnBeginPrinting – шрифты, ширина/кол-во строк**
- **CView::OnPrepareDC – координаты лев.верхнего угла, режим сопоставления для каждой страницы документа**
- **CView::OnPrint – колонтитулы, печать и предварительный просмотр для каждой страницы**
- **CView::OnEndPrinting – освобождение ресурсов**

Графическая библиотека GDI+  
Определяет внешний графический  
облик ОС, начиная с Windows XP и  
Windows Server 2003



# Преимущества от GDI

## Реализация:

- Прозрачное управление памятью (собственный менеджер в куче)
- Перегрузка имен функций (функции одного назначения отличаются только по своим параметрам)
- Собственное пространство имен (Gdiplus)
- Перегрузка операторов

## Архитектурные новинки:

- Аппаратная абстракция (перенос на 64 битную платформу)
- Новый дизайн графических функций/объектов
- Разделение функций закрашки и отрисовки
- Увеличившаяся поддержка траекторий (paths) и их взаимодействия с регионами

# Новые технологии и возможности:

- Градиентная закрашка
- Поддержка прозрачности
- Режимы улучшения изображения
- Сплаины (лекало) метод DrawCurve
- Траектории
- Координатные преобразования (объект Matrix позволяет поворот, перенос, масштабирование  
Регионы не привязаны к координатным устройствам
- Работа с растрами
- Поддержка всевозможных графических форматов: BMP, GIF, TIF, JPEG, PNG, ICO, WMF, EMF

Ключевым классом является `Graphics`, который содержит 200 методов, отвечающих за рисование, отсечение и параметры устройства вывода. То есть выполняет аналогичную функцию контекста устройств в GDI.

Отличие заключается в изменении программной модели: вызываются методы класса.

**OpenGL** это оптимизированная, высокопроизводительная графическая библиотека функций и типов данных для отображения двух-и трехмерной графики.

Стандарт OpenGL был утвержден в 1992г. Он основан на библиотеке IRIS GL, разработанной компанией Silicon Graphics ([www.sgi.com](http://www.sgi.com)).

OpenGL поддерживают все платформы. Кроме того, OpenGL поддерживана аппаратно. Существуют видеокарты с акселераторами и специализированные SD-карты, которые выполняют примитивы OpenGL на аппаратном уровне.

Около двадцати Windows GDI-функций создано специально для работы с OpenGL.

Большая часть из них имеет префикс wgl (аббревиатура от Windows GL).

Эти функции являются аналогами функций с префиксом glx, которые подключают OpenGL к платформе Xwindow System.

Существует несколько Win32-функций для управления форматом пикселей и двойной буферизацией.

Они применимы только для специализированных окон OpenGL.

Другими разновидностями графического интерфейса являются WinG и DirectDraw – используемый для игровых и мультимедийных разработок.

WinG позволяет программам копировать графические массивы из памяти напрямую в память графического адаптера; поддерживается классами OWL.

Для увеличения производительности игровых разработок был выпущен , DirectDraw с прямым доступом к видеопамяти без необходимости копирования; этот интерфейс не поддерживается классами OWL.