

1. Описать основное назначение библиотеки GDI.
Проанализировать принципы работы этой библиотеки.
Сделать сравнительный анализ библиотек GDI и GDI+.

- **Что такое GDI+ и за что оно отвечает?**

GDI+ является частью операционной системы Windows XP, с помощью него можно разрабатывать Windows и Web приложения, которые позволяют работать с векторной и растровой графикой, которые будут взаимодействовать с графическими устройствами, такими как монитор компьютера, принтер или другие устройства отображения. В результате мы получаем, например, картинку на мониторе. Этим посредником и является библиотека GDI+.

GDI+ не взаимодействует с устройствами отображения напрямую, а использует для этого драйвер устройства. Примеры GDI+: вывод простого текста, рисование линии или прямоугольника, печать.

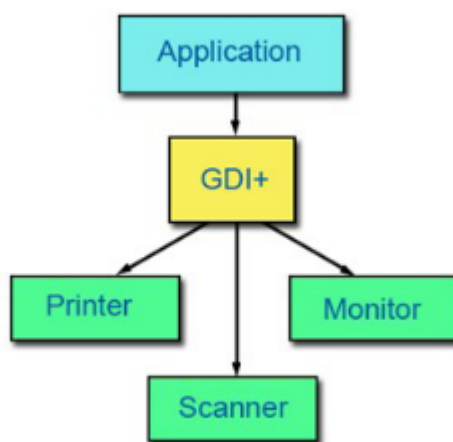


Схема взаимодействия

Рассмотрим работу GDI+ на примере рисования линии: предположим, что приложение рисует линию. Линия будет передана как набор последовательных пикселей, у которой имеется начальная и конечная точка. Рисуя эту линию, монитору необходимо знать, в каком месте их(эти точки) рисовать. Для того, чтобы приказать монитору нарисовать это пиксели, в этом приложении будет использоваться метод DrawLine() библиотеки GDI+. GDI+ дает инструкции определённой системе отобразить линию в виде последовательностей пикселей.

- **Анализ принципов работы GDI**

Для того, чтобы получить какое-либо изображение(рисунок) на экране, монитора, необходима видеокарта. Компьютер дает для нее специфические команды, а та в свою очередь заставляет монитор отображать то, что нужно пользователю. У разных видеокарт разных производителей есть свой собственный специфический набор команд и возможностей. GDI позволяет пользователям абстрагироваться от этих ограничений и особенностей, скрывая разницу между этими устройствами. Пользователю нет необходимости писать код для определенного драйвера, так как GDI решает эти задачи самостоятельно. Пользователю нужно всего лишь вызвать соответствующие методы, чтобы вывести изображение на принтер или монитор.

- **Сравнение GDI и GDI+**

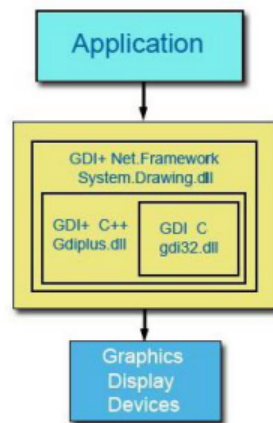
GDI — это библиотека Gdi32.dll, она использовалась в ранних версиях Windows и базируется на старом Win32API с функциями языка C. Мы можем использовать ее функциональность в управляемом коде .NET. Чтобы применить библиотеку GDI в нашем приложении, мы должны импортировать ее с помощью типа DLL Import Attribute:

```
[System.Runtime.InteropServices.DllImportAttribute("gdi32.dll")]
```

После этого мы сможем использовать функциональность gdi32.dll библиотеки в нашем Net приложении. Вряд ли мы будем использовать ее в наших приложениях, ведь библиотека GDI+ на управляемом языке решает многие задачи.

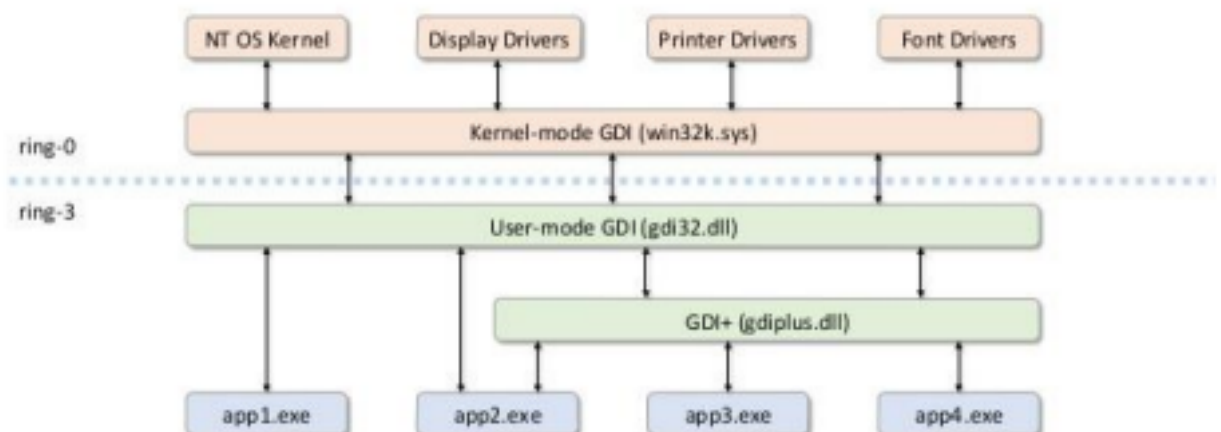
GDI+ является компонентом операционных систем Windows XP и Windows Server2003. Это оболочка вокруг старой библиотеки GDI, она написана на языке C++ и представляет улучшенную производительность и более интуитивно понятную модель программирования, представляясь библиотекой Gdiplus.dll. Может применяться как в управляемом Net коде (заданной сборкой System.Drawing.dll), так и в неуправляемом.

Фактически библиотека NET.Framework GDI+ также является оболочкой вокруг GDI+ языка C++. Представляет более продвинутый API, включая автоматическое управление памятью, межъязыковую интеграцию, улучшенную безопасность, отладку, развертывание и многое другое.



Взаимоотношения библиотек GDI и GDI+

Windows GDI – simplified architecture



1. Описать основное назначение контекста устройства.

Контекст Устройства (DC) представляет объект Windows, который содержит набор графических объектов, информацию об их атрибутах рисования, а также определяет графические режимы устройства

отображения.

Прежде чем что-либо отобразить, например, на мониторе, приложению необходимо получить этот контекст до того, как посылать выходной поток устройству. В Net.Framework это решается с помощью класса System. Drawing.Graphics, в который и помещен контекст устройства DC.

Благодаря взаимодействию DC наше приложение может быть оптимизировано, например, при прорисовке на экране конкретной области. С помощью контекста устройства мы можем определить в каких координатах и как отображать то, что нам необходимо.

Графические объекты представляются такими классами как Pen – для рисования линий, Brush – для рисования и заполнения форм, Font – для отображения текста, Image и другие.



3. Описать функциональность библиотеки .NET Core, определенной в пространстве имен System.Drawing.

Базовая функциональность управляемого GDI+ представлена библиотекой NET.Framework, определенной в пространстве имен System.Drawing. В этом пространстве можно найти классы, представляющие изображения, кисти, перья, шрифты и другие типы, позволяющие работать с графикой. Дополнительная функциональность

обеспечиваются подпространствами: `System.Drawing.Desing`, `System.Drawing.Drawing2D`, `System.Drawing.Imaging`, `System.Drawing.Printing`, `System.Drawing.Text`. Следующая таблица представляет краткое описание этих пространств имен.

Пространство имен	Описание
<code>System.Drawing</code>	Базовое пространство имен GDI+ определяет множество типов для основных операций визуализации, например <code>Graphics</code> определяет методы и свойства рисования на устройствах отображения, типы <code>Point</code> и <code>Rectangle</code> например инкапсулируют примитивы GDI+, класс <code>Pen</code> используется при рисовании линий и кривых, классы производные от абстрактного типа <code>Brush</code> используются для заполнения внутренних областей графических форм таких как прямоугольники и эллипсы. Не поддерживается в Windows и ASP.NET сервисах.
<code>System.Drawing.Desing</code>	Пространство имен содержащее типы, обеспечивающие базовую функциональность для разработки расширений пользовательского интерфейса времени выполнения и их размещение

	в панели инструментов <code>ToolBox</code> , также включает предопределенные диалоговые окна например: <code>FontEditor</code> представляет редактор выбора и конфигурирования объектом <code>Font</code> , <code>ColorEditor</code> представляет редактор для визуального выбора цвета, Тип <code>ToolBoxItem</code> базовый класс предназначен создания и визуального отображения <code>ToolBox</code> элемента на панели инструментов. Не поддерживается в Windows и ASP.NET сервисах.
<code>System.Drawing.Drawing2D</code>	Пространство имен используется для поддержки двумерной и векторной графики. В свою очередь оно сгруппировано по категориям: <ul style="list-style-type: none"> а) типы кистей (<code>PathGradientBrush</code> и <code>HatchBrush</code> типы позволяющие заполнять геометрические формы повторяющимся узором либо градиентом); б) перечисления связанные с рисованием линий <code>LineCap</code> и <code>Custom LineCap</code> типы определяющие стили концов линий, <code>LineJoin</code> перечисления определяющие как линии будут соединяться между собой, <code>PenAlignment</code> перечисления определяющие как объект <code>Pen</code> будет выравниваться относительно виртуальной линии, <code>PenType</code> перечисления определяющие заполнение линии; в) перечисления связанные с заполнением геометрических форм и путей <code>HatchStyle</code> перечисление определяющие стиль заполнения класса <code>HatchBrush</code>, <code>Blend</code> определяет смешивание для <code>LinearGradientBrush</code>, перечисления <code>FillMode</code> определяют стиль заполнения для типа <code>GraphicsPath</code>; г) геометрические трансформации <code>Matrix</code> класс представляющий матрицу 3x3 хранящий информацию о трансформировании над векторной графикой, изображением или текстом.

	<p>Не поддерживается в Windows и ASP.NET сервисах.</p>
System.Drawing.Printing	<p>Классы, имеющие отношение к сервису печати в Windows Forms, например PrintDocument, PrintSettings, PageSettings, печать производится с помощью вызова метода PrintDocument.Print() при этом срабатывает событие PrintPage, которое разработчик может перехватывать, PrinterResolution представляет разрешение, поддерживаемое принтером, перечисления PaperKind определяет стандартные размеры бумаги, например A3 или A4 и множество других типов.</p> <p>Не поддерживается в Windows и ASP.NET сервисах а также в приложениях ASP.NET</p>
System.Drawing.Imaging	<p>Содержит классы, позволяющие манипулировать графическими изображениями, например, класс ImageFormat определяет формат файла изображения, перечисление ImageFlags представляющее как данные о пикселях содержатся в изображении.</p> <p>Не поддерживается в Windows и ASP.NET сервисах.</p>
System.Drawing.Text	<p>Пространство, которое содержит классы для управления шрифтами, например, InstalledFontCollection позволяет получить список шрифтов, установленных на данной системе, TextRenderingHint определяет качество визуализации текста, класс PrivateFontCollection обеспечивает доступ к семейству шрифтов клиентского приложения.</p> <p>Не поддерживается в Windows и ASP.NET сервисах.</p>