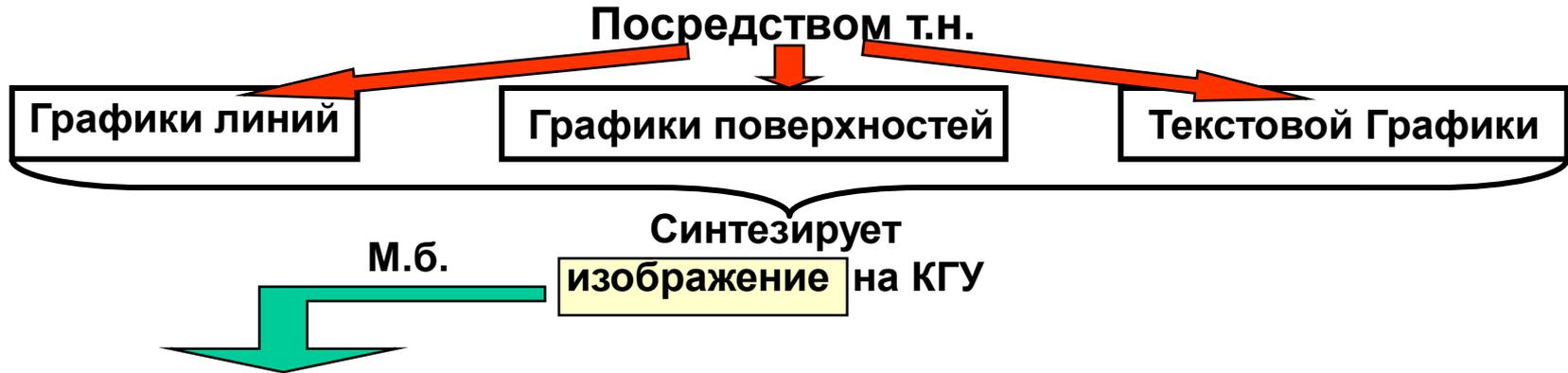


Лекция № 4

**Часть I. Машинная Графика
Графическая Система
Базовая Графическая Система**

Графическая Система



Структурировано на подизображения - сегменты

М.б.

Трансформированы:

Поворот
Сдвиг
Масштабирование

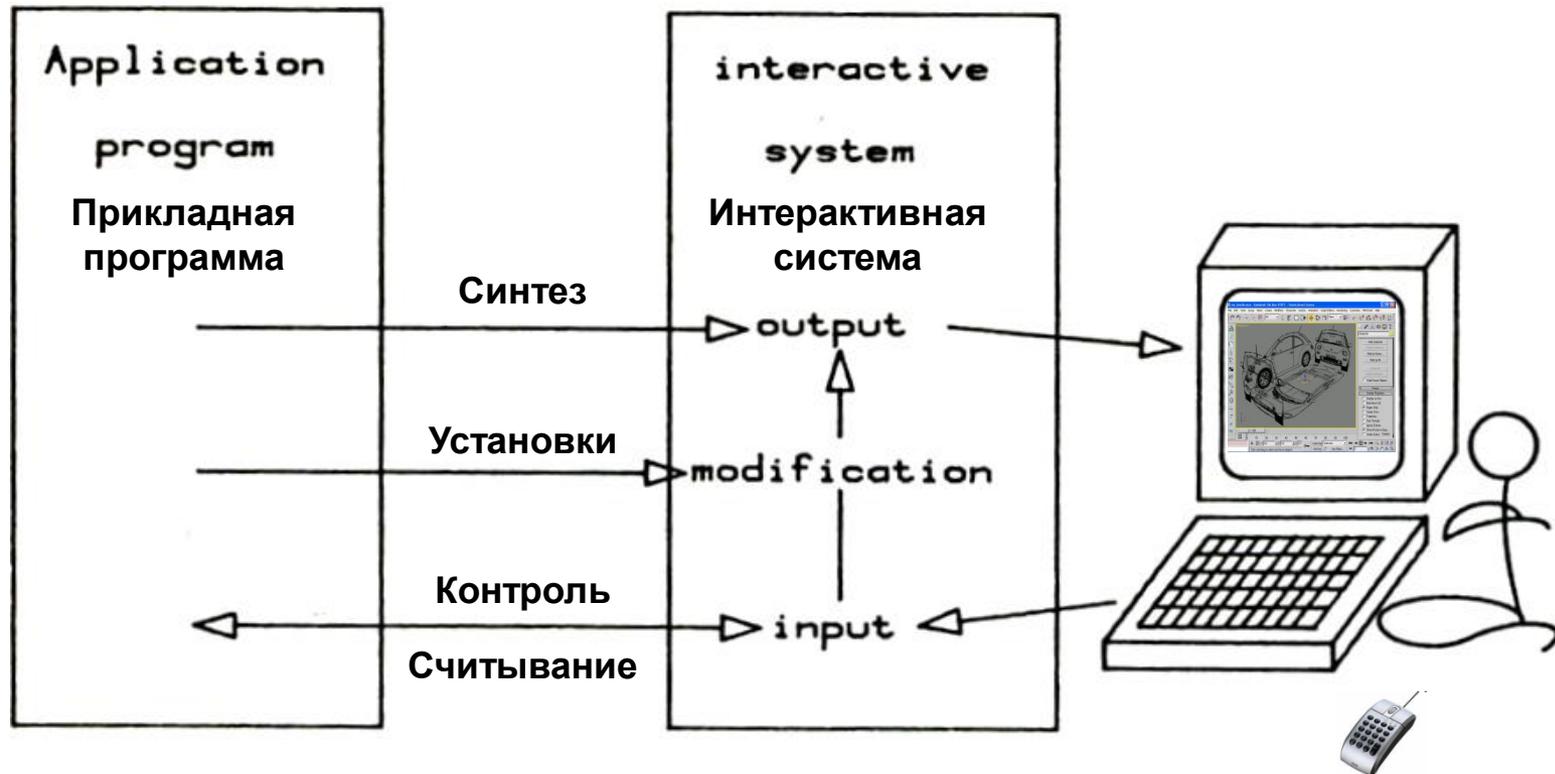
Удалены

Выделены

Для полного представления человеко-машинного интерфейса в МГ – в наборе средств и методов, включенных в состав **Графической Системы** – были разработаны средства и методы, которые сделали возможным графический ввод и обеспечили возможность интерактивной работы пользователя с изображением.

Системы машинной графики.

Интерактивные Графические системы.



Интерактивные **системы МГ** – это Программно-Технические комплексы (ПТК), которые предназначены для воспроизведения графического изображения (аналогично Пассивным системам МГ), но, кроме того, оснащены Hard-Soft компонентами, которые поддерживают интерактивный способ работы пользователя, т.е. дают ему возможность манипулировать изображением в реальном времени, и предоставляют ему средства удобного и легко осваиваемого графического диалога.

Все вышесказанное позволяет сформулировать требования к Графической системе как к ядру МГ:

Развитые средства и методы для синтеза изображений – пассивные системы

Развитые средства и методы для манипулирования изображением, моделирование работы разнообразных устройств графического ввода – интерактивные системы

Разработка графических и формальных типов данных, которыми манипулирует система и которые она предоставляет пользователю:

Поверхности, линии, тексты;

Возможность структурировать изображение, установить связи между отдельными частями изображения;

Разработать и реализовать алгоритмы манипулирования изображением

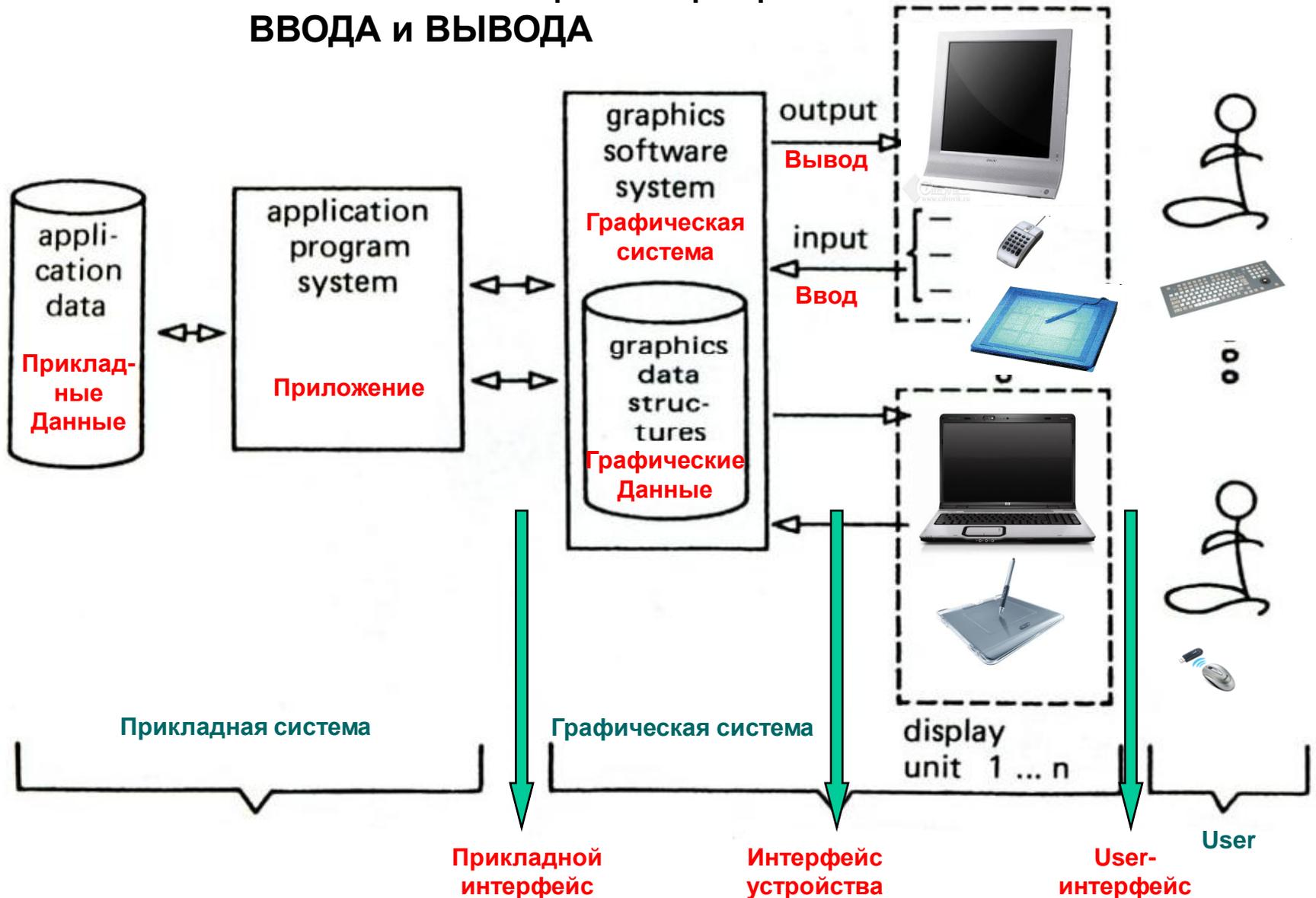
Бурное развитие Hard-Soft технологий в области МГ, а также разработка стандартов в области матобеспечения, заставило искать общую методологию и терминологию для ГС. В результате была разработана единая модель конфигурации Графической Системы, как

Программно-Технического Комплекса



Базовая конфигурация Графической Системы

Основана на основополагающих концепциях:
ВВОДА и ВЫВОДА



Базовая Графическая Система

К вопросам стандартизации в МГ вплотную приступили в конце 70-х годов. Тогда и был предложен первый проект GSPC CORE SYSTEM, разработанный американским комитетом по стандартизации в машинной графике GSPC ACM\SIGGRAPH, оказавший большое влияние на разработку первого Международного стандарта Graphical Kernel System (GKS) – Ядро Графической Системы (ЯГС).

GKS – Ядро Графической Системы (ЯГС) определяет :

Набор функций синтеза изображений

Представления изображений на внешних устройствах

Сегментации изображений

Преобразования

Графического ввода

Впервые разработана единая методологическая база, объединяющая на своей основе различные концепции, касающиеся различных сторон проблематики машинной графики. Это фундамент для единой трактовки и общей терминологии, необходимых при создании графических систем и их использования, для обсуждения проблем машинной графики специалистами и для обучения методам, концепциям и приложениям машинной графики.

Модель оболочек GKS

GKS - набор функций для выполнения графических действий, описанных в языково-независимой форме.

Реально эти функции - подпрограммы (процедуры) конкретного языка программирования. Такая зависящая от языка программирования часть реализации, в которую заключено языково-независимое ядро системы, называется *языковой оболочкой*.



Модель оболочек GKS

Функции, обеспечиваемые языковой оболочкой, могут использоваться прикладным программистом наравне с запросами к операционной системе. Специфические слои, зависящие от приложения, могут быть построены над языковой оболочкой GKS (картография, машиностроительные САПР, Иллюстративные программы, средства построения диаграмм и пр.).

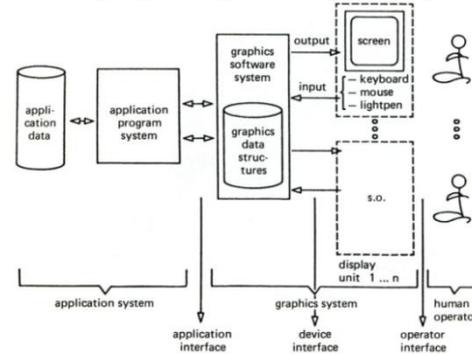
Модель оболочек, представленная на предыдущем слайде, отражает место GKS в графической системе. На каждом уровне доступны функциональные возможности всех подчиненных оболочек (слоев). Таким образом, прикладная программа имеет в своем распоряжении ресурсы некоторого числа проблемно-ориентированных оболочек (слоев), языковой оболочки GKS и операционной системы. Доступ же к графическим возможностям осуществляется исключительно через функции GKS.

Рассмотрим интерфейсы GKS



Интерфейсы Ядра Графической системы (GKS)

Детализируем схему базовой конфигурации ГС, раскрывая понятие интерфейса:



Интерфейс
Оператора -
Userinterface



Интерфейсы Ядра Графической системы (GKS)

Наиболее внешним относительно GKS является **языково-независимый проблемный интерфейс**. Языковой интерфейс, (имеем в виду языки программирования высокого уровня), связывает языковую оболочку с проблемными оболочками. Спецификации этих интерфейсов разработаны в процессе создания реализаций стандарта GKS.

GKS называют ядром системы не только потому, что оно обеспечивает множество различных приложений с помощью базовых графических средств. Ядро графической системы также позволяет задавать графические функции независимо от конкретных графических устройств. Этим объясняется наличие еще одного **важного интерфейса между ядром системы и различными устройствами ввода-вывода, доступными ему**. Прямой и обратный перевод приборно-независимого представления функций, принятого внутри ядра, в различные представления, специфические для РАЗЛИЧНЫХ графических станций, осуществляют **драйверы устройств**.

Драйвер устройства (драйвер графической станции) – это приборно-зависимая часть реализации GKS, предназначенная для поддержки графического устройства. Драйвер устройства обеспечивает генерацию графического вывода и интерактивное взаимодействие в формате данного устройства.

На предыдущем слайде GKS представлена как посредник между проблемным интерфейсом и интерфейсом графической станции. В случае интерактивной станции ее оператор взаимодействует с системой с помощью **интерфейса оператора**.

Рассмотрим основные понятия GKS



Основные понятия Ядра Графической системы (GKS)

Существует тесная взаимосвязь между концепциями, положенными в основу графической системы, и задачами, для решения которых она предназначена.

Отметим некоторые из этих задач:

синтез и воспроизведение изображений;

направление частей изображения, определенных в различных пользовательских системах координат, на различные графические станции и преобразование их координат в координаты соответствующих устройств;

управление станциями, к которым имеет доступ система;

обслуживание ввода данных со станций;

поддержка разбиения изображения на части, которые можно независимо обрабатывать (рисовать, преобразовывать, копировать, удалять);

долговременное хранение изображений.

Рассмотрим основные концепции GKS, позволяющие решить эти задачи



Основные концепции Ядра Графической системы (GKS)

Концепция Графического Вывода: одним из главных предназначений системы является создание изображений. Для решения этой задачи служит концепция *графического вывода*.

Функциональный интерфейс графической системы предоставляет для построения изображения базовые элементы, называемые **примитивами вывода**, чье визуальное представление на КГУ определяется **набором атрибутов** (например, цветом, толщиной линии).

Примитив вывода
(Output primitive):
базовый графический элемент, который может использоваться для построения изображения. В GKS к примитивам вывода относятся ЛОМАНАЯ, ПОЛИМАРКЕР, ТЕКСТ, ПОЛИГОНАЛЬНАЯ ОБЛАСТЬ, МАТРИЦА ЯЧЕЕК и ОБОБЩЕННЫЙ ПРИМИТИВ ВЫВОДА.

Изображение
(Display image):
совокупность графических примитивов и/или сегментов, которая может быть одновременно выведена на носитель изображения.

Атрибут:
характеристика примитива вывода или сегмента, например – цвет примитива, тип и толщина линии, выделение, межлитерный просвет и т.д.

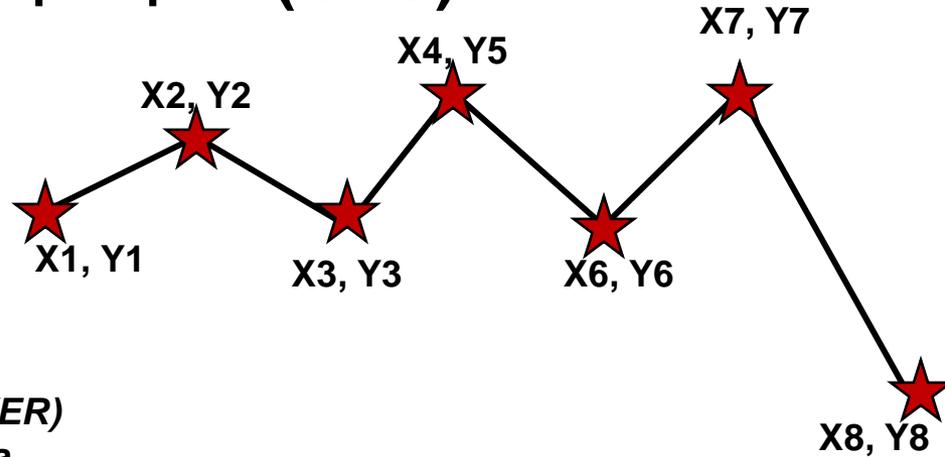


Основные концепции (GKS)

Концепция Графического Вывода

Примитивы вывода:

Векторный примитив: **ЛОМАНАЯ (POLYLINE)**
GKS генерирует набор отрезков прямых, соединяющих заданную последовательность точек.

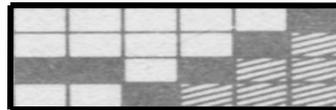


Точечный примитив: **ПОЛИМАРКЕР (POLYMARKER)**
GKS генерирует набор символов некоторого типа, которые центрируются в указанных точках.

Текстовый примитив: **ТЕКСТ (TEXT)**
GKS генерирует строку литер с началом в указанной позиции.



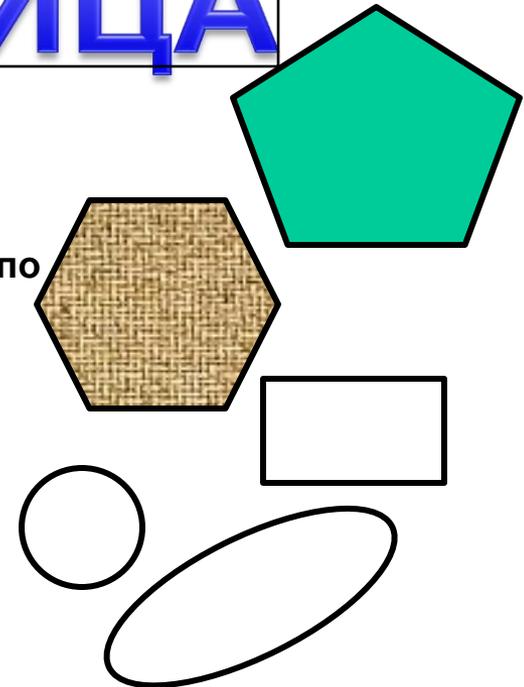
Растровые примитивы: **ПОЛИГОНАЛЬНАЯ ОБЛАСТЬ (FILL AREA)**
GKS генерирует многоугольник; область, которую он ограничивает, может быть пустой, иметь фоновую окраску, быть покрытой узором по шаблону или заштрихованной.



МАТРИЦА ЯЧЕЕК (CELL ARRAY)

GKS генерирует матрицу прямоугольных ячеек, каждой из которых присвоен индивидуальный цвет.

Примитив общего назначения: **ОБОБЩЕННЫЙ ПРИМИТИВ ВЫВОДА (ОПВ) (GENERALIZED DRAWING PRIMITIVE - GDP)**



Основные концепции (GKS). Концепция Графического Вывода

Атрибуты Примитивов вывода:

ЛОМАНАЯ (POLYLINE)

Тип линии, толщина линии, цвет

ПОЛИМАРКЕР (POLYMARKER)

Тип маркера, масштаб маркера, цвет

ТЕКСТ (TEXT)

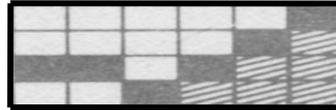
Шрифт, Высота литер, Вертикаль литеры, Масштаб расширения литеры, Направление текста, Межлитерный просвет, Выравнивание текста, цвет

ПОЛИГОНАЛЬНАЯ ОБЛАСТЬ (FILL AREA)

Вид заполнения: пусто, заливка цветом, по шаблону (размер шаблона, матрица шаблона, точка привязки шаблона), штриховка (вид штриховки)

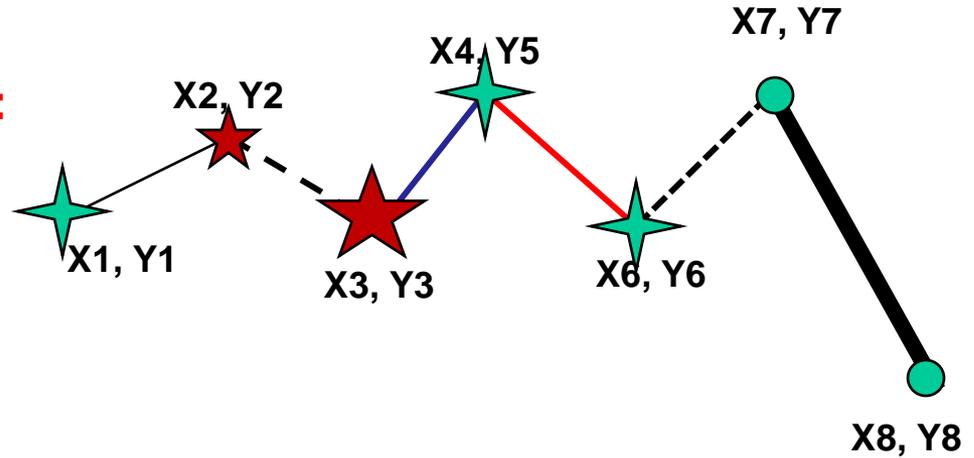
МАТРИЦА ЯЧЕЕК (CELL ARRAY)

Цвет



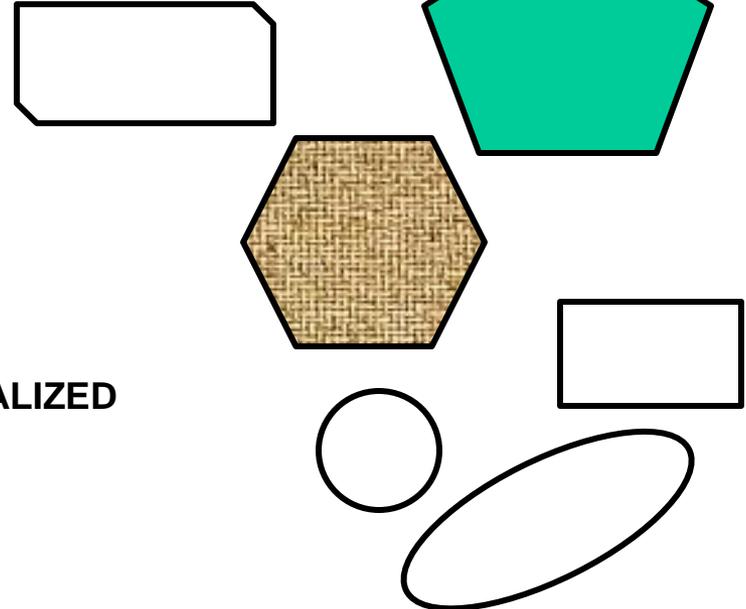
ОБОБЩЕННЫЙ ПРИМИТИВ ВЫВОДА (ОПВ) (GENERALIZED DRAWING PRIMITIVE - GDP)

Цвет



МАТРИЦА

(X, Y)



Основные концепции (GKS). Системы координат и преобразования

Примитивы вывода могут задаваться при создании в одной или нескольких системах координат пользователя. Эти примитивы должны быть размещены на носителе изображения различных графических станций с различными приборными системами координат. Последовательность изменений, претерпеваемая графическим выводом на пути от прикладной программы к носителю изображения устройства, называется *видовым конвейером* (viewing pipeline) .

Управление направлением перемещения и преобразованием примитивов вывода, проходящих через видовой конвейер, берет на себя графическая система. Используя соответствующие функции, прикладная программа может воздействовать на преобразование вывода. В трехмерных системах видовое преобразование трехмерных координат на двумерную картинную плоскость является одной из операций видового конвейера.