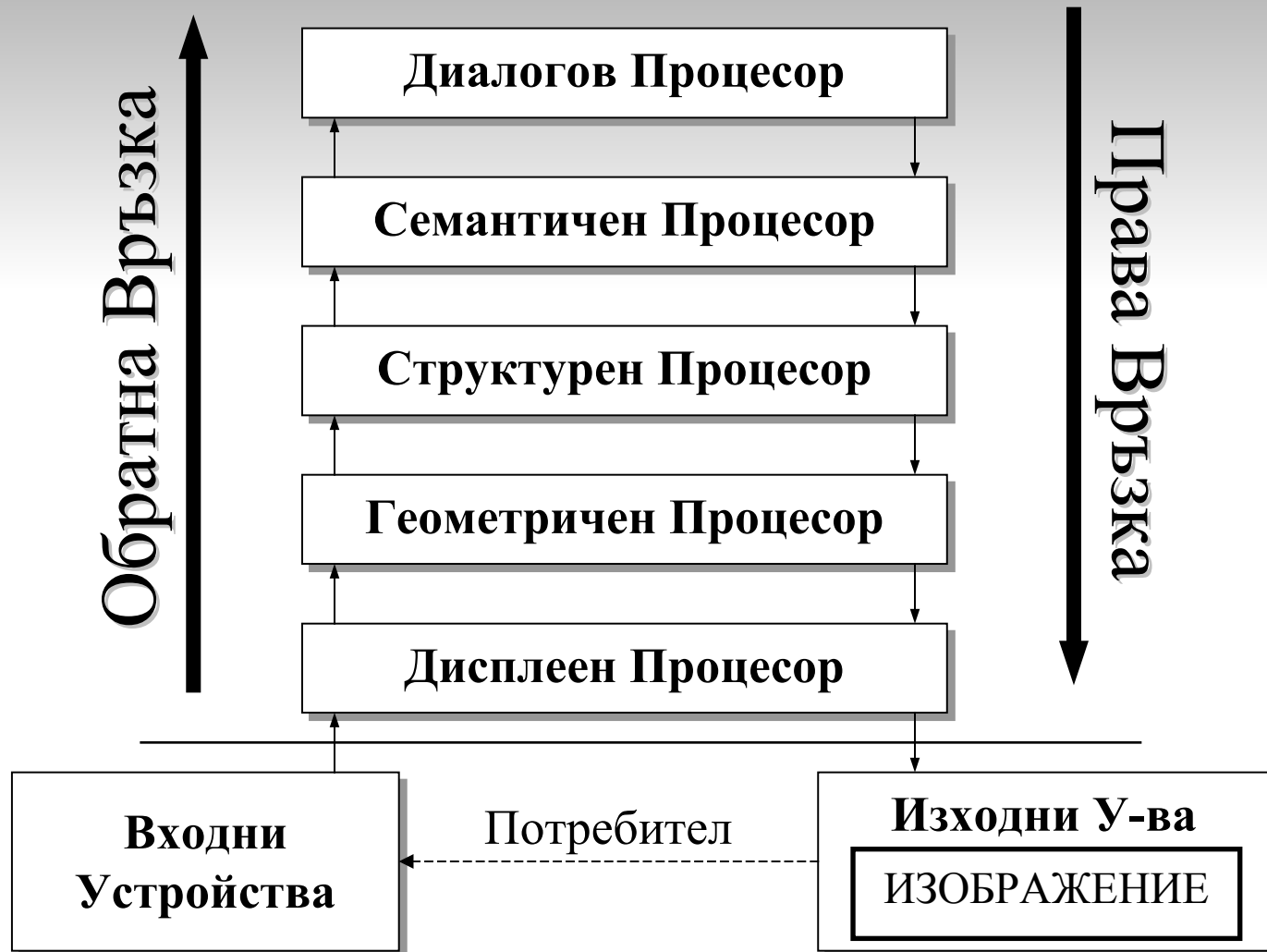




Компютърна Графика и Презентации

Виртуални машини:
Дисплеен, Геометричен, Структурен,
Семантичен и Диалогов процесори

Модел на диалогова графична система

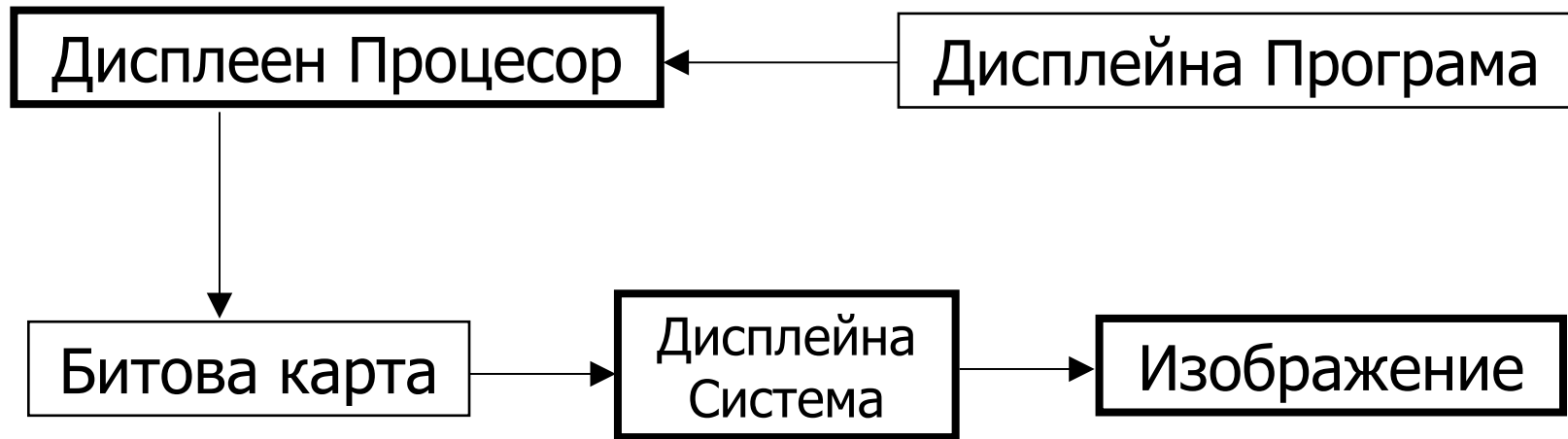


Дисплеен процесор

- Дисплейният процесор е виртуална машина, която **изпълнява** някаква **програма**, и като резултат произвежда **някакво изображение (рисунка)** чрез **графопостроител**.
- Обикновено **Дисплейният процесор** изпълнява програма, и произвежда битова карта, която е вход за Дисплейната система. Този най-често срещан процес на визуализация се нарича още **растеризация**.

Схема на виртуална машина

Дисплеен процесор



Дисплеен процесор

Виртуалната машина Дисплеен процесор се определя като алгоритъм, преобразуващ (конвертиращ) представянето на изображение като дисплейна програма в представяне като битова карта.

Как се дефинира работата на ДП

- Дефиниционната област – елементите и структурата на дисплейната програма;
- Областта от стойности – елементите и структурата на битовата карта;
- Алгоритъма – правилото, по което се извършва съпоставянето на елементите от двете области.

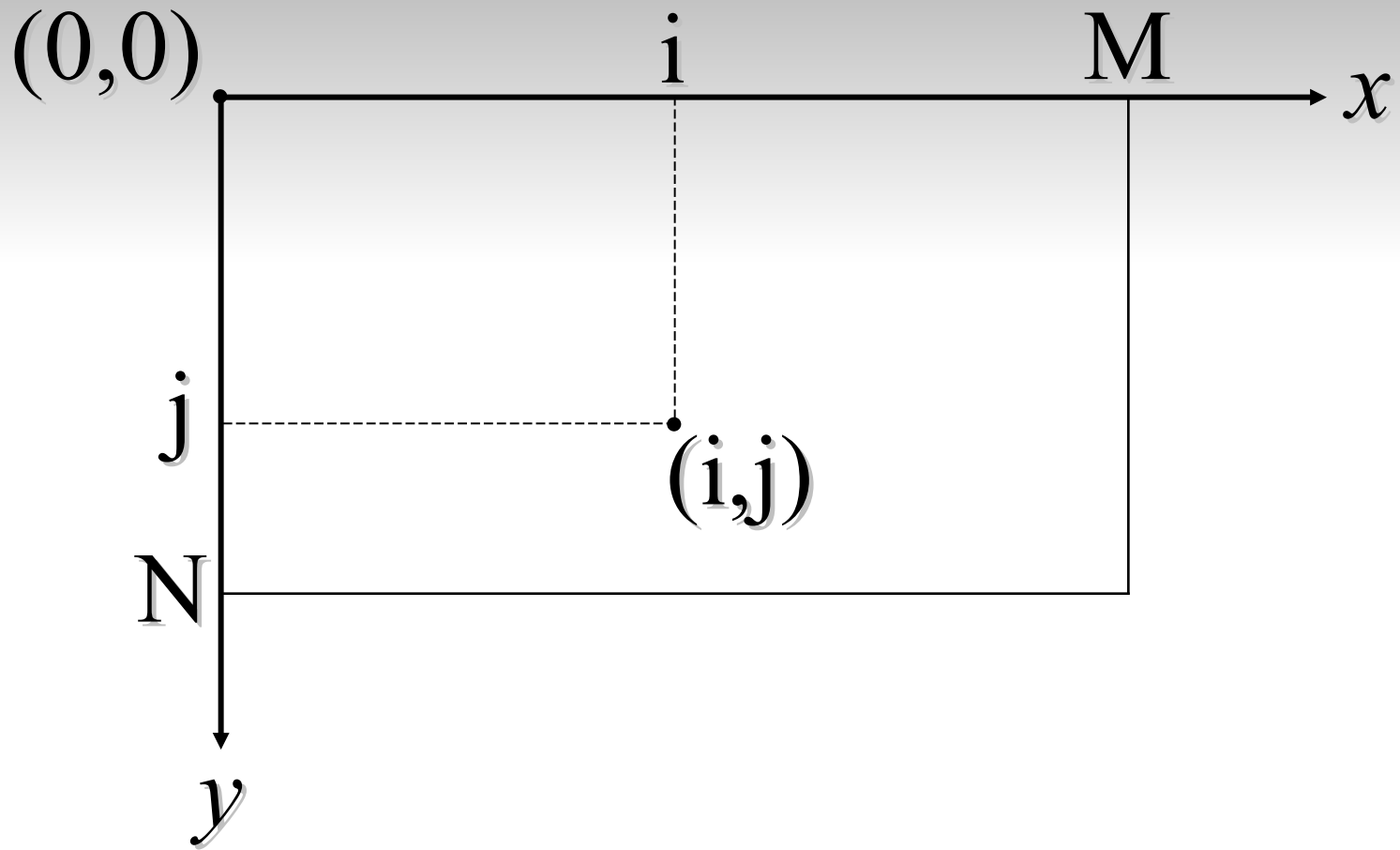
Екранна област (SD)

Екранната област се разглежда логически като множеството от наредени двойки числа $\{ (i,j) \mid 0 \leq i \leq M, 0 \leq j \leq N \}$, където M и N са естествени числа.

Всяка такава двойка числа се нарича точка, а числата, които я съставят – координати.

Първото число се нарича абсциса на точката или x -координата, а второто – ордината на точката или y -координата.

Екранна област (SD)



Характеристики на SD (1/2)

- Пространството е дискретно – координати на точките могат да бъдат само цели числа, от предварително определени затворени интервали;
- Битовата карта може да се разглежда като масив с $M \times N$ елемента, като всеки елемент се състои точно от Q бита, кодиращи визуализационните характеристики на пиксела.

Характеристики на SD (2/2)

- Тъй като лъча обхожда пикселите на екрана от ляво на дясно и отгоре надолу, то мястото на всеки пиксел битовата карта се определя по формулата:

$$k = (y * M) + x,$$

където (x, y) са координатите на точка, k е мястото на визуализационните характеристики на тази точка в битовата карта и:

- $0 \leq x \leq M,$
- $0 \leq y \leq N,$
- $0 \leq k \leq M * N + M + N.$

Дисплеен файл

- Дисплейната програма се нарича още **Дисплеен файл**;
- Това е основното представяне на изображението (то се получава като изход/данни от друга виртуална машина), като то непрекъснато се променя в процеса на диалог;
- Описва елементите на изображението.

Дисплейният файл описва:

- Типа на графичните примитиви, от които се състои изображението;
- Визуализационните характеристики на елементите на изображението;
- Местоположението и ориентацията в екранното пространство на графичните примитиви в изображението;
- Имена на групите от графични примитиви, чрез които се разпознават части на изображението.

Графични примитиви

Възприето е типът на графичните примитиви да се определя чрез команди за Дисплейният процесор, т. е. на всеки тип графичен примитив съответства команда от езика на Дисплейният процесор.

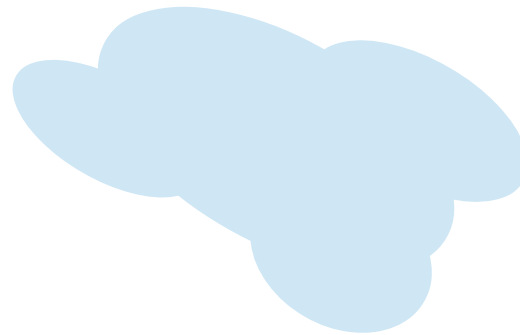
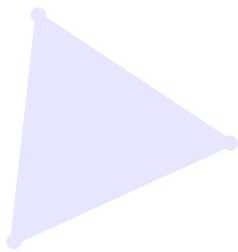
Най-често графичните примитиви биват:

- Точка;
- Отсечка;
- Знак (Литера).



Графични обекти

Графичен обект се нарича множество от примитиви, притежаващи еднакви визуализационни характеристики, статус и име.



Визуализационни Характеристики

Визуализационните характеристики се описват чрез команди на дисплейният процесор, имащи област на действие до следваща команда от този вид.

Например, ако има команда за смяна на цвета, с който се изобразяват примитивите, то цвета е един и същ до промяната му с друга такава команда.

СИМВОЛИ

- Дисплейната програма може да съдържа и подпрограми;
- Тези подпрограми се наричат **СИМВОЛИ**;
- Изпълнението им се инициира чрез команда за изпълнение на символ, която съдържа името на подпрограмата;
- Всяко изпълнение на символ създава екземпляр на символа, който става част от графичният обект, иницирал символа.

Идентифициране на Графичен обект

- Идентификация обикновено е по зададена точка;
- Начин на действие – изпълнява се отново дисплейната програма, но без запис в битовата карта;
- Резултат – име на графичен обект(и).

Идентифицираните обекти трябва да бъдат открити от останалото изображение, и това е сигнал за оператора, че даденият обект е указан от него, но това е диалогова функция на системата.

Геометричен процесор

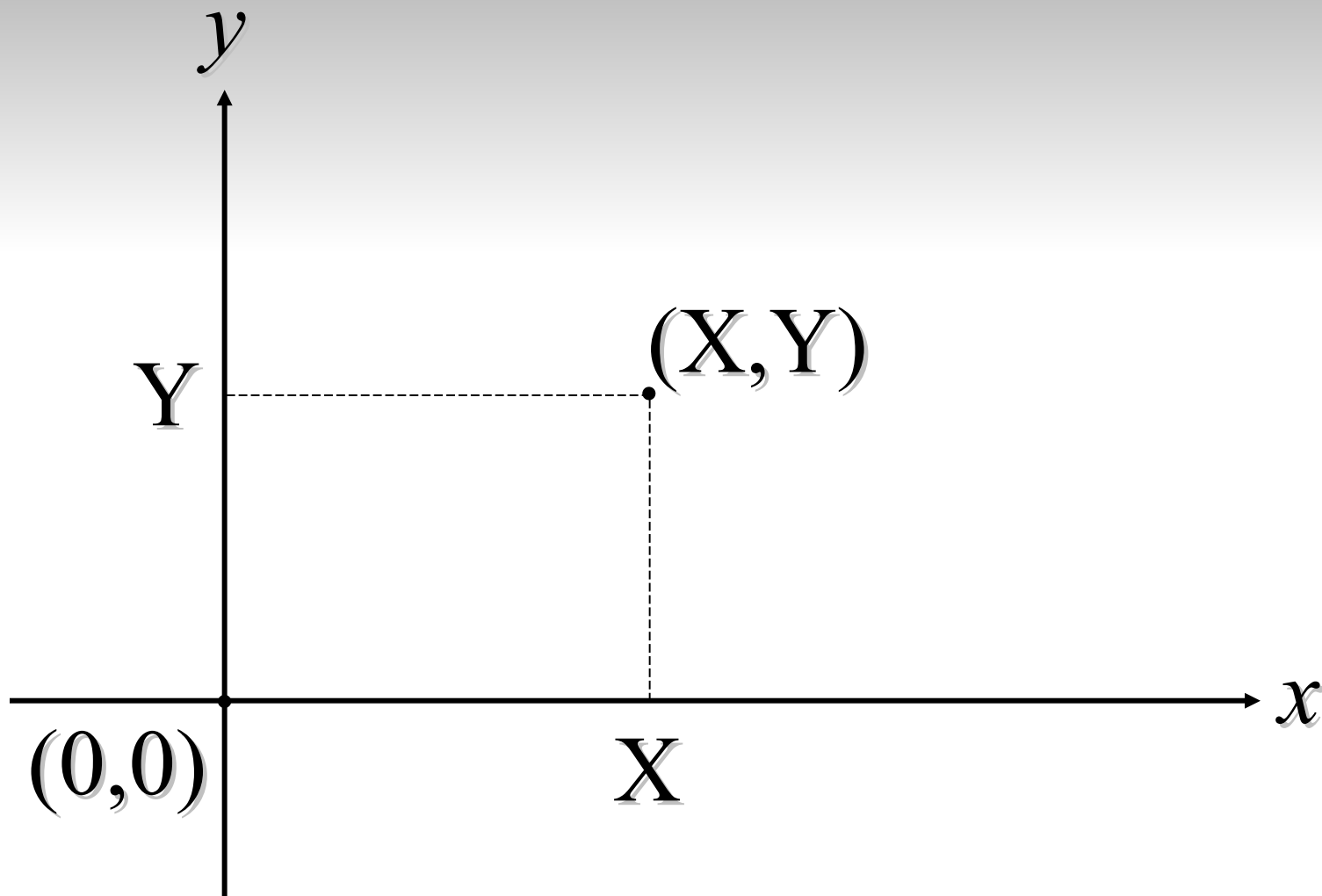
Основното предназначение на Геометричният процесор е реализиране на геометрични преобразувания с изображението.

Това обикновено се свежда до геометрични преобразувания на точки и вектори, тъй като обектите в изображението се описват на базата на такива математически обекти.

Световна област (WD)

- **Световната област** се обозначава като **WD** (World Domain), и дефинира като $WD = \mathbb{R} \times \mathbb{R}$, където с \mathbb{R} се обозначава множеството на реалните числа;
- В световната област обикновено се въвежда дясноориентирана декартова координатна система.

Схема на Световната област (WD)



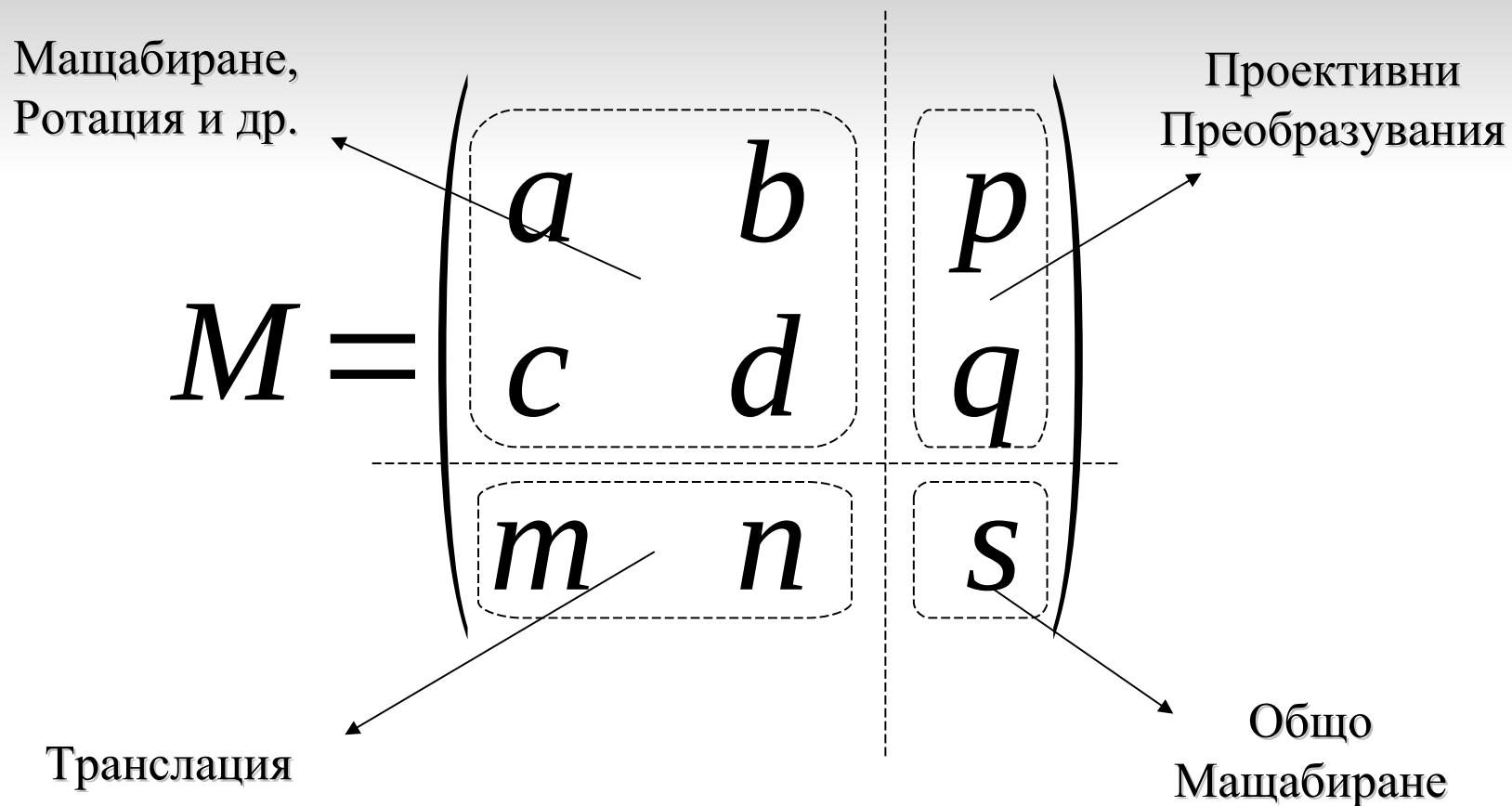
Световна област (WD)

- Описанието (именуването) на точка или вектор в WD се реализира декартови координати;
- Използват хомогенни координати за описание на точките и векторите;
- Геометричните преобразования се реализират като умножение на матрици.

Хомогенни координати

- Точка в двумерната световна област с декартови координати (x, y) се описва с хомогенни координати, чрез тримерният вектор-ред (X, Y, H) , където $H \neq 0$;
- От декартови в хомогенни координати се преминава чрез съответствието $X=x, Y=y, H=1$, а преходът от хомогенни в декартови става, чрез съответствието $x=X/H, y=Y/H$, при положение, че $H \neq 0$;
- Вектор в двумерната световна област с декартови координати (V_x, V_y) се описва с хомогенни координати, чрез тримерният вектор-ред $(V_x, V_y, 0)$.

Геометрични Преобразования (1/2)



Геометрични Преобразования

(2/2)

$$P = (X \quad Y \quad H)$$

$$P \cdot M = P'$$

$$P = P' \cdot M^{-1}$$

$$M_1 \cdot M_2 \neq M_2 \cdot M_1$$

Пример – Без преобразование

$$M_I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Пример – Транслация

$$M_{T(dx, dy)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dx & dy & 1 \end{pmatrix}$$

Пример – Ротация

$$M_{R(\alpha)} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Пример – Масштабиране

$$M_{S(u,v)} = \begin{pmatrix} u & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Пример – Общо мащабиране

$$M_{S(s)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/s \end{pmatrix}$$

Пример – Накланяне по x

$$M_{Sx(\lambda)} = \begin{pmatrix} 1 & \lambda & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Пример – Накланяне по у

$$M_{Sy(\lambda)} = \begin{pmatrix} 1 & 0 & 0 \\ \lambda & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Пример – Огледално отр. по x

$$M_{Mx} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Пример – Огледално отр. по у

$$M_{My} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Трансформации в п-вото (3D)

В тримерното пространство аналогично се използват матрици 4×4 . По подобен начин могат да се запишат матриците за основните прости трансформации.

Геометрични Преобразования в 3D

$$M = \begin{pmatrix} a & b & c & p \\ d & e & f & q \\ g & h & i & r \\ l & m & n & s \end{pmatrix}$$

Проективна трансформация (едноточкова перспектива по Z)

$$M_{Pz(p)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1/p \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Обекти и Операции

Операциите с геометрични обекти могат да бъдат групирани в следните няколко класа:

- Генерация на обект;
- Информация за обект;
- Векторна алгебра;
- Геометрични преобразования;
- Координатни трансформации;
- Класификация на точкови множества;
- Изчисляване на сечения на точкови множества.

Структурен процесор

Работата с изображение само чрез примитиви, не е подходяща за оператора. Поради тази причина се въвежда виртуална машина Структурен процесор. Неговата функция е поддържането **структурата** на изображението.

Структурен процесор

Поддържането на структурата означава възможност за установяване или премахване на отношения между елементи на изображението.

Това е процес на формиране на състав на система от примитивни елементи, чрез установяване на отношения.

СтП се занимава с

- Вида на отношенията, чрез които се формират структурните елементи на изображението;
- Допустимите отношения между структурните елементи на изображението;
- Откриването на указан елемент;
- Работата с Прозорците.

Структурни Елементи

- Изображението е съвкупност от графични примитиви с различни визуализационни характеристики;
- Определянето на структуриран елемент от изображението обикновено става чрез определянето на някакво подмножество от примитиви;
- Структурни елементи.

Основни операции

- Добавяне на елемент към структурен елемент;
- Отстраняване на елемент от структурен елемент.

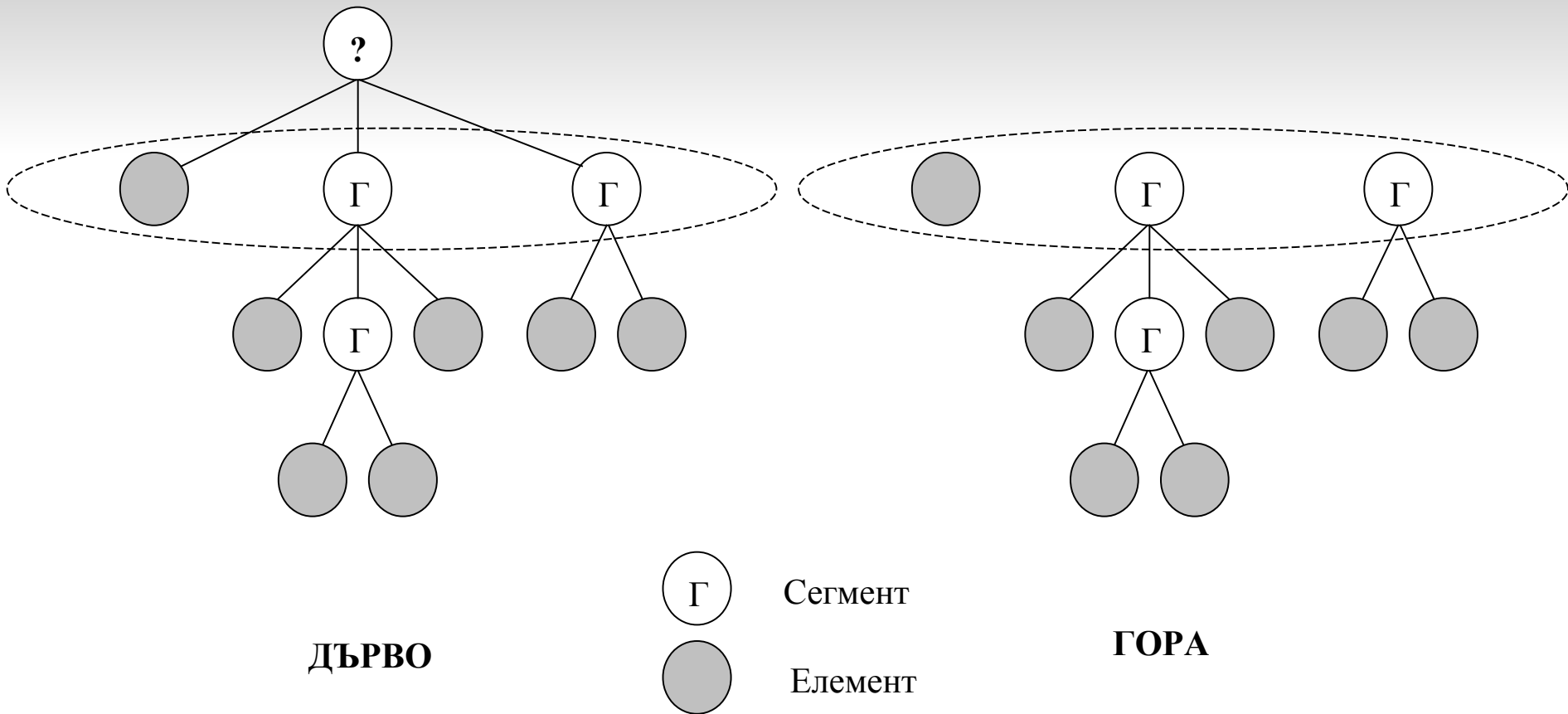
Йерархична структура

Системите за работа с графична информация трябва да дават възможност, структурата на изображението да бъде **йерархична**, с произволен брой нива.

Йерархична структура

- В дисплейният файл може да се зададе йерархия с много нива, като се допуска изпълнение на символ от символ;
- Целта на поддържане на йерархия е лесно указване и работа със сложни (съставни) части от изображението;
- Йерархията може лесно да се поддържа в структурата от данни (модела), и това означава, че не е необходимо явното й поддържане в дисплейният файл.

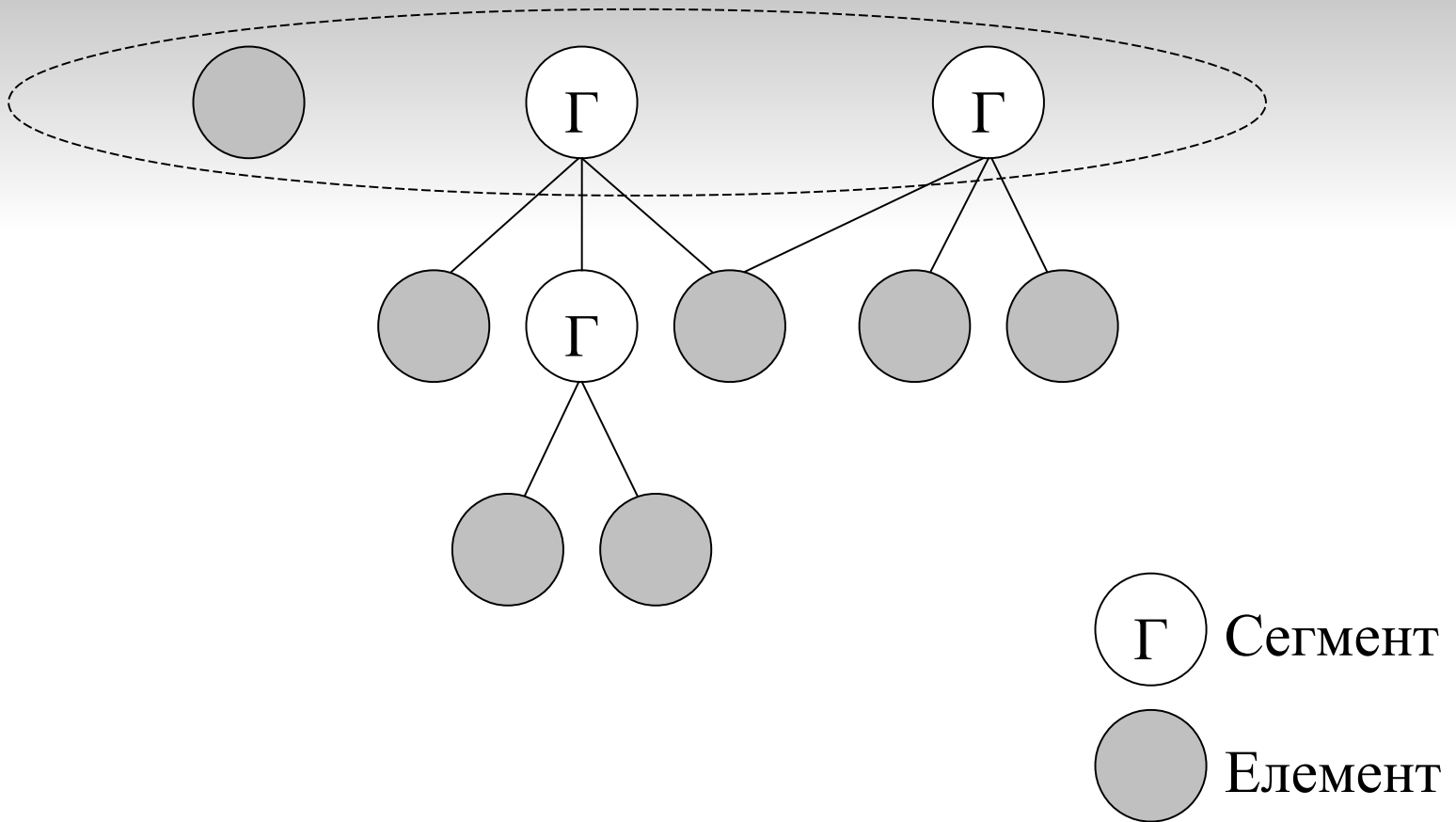
Често използвани структури от данни



Йерархична структура

- Сегмент – именувана съвкупност от елементи и сегменти;
- Понятието сегмент предполага йерархична структура на данните, т. е. **Дърво**;
- Понякога се допуска един сегмент да принадлежи на няколко елемента и тогава структурата е **Ацикличен Граф**.
Затова в КГ много често се използва термина **Граф на сцената**.

Ацикличен Граф



Рамка (1/2)

За да се реализира визуализация и манипулация с изображение, трябва да се дефинира прозорец в WD и отворстие на гледане в SD .

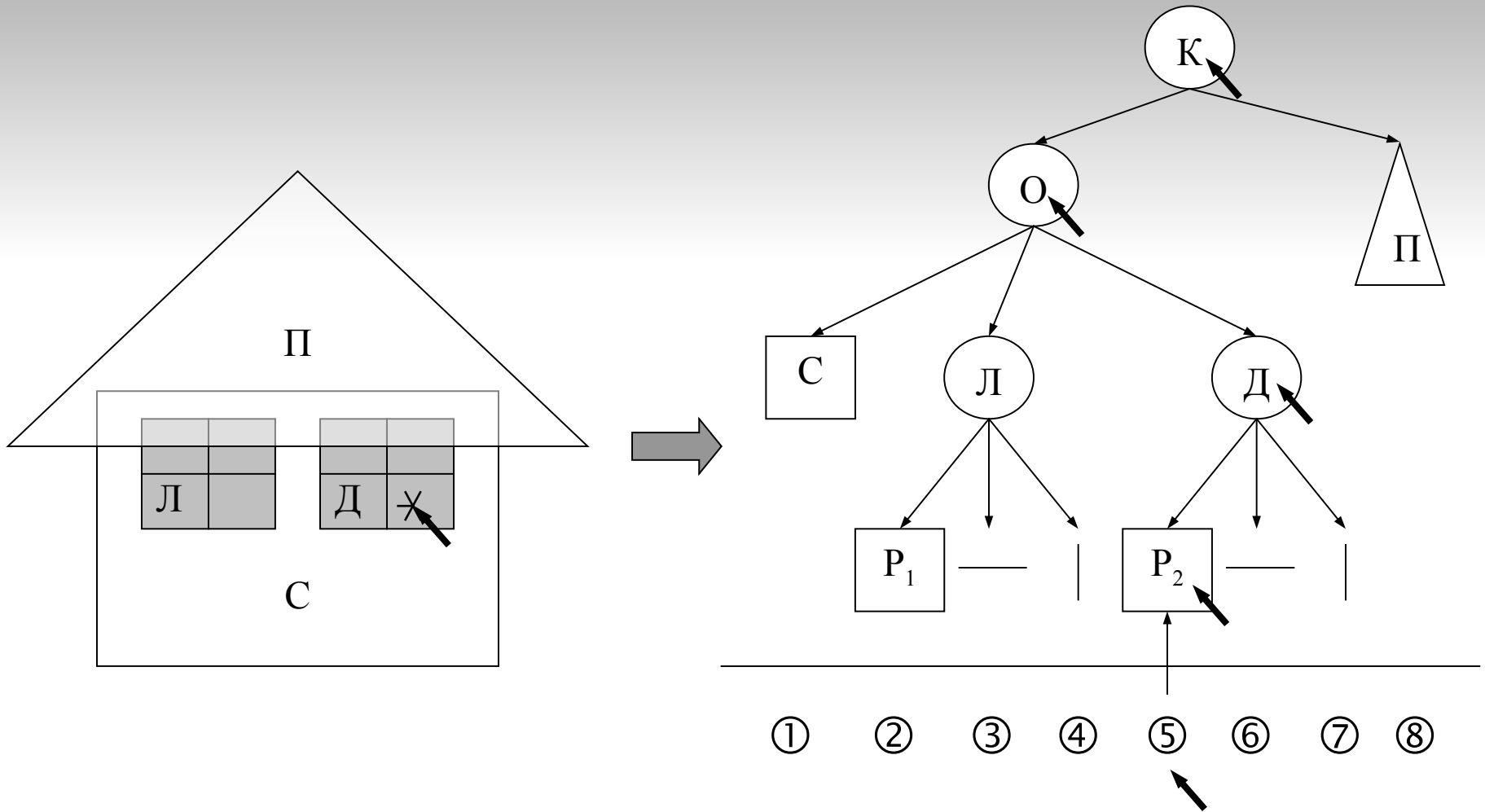
Параметрите определящи налагането на двата прозореца един върху друг е даннов обект, наречен **Рамка**.

Рамка (2/2)

Структурният процесор поддържа съвкупност от рамки, като винаги една от тях е активна и е свързана със структурата от данни описваща изображението (модела).

Чрез активната рамка се извършва манипулацията с модела. Рамките са обект на Диалоговият процесор.

Указване на елемент



Семантичен процесор

Системите за работа с графична информация се използват за създаване на графични модели (изображения) на някакви реални процеси или явления.

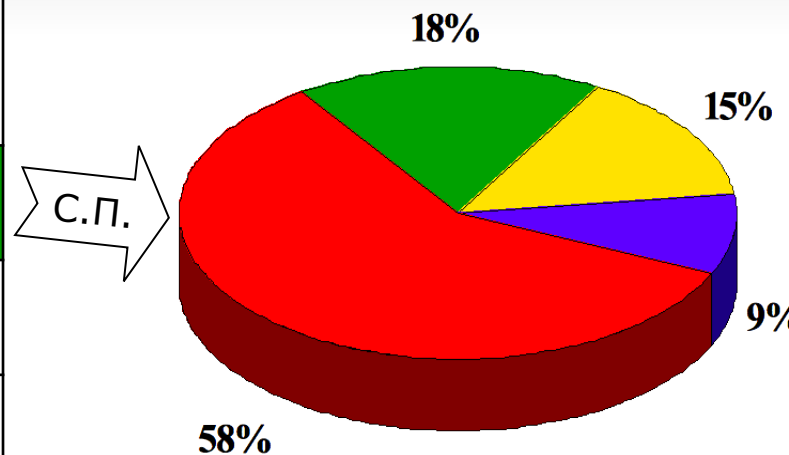
Необходими са алгоритми за получаване на изображението на базата на някакви други данни, тогава възможностите за манипулация с изображението ще имат спомагателен характер.

Семантичен процесор

Негова основна задача е да подпомага изграждането на **не графични модели** и да преобразува тези модели в **графични изображения**.

Пример

	1-во трим.	2-ро трим.	3-то трим.	4-то трим.
Изток	20	12	80	24
Запад	30	40	70	40
Север	45	45	90	80



Диалогов процесор

От диалоговият процесор в най-голяма степен зависи успешното протичане на диалогът човек-система.

Диалогов процесор

За осъществяването на диалог е необходима
Права и Обратна връзка.

Обратната връзка се изразява в предаване на команди към системата, в резултат на чието изпълнение, по правата връзка се получава изображение.

От гледна точка на оператора може да се каже, че правата връзка е предаването на командите, а визуализацията – обратната.

Диалогов процесор

- Устройства за диалог;
- Събитие;
- Рамка.

Устройства за диалог

- Множество от диалогови устройства, с които оператора разполага;
- Характеризират се със **състояние**, което оператора **може да променя**.

Устройства за диалог

От това как промяната на състоянието се интерпретира от системата се делят на:

- **Пасивни** – могат да бъдат в едно от множество равноправни състояния;
- **Активни** – могат да бъдат в едно от две неравноправни състояния.

Пространство на състоянията

Множеството от състояния, в които може да бъде едно устройство се определя като **пространство на състоянията** на устройството и за Диалоговият процесор са важни пространствата на състоянията на устройствата за диалог.

Видове пространства на състоянията

- **Структурирани** – определят се като декартово произведение на пространства на състояния

Например мишката има структурирано пространство;

- **Неструктурирани** – биват дискретни и скаларни

Например дискретно пространство е пространство от състояния на клавиш, а пример за скаларно е пространство от състояния на аналогово-цифров преобразувател.

Диалогов процесор

Основната функция на
Диалоговият процесор
е адекватна реакция на състоянията на
активните устройства за диалог, като се
отчитат състоянията на пасивните
устройства.

СЪБИТИЯ

Събитие е промяна на състоянието на активно устройство.

На всяко събитие трябва да има незабавна реакция от страна на диалоговият процесор.

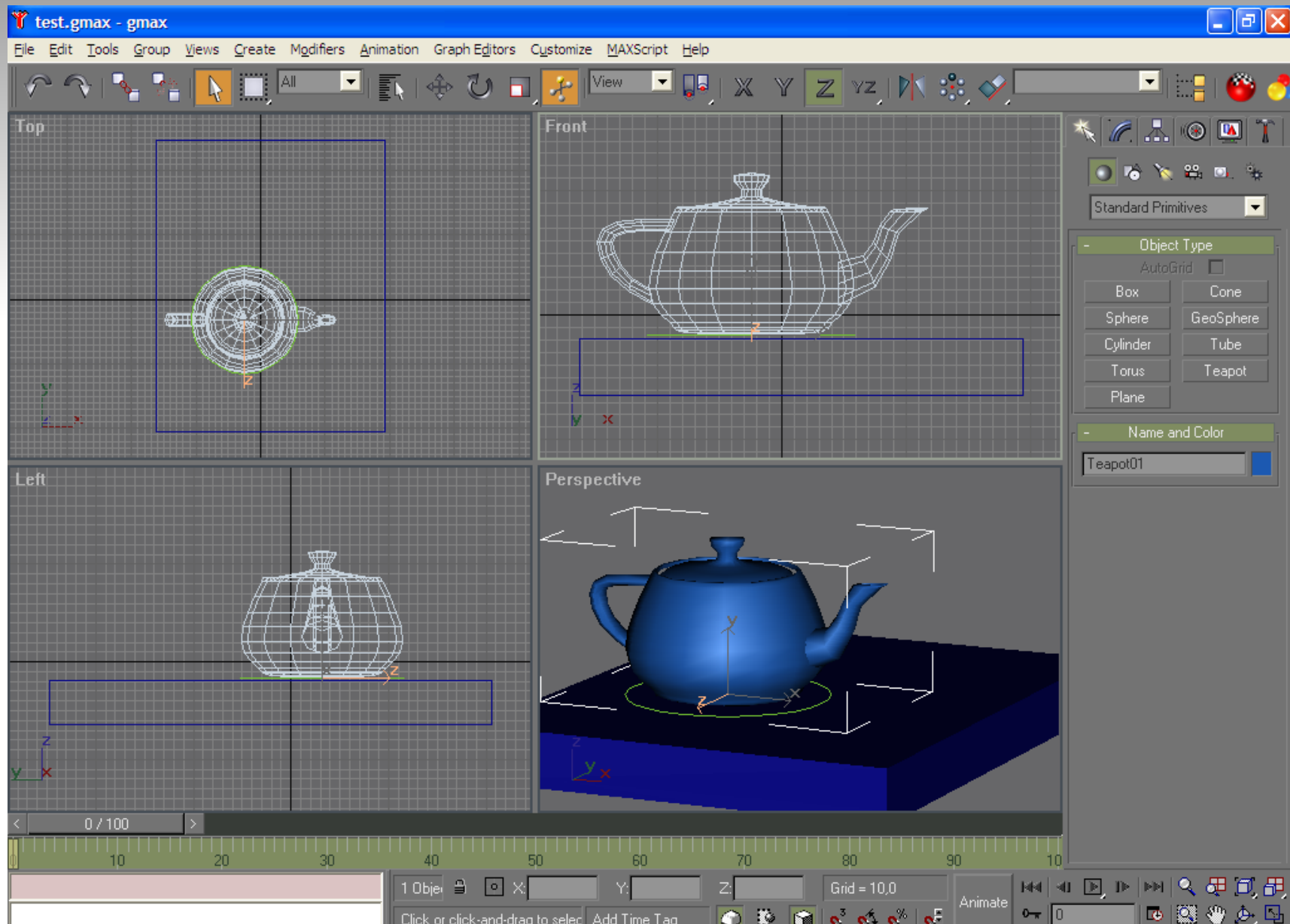
Рамки

Как се установява указана рамка?

Върху екрана на дисплея може да бъдат изобразени повече от едно отвори на гледане, всяко от които определя еднозначно една рамка.

Терминът “Прозорец” се е наложил за обозначаване на отворието на гледане.

Пример



Рамки

Във всеки един момент, един от прозорците е активен и само чрез него може да се извършва манипулация с изображението (модела).

Това се отразява в структурата от данни на изображението, като активният прозорец е първи в списъка с рамки.

Графичен интерфейс

Всяка конкретна реализация на
Графичен Потребителски Интерфейс
(ГПИ/GUI)

е една конкретна специализирана система за
работа с Графична информация.

Графичен интерфейс

Системите за работа с прозорци са системи, реализиращи графичен интерфейс.

Графичен интерфейс

Структурата на прозореца е изградена от (тези области може да се препокриват):

- Полета (области) за начални данни;
- Полета (области) за резултати;
- Изображения (пиктограми, икони), символизиращи различни обекти, в частност команди;
- Като елементи на прозореца могат да участват и други под-прозорци, т. е. има възможност за йерархия.

Графичен интерфейс

Графичен интерфейс – характеристики:

- Възможност за комуникация чрез прозорец;
- Възможност за задаване на команди чрез указване на пиктограма;
- Възможност за манипулация с прозорец.

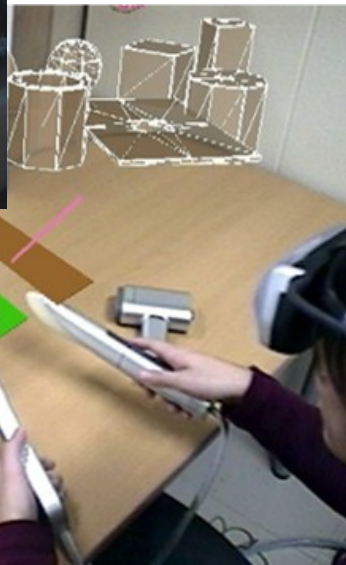
WIMP

- Windows – Прозорците съдържат изпълняващи се паралелно (или псевно-паралелно) програми, изолирани визуално една от друга;
- Icons – Иконите са средство за бърз визуален достъп (изпълнение) на определена операция, задача или друга програма;
- Menus – Менюта са текстово или графично средство за избор на операции, задачи и др.;
- Pointer – Указателя е визуален екранен елемент/ символ, контролиран от физическите движения на указващо входно устройство („мишка“).

Post-WIMP

- Windows – Прозорци
- Icons – Икони
- Menus – Менюта
- Point – Посочване





а ГПИ

Виртуални машини

Въпроси?

