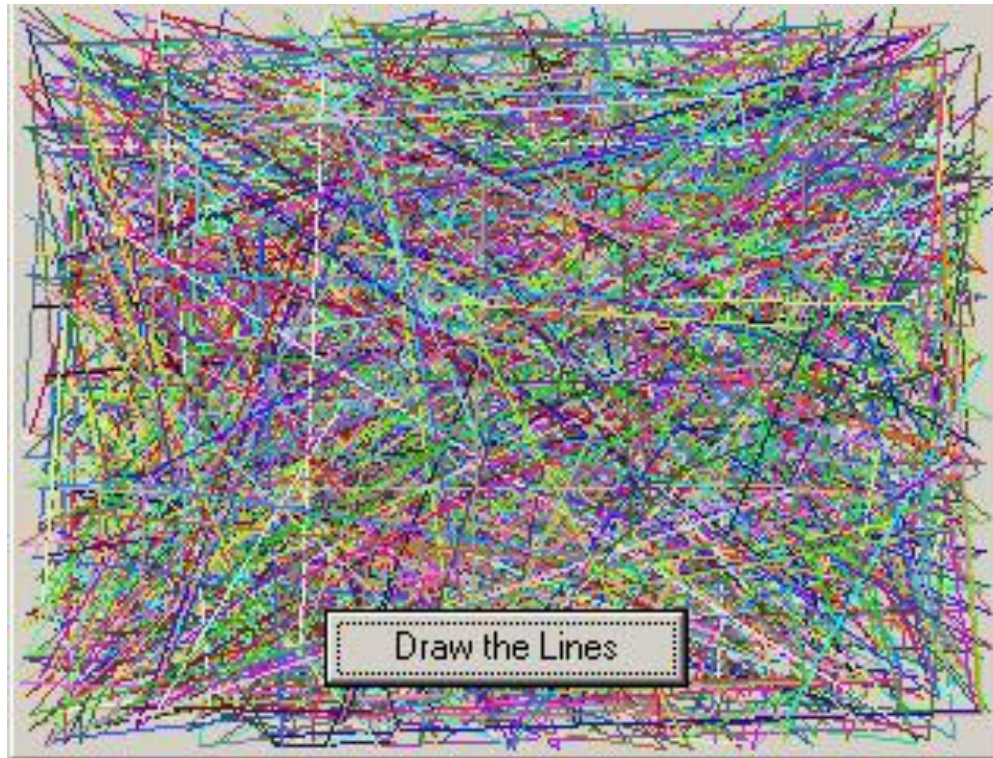


GDI

(Graphics Device Interface)



GDI (Graphics Device Interface) - один из трёх основных компонентов или «подсистем» (+ ядро Windows и Windows API), составляющих пользовательский интерфейс Microsoft Windows.

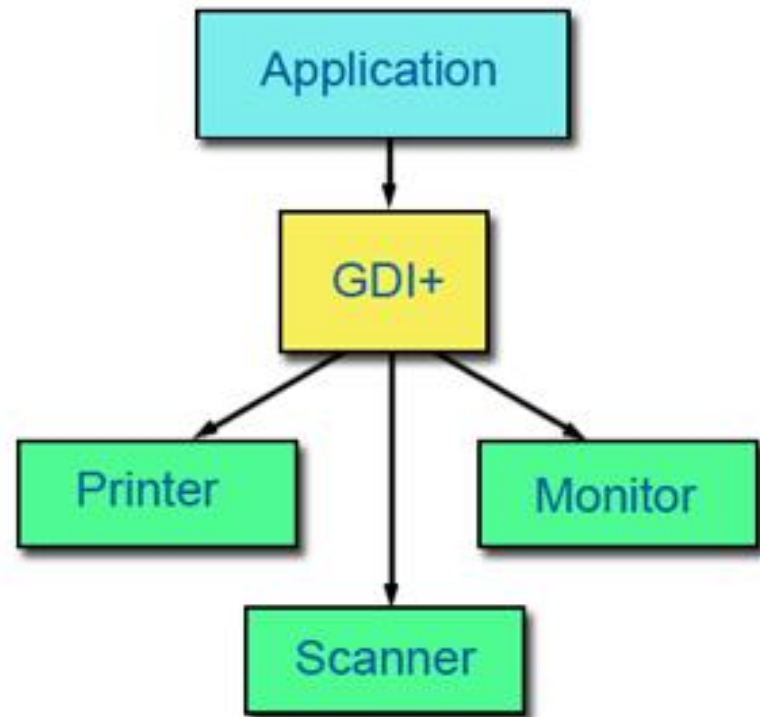
GDI - это интерфейс Windows для представления графических объектов, и передачи их на устройства отображения, такие как **мониторы и принтеры**.

GDI отвечает за отрисовку линий и кривых, отображение шрифтов и обработку палитры. Он не отвечает за отрисовку окон, меню и т. п., эта задача закреплена за пользовательской подсистемой, располагающейся в user32.dll.

Одно из преимуществ использования GDI - унификация работы с различными устройствами. Используя GDI, можно одними и теми же функциями рисовать на разных устройствах, таких как экран или принтер, получая на них практически одинаковые изображения.



GDI позволяет нам абстрагироваться от ограничений, связанных с разнообразием графических устройств, скрывая разницу между этими устройствами

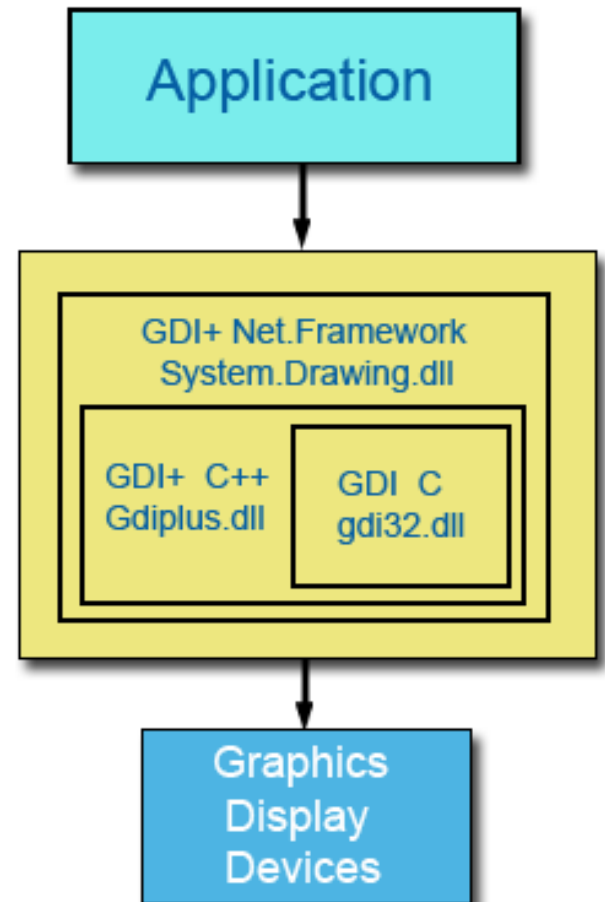


Приложения **GUI** (Graphical User Interface), взаимодействуя с этими устройствами, представляет данные в понятном для пользователя виде, используя при этом посредника (**GDI**), который принимает данные программы и направляет их устройству отображения.

GDI - это библиотека **Gdi32.dll**, она использовалась в ранних версиях Windows и базируется на старом Win32API с функциями языка C

GDI+ является компонентом операционных систем WindowsXP и Windows Server2003. Эта оболочка вокруг старой библиотеки **GDI**, она написана на языке C++ и представляет улучшенную производительность и более интуитивно понятную модель программирования, представляясь библиотекой **Gdiplus.dll**

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



- **Градиентная закрашка.**

- **Поддержка прозрачности.**

- **Режимы улучшения изображения:** позволяют значительно улучшить пользовательское восприятие за счет сглаживания контурных неровностей.

- **Пути:** существуют независимо от контекста рисования и представляют собой мощное средство создания сложных векторных объектов.

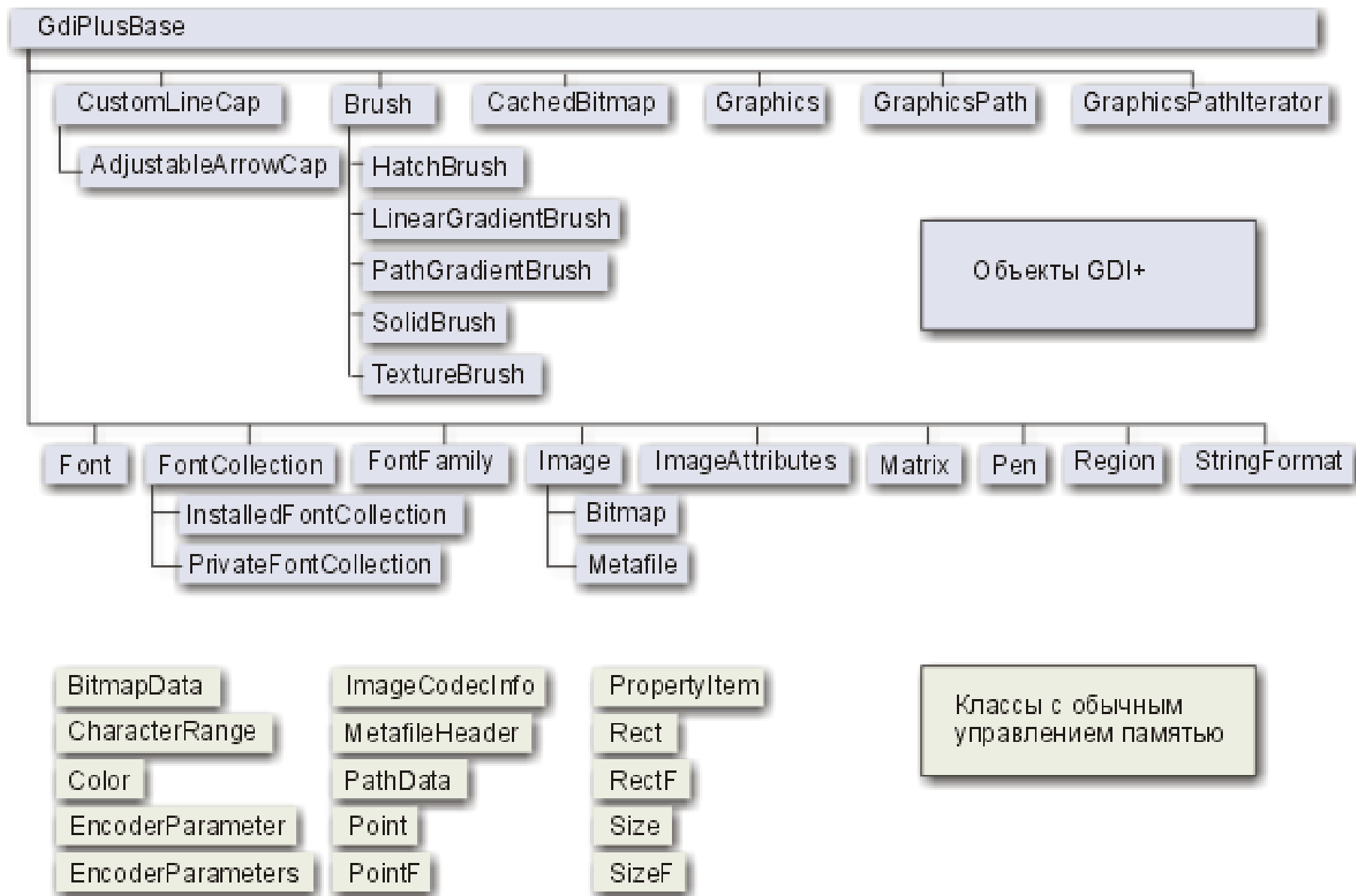
- **Координатные преобразования:** объект Matrix позволяет осуществлять операции поворота, переноса, масштабирования и отражения объектов GDI+.

- **Работа с растрами.**

- **Поддержка популярных форматов графических файлов:** необычайно приятное новшество для всех программистов, имеющих дело с разными графическими форматами. Поддерживаются форматы **BMP**, **GIF**, **TIFF**, **JPEG**, **Exif** (расширение **TIFF** и **JPEG** для цифровых фотокамер), **PNG**, **ICON**, **WMF** и **EMF**.

- **Формат EMF+:** для описания новых возможностей был создан новый формат метафайла EMF+, который позволяет сохранить на диск и затем проиграть последовательность графических команд.

Пространство System.Drawing

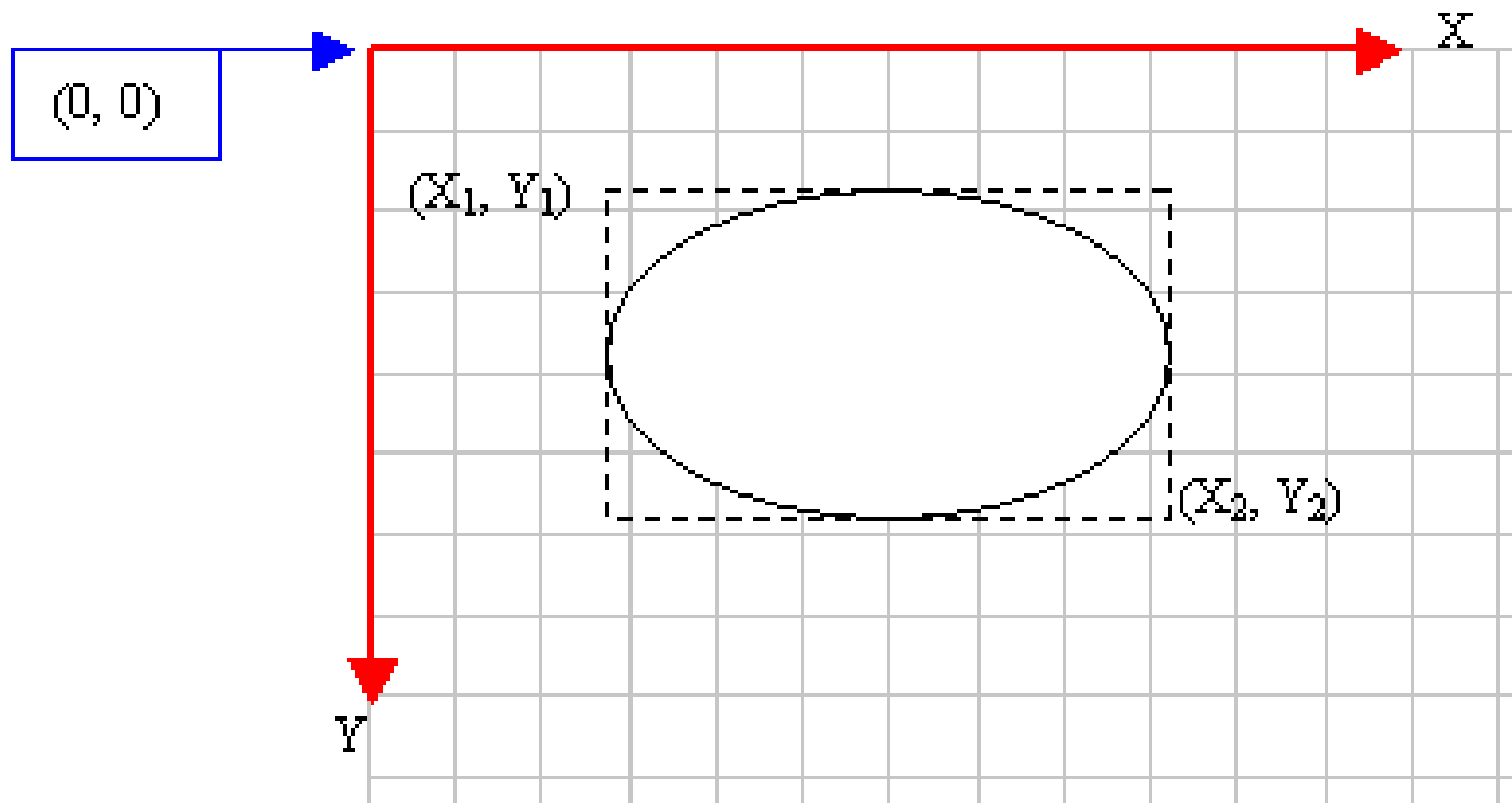


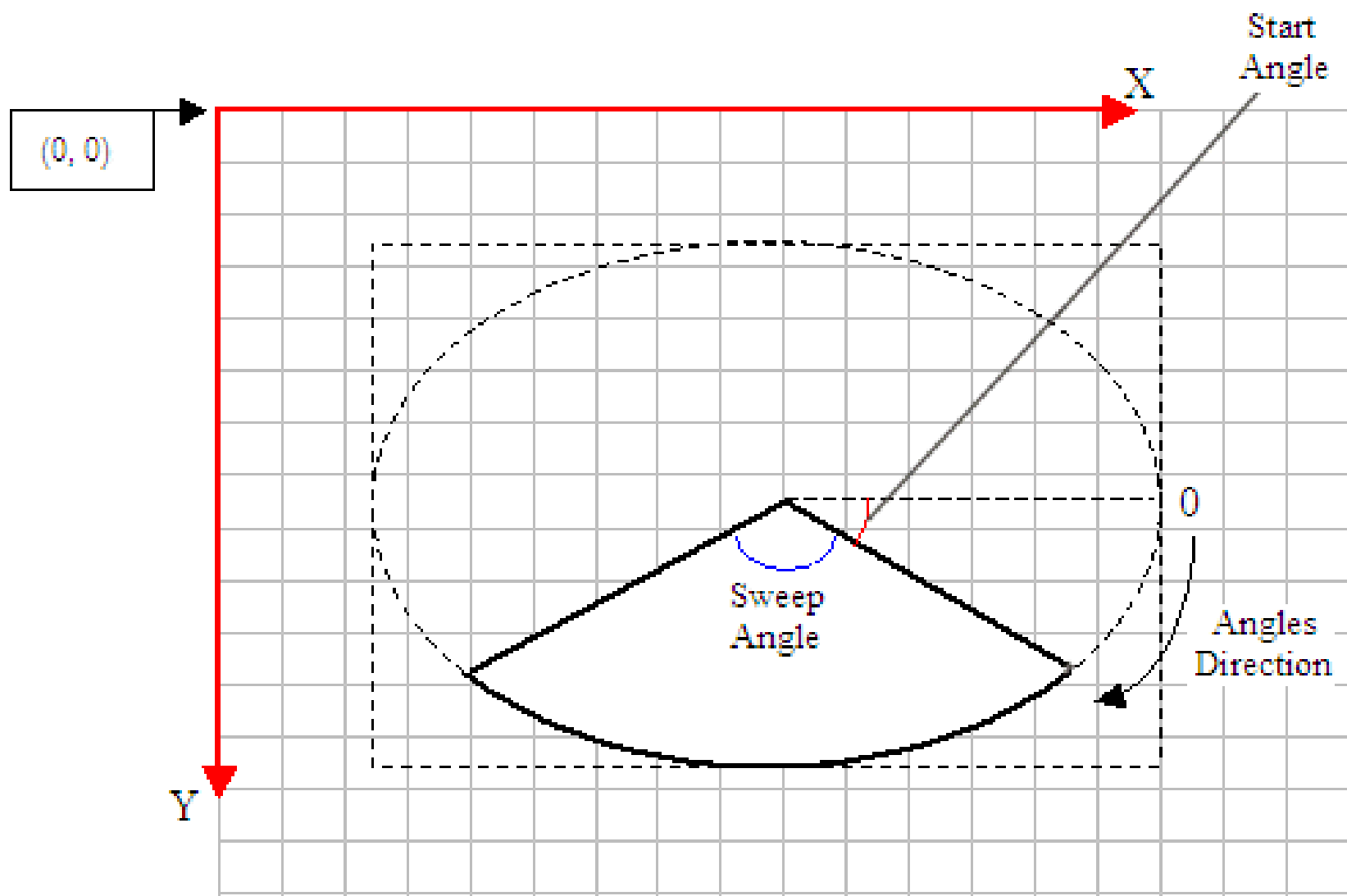
В этом пространстве можно найти классы, представляющие изображения, кисти, перья, шрифты и другие типы, позволяющие работать с графикой.

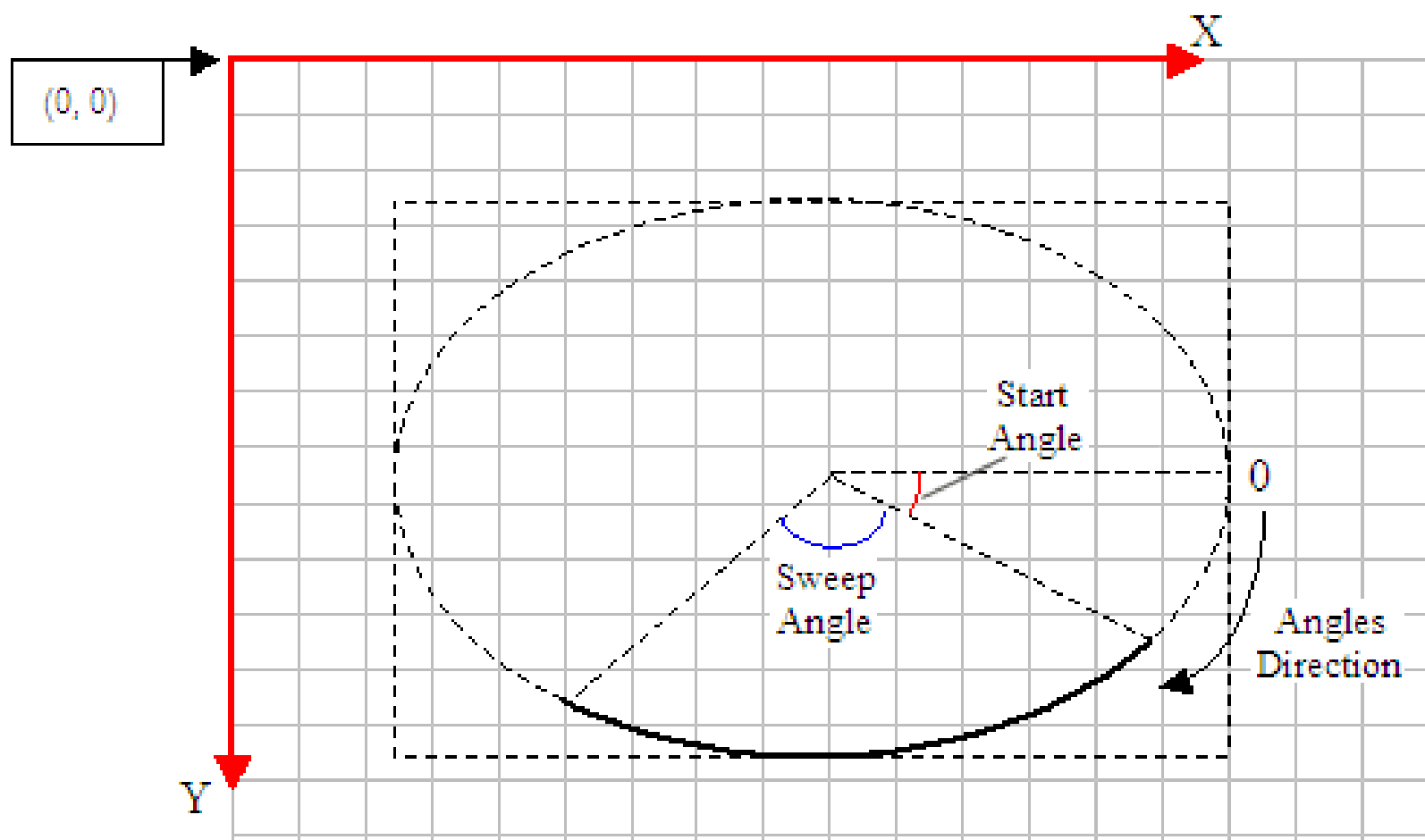
Дополнительная функциональность
обеспечивается следующими подпространствами
имен:

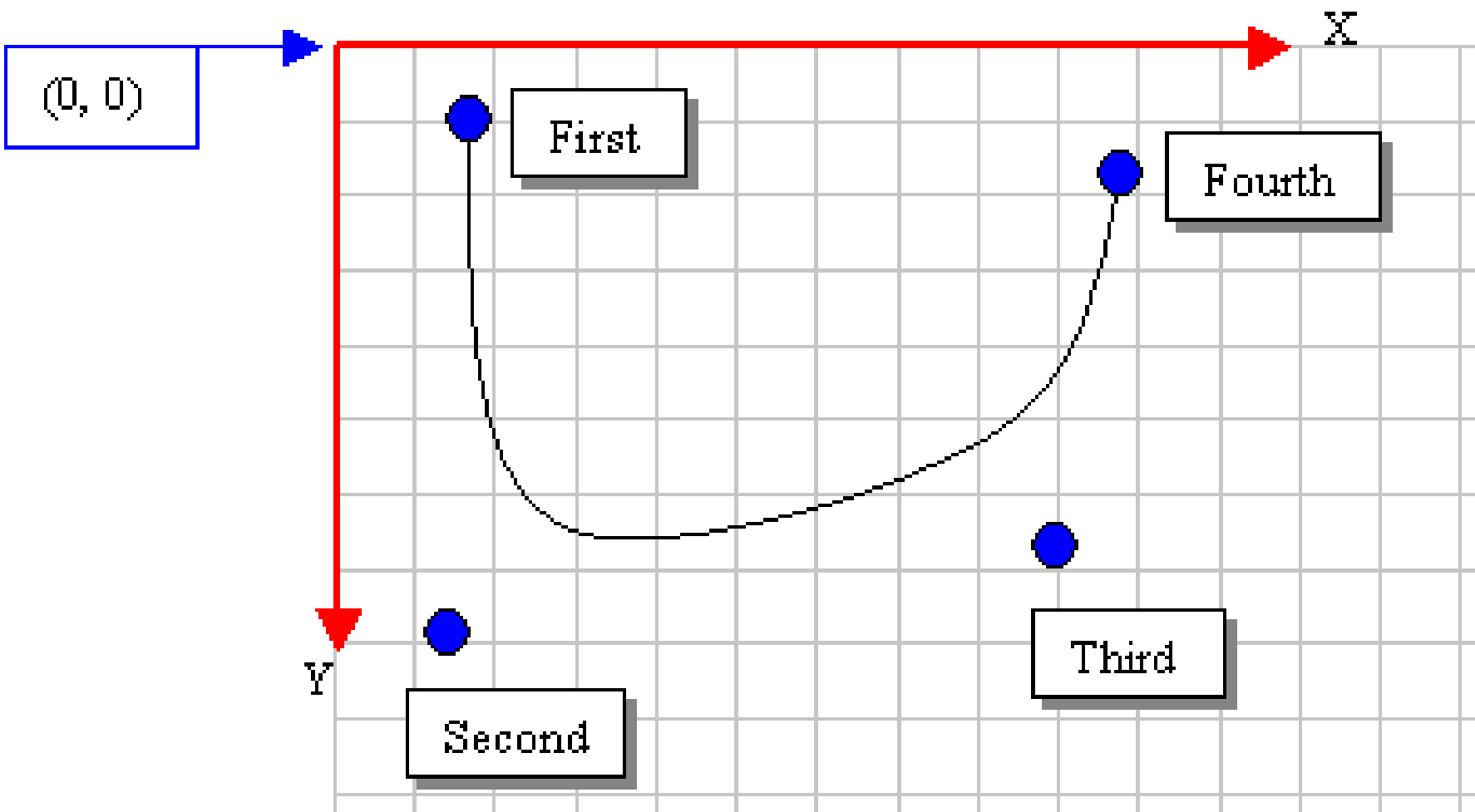
System.Drawing.Desing,
System.Drawing.Drawing2D,
System.Drawing.Imaging,
System.Drawing.Printing,
System.Drawing.Text.

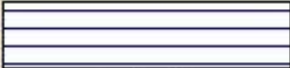


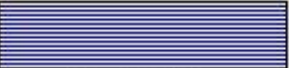


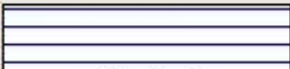














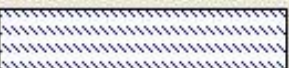





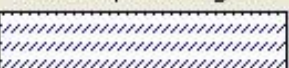




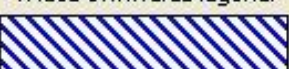
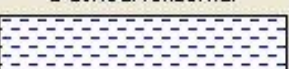
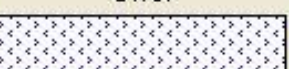



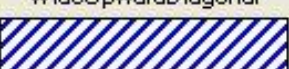

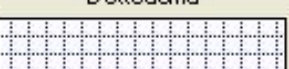

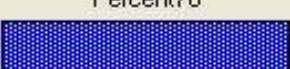





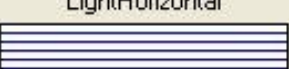


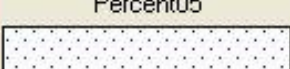




System.Drawing	Базовое пространство имен GDI+ определяет множество типов для основных операций визуализации, инкапсулируют примитивы GDI+.
System.Drawing.Desing	Пространство имен содержащее типы, обеспечивающие базовую функциональность для разработки расширений пользовательского интерфейса.
System.Drawing.Drawing2D	Пространство имен используется для поддержки двумерной и векторной графики
System.Drawing.Printig	Классы, имеющие отношение к сервису печати в Windows Forms
System.Drawing.Imaging	Содержит классы, позволяющие манипулировать графическими изображениями
System.Drawing.Text	Пространство, которое содержит классы для управления шрифтами

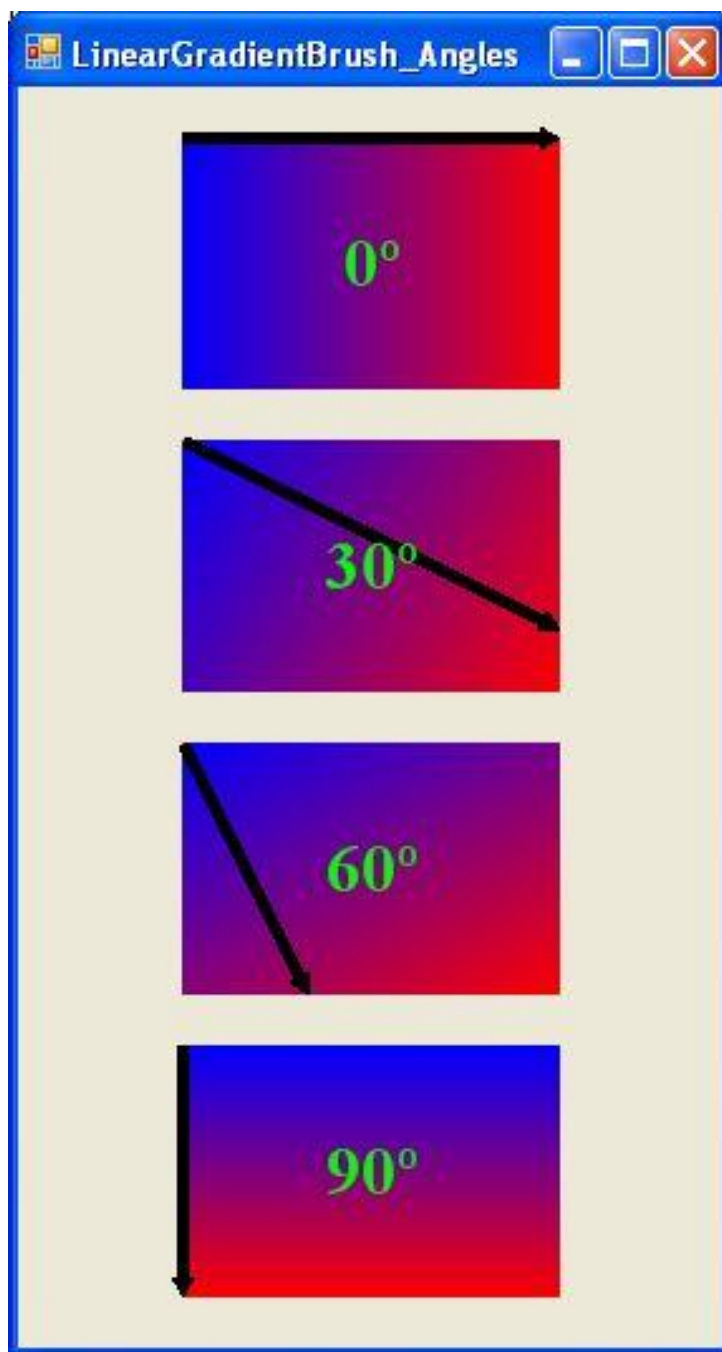








 Horizontal	 Percent10	 Percent90	 NarrowHorizontal	 Wave	 Sphere
 Horizontal	 Percent20	 LightDownwardDiagonal	 DarkVertical	 DiagonalBrick	 SmallGrid
 Vertical	 Percent25	 LightUpwardDiagonal	 DarkHorizontal	 HorizontalBrick	 SmallCheckerBoard
 ForwardDiagonal	 Percent30	 DarkDownwardDiagonal	 DashedDownwardDiagonal	 Weave	 LargeCheckerBoard
 BackwardDiagonal	 Percent40	 DarkUpwardDiagonal	 DashedUpwardDiagonal	 Plaid	 OutlinedDiamond
 LargeGrid	 Percent50	 WideDownwardDiagonal	 DashedHorizontal	 Divot	 SolidDiamond
 LargeGrid	 Percent60	 WideUpwardDiagonal	 DashedVertical	 DottedGrid	
 LargeGrid	 Percent70	 LightVertical	 SmallConfetti	 DottedDiamond	
 DiagonalCross	 Percent75	 LightHorizontal	 LargeConfetti	 Shingle	
 Percent05	 Percent80	 NarrowVertical	 ZigZag	 Trellis	



Перья, класс Pen



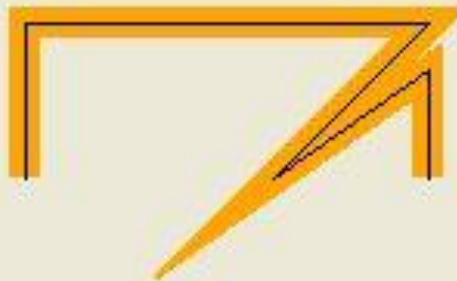
LineJoins



Bevel



Miter



MiterClipped



Round





PenCaps



AnchorMask



ArrowAnchor



DiamondAnchor



Flat



NoAnchor



Round



RoundAnchor



Square



SquareAnchor



Triangle





DashCaps



Flat



Round



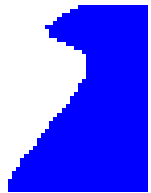
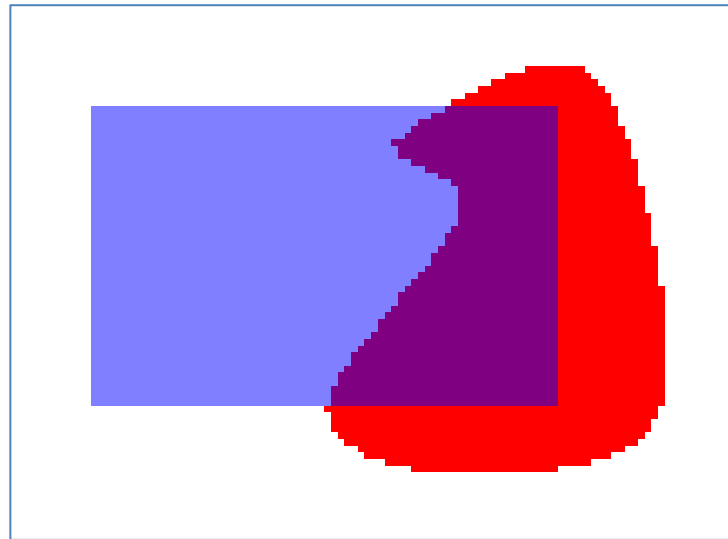
Triangle





SmoothLines

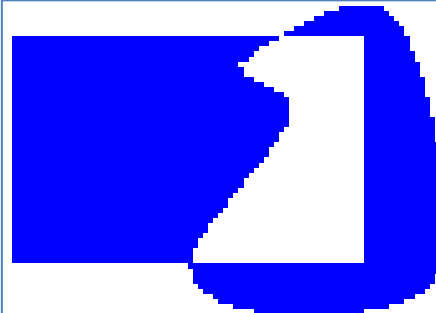




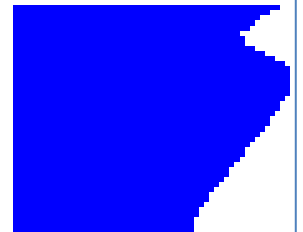
Intersection



Union



Xor



Exclude

Траектории в **GDI++** представлены классом **GraphicsPath** пространства имен **System.Drawing.Drawing2D**.

GraphicsPath представляет

- набор линий,
- кривых,
- графических объектов (прямоугольники, эллипсы и текст).

Траектории используются, например, для рисования контуров, заполнения внутренних областей, и при создании области отсечения.

Методы, для добавления фигур к объекту GraphicsPath

Название	Назначение
AddClosedCurve	Замкнутую кривую, заданную массивом точек
AddEllipse	Эллипс
AddPath	Заданный экземпляр объекта GraphicsPath
AddPie	Окружность
AddPolygon	Многоугольник, заданный массивом точек
AddRectangle	Прямоугольник
AddString	Графическое представление строки символов с заданным начертанием

методы объекта GraphicsPath, позволяющие добавлять
новые элементы контура фигуры

Название	Назначение
AddArc	Дугу
AddBezier	Кривую Безье
AddBeziers	Набор кривых Безье
AddCurve	Кривую, заданную массивом точек
AddLine	Прямую
AddLines	Набор соединенных прямых

<u>CloseAllFigures</u>	Замыкает все незамкнутые фигуры в этом контуре и открывает новую фигуру. Каждая незамкнутая фигура замыкается путем соединения ее начальной и конечной точек линией.
<u>CloseFigure</u>	Замыкает текущую фигуру и открывает новую фигуру. Если текущая фигура содержит последовательность соединенных линий и кривых, метод замыкает ее путем соединения начальной и конечной точек линией.
<u>StartFigure</u>	Открывает новую фигуру, не замыкая при этом текущую фигуру. Все последующие точки, добавляемые к контуру, добавляются к этой новой фигуре.

Перечисление PathPointType

Описание	
Bezier	Кривая Безье, используемая по умолчанию.
Bezier3	Кривая Безье третьего порядка.
CloseSubpath	Конечные точки субконтура.
DashMode	Соответствующий сегмент является пунктирным.
Line	Отрезок прямой.
PathMarker	маркер контура.
PathTypeMask	Точка маски.
Start	Начальная точка объекта GraphicsPath .

Кривая безье определяется начальной точкой, двумя контрольными точками и конечной точкой. В отличие от квадратичной кривой, кривая безье определяется двумя контрольными точками, а не одной, что позволяет создавать более сложную кривизну.

