



Компютърна Графика

Генеративна Компютърна графика.
Методология за Графично моделиране.
Модел на диалогова графична система.

доц. д-р Александър Пенев

Генеративна КГ

Генеративната КГ се занимава с:

- ❖ Построяване на обекти (модели);
- ❖ Генерация на изображение;
- ❖ Преобразуване на обекти и изображения;
- ❖ Идентификация на обекти.

Обща постановка

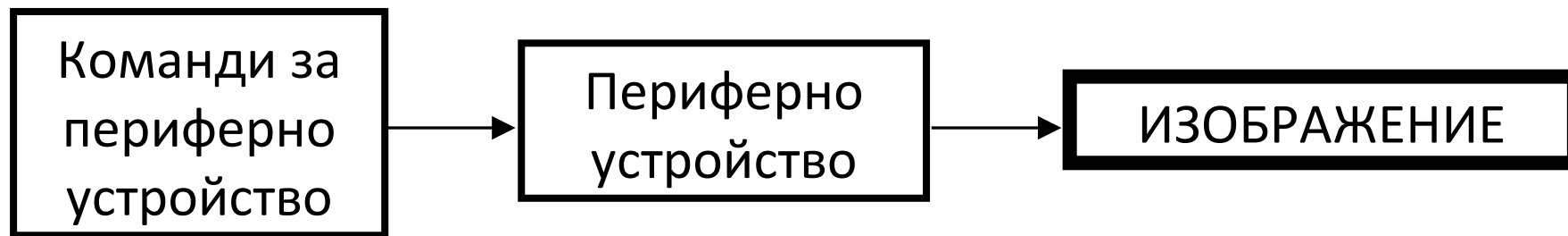
Генеративната компютърна графика се занимава
принципно с производство на изображения
(картини) чрез компютър.

1. Изображение Създадено от Команди

1. Изображение създадено от команди

Под производство на изображения се разбира
някаква последователност от команди за
управление на някакво периферно устройство,
като всяка команда причинява добре
дефинирана промяна в състоянието на
периферното устройство.

Изображение създадено от команди



Изображение създадено от команди

Понятието изображение може да бъде дефинирано по два начина:

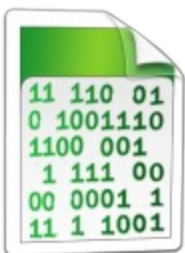
- ❖ Като реална (физическа) картина;
- ❖ Като последователност от команди на конкретно периферно устройство.

Изображение създадено от команди

Последователност от команди на периферно устройство се обозначава с термина **представяне на изображение** или **програма за периферно устройство**

Човекът-оператор сравнява изображението създадено от него с абстрактният модел и при откриване на несъответствие променя командите.

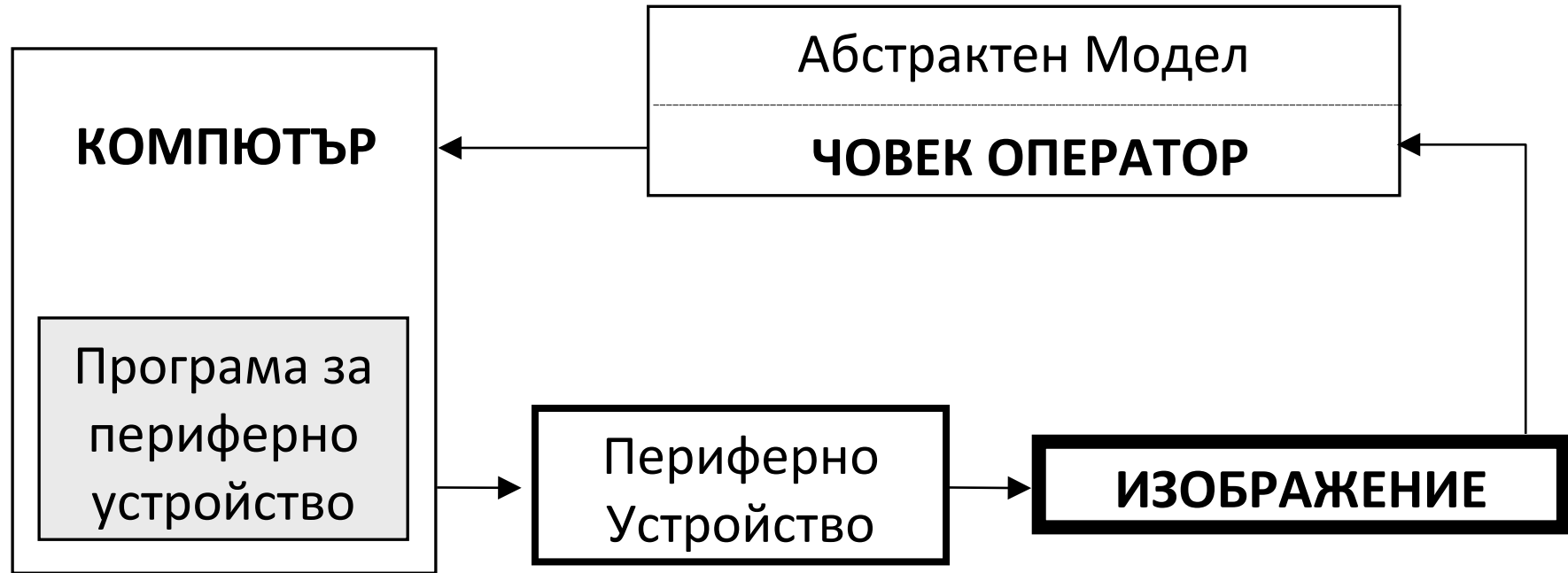
Пример



Изображение създадено от команди

Човекът-оператор трябва да е
Системен Програмист

Работа с изображение създадено от команди

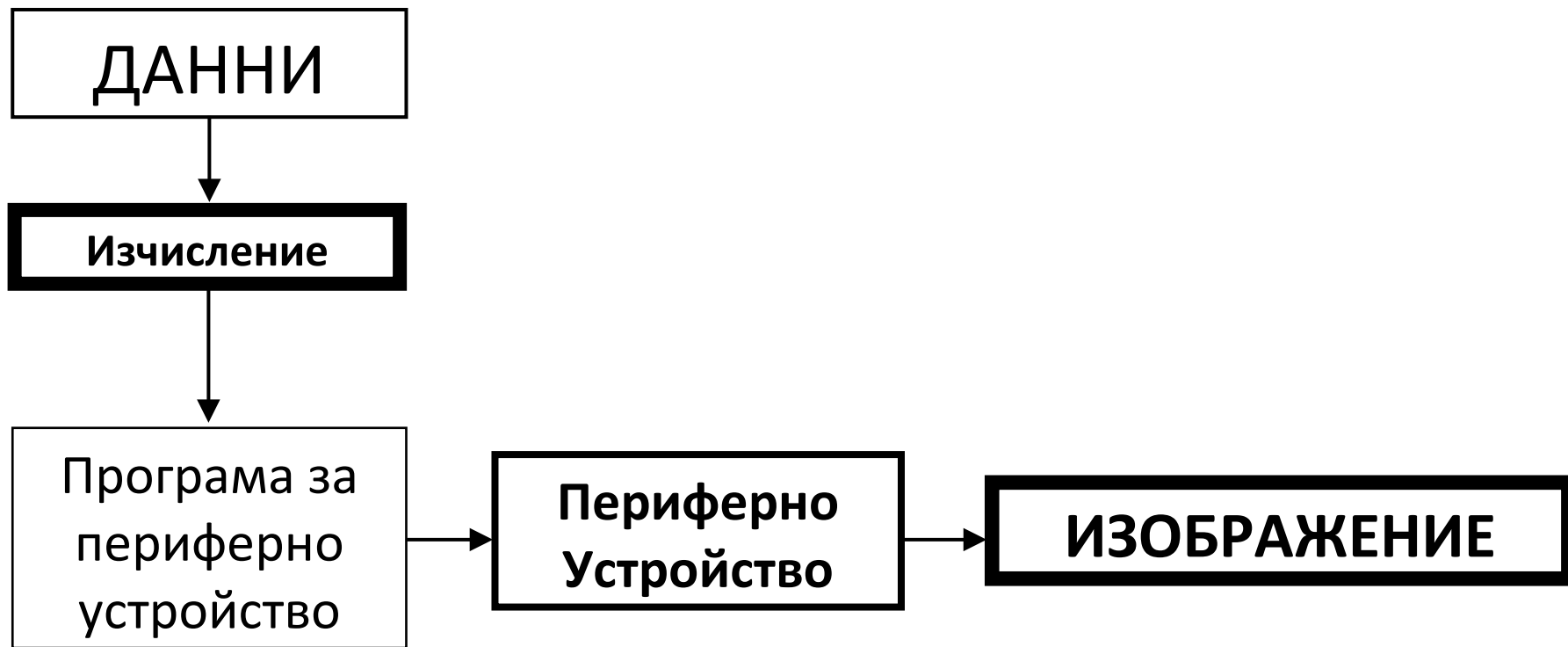


2. Изображение, създадено чрез „Изчисляване“

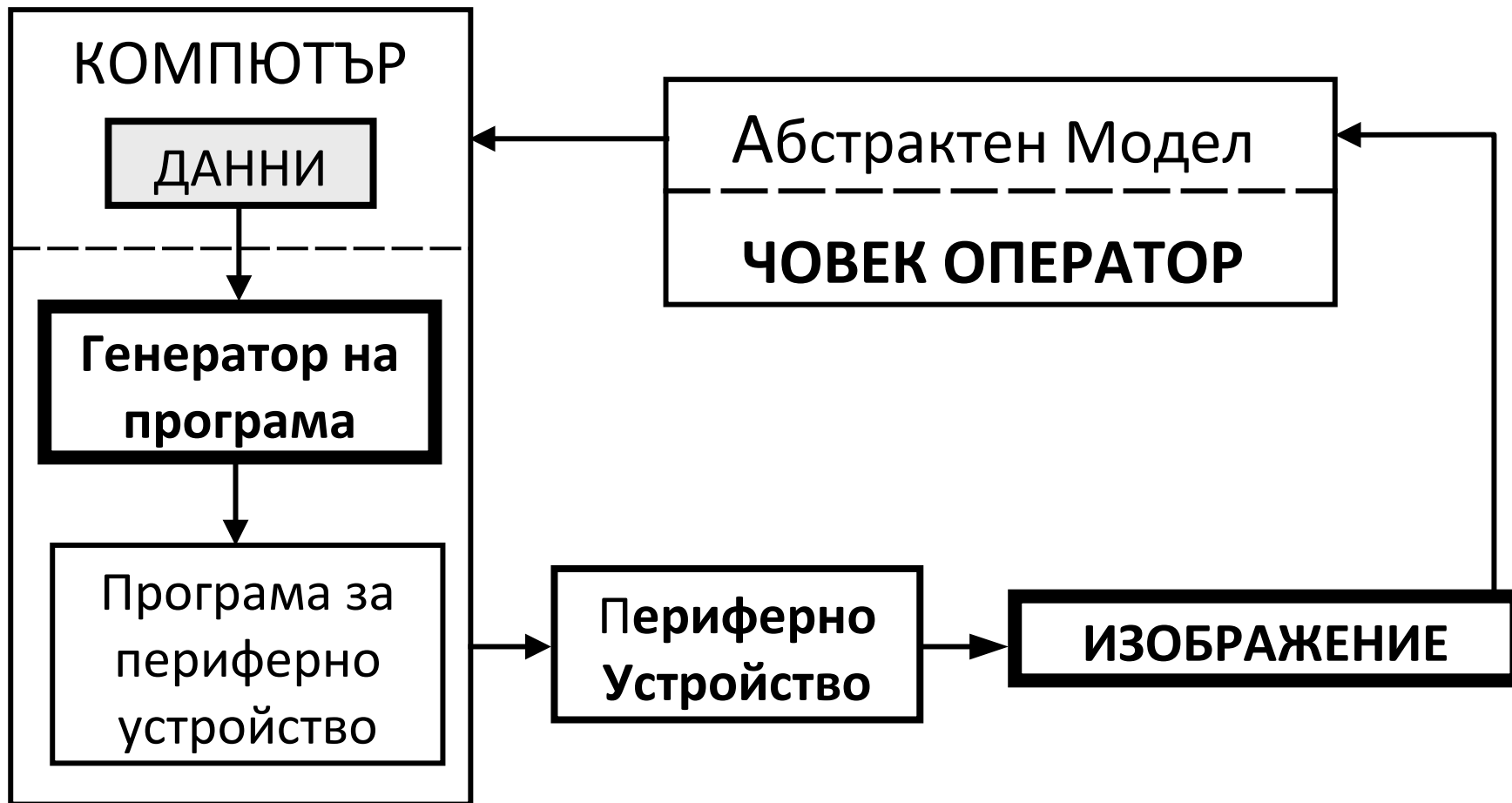
2. Изображение, създадено чрез “Изчисляване”

Под производство на **изображения** се разбира някакъв **алгоритъм** (изчисление), който обработва някаква **структура от данни** (модел) и произвежда команди за управление на **периферното устройство** .

Изображение, създадено чрез “Изчисляване”



Изображение, създадено чрез “Изчисляване”



Пример

```
void main()  
{  
  int a, b;  
  a = 5;  
  b = 6;  
}
```



```
11 110 01  
0 1001110  
1100 001  
1 111 00  
00 0001 1  
11 1 1001
```



+данни

```
void main()  
{  
  int a, b;  
  a = 5;  
  b = 6;  
}
```



```
start page  
color blue  
move 10,10  
line 20,90  
show page
```



+данни

Изображение, създадено чрез “Изчисляване”

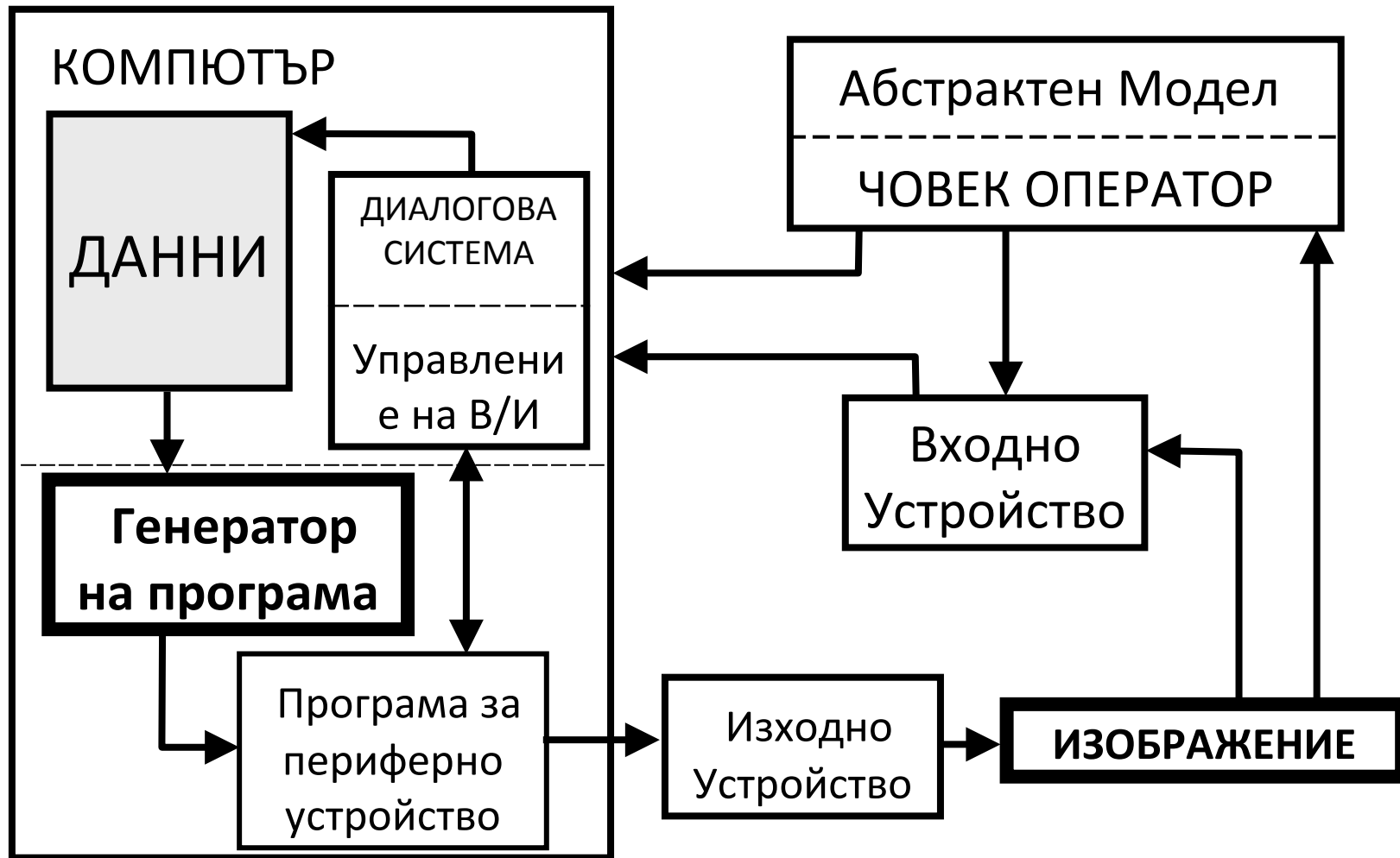
Човекът-оператор трябва да е
Програмист

Изображение, създадено чрез “Изчисляване”

Тази схема за получаване на изображение чрез компютър е характерна за средния период на развитие на генеративната Компютърна графика

3. Диалогова Работа с Изображение

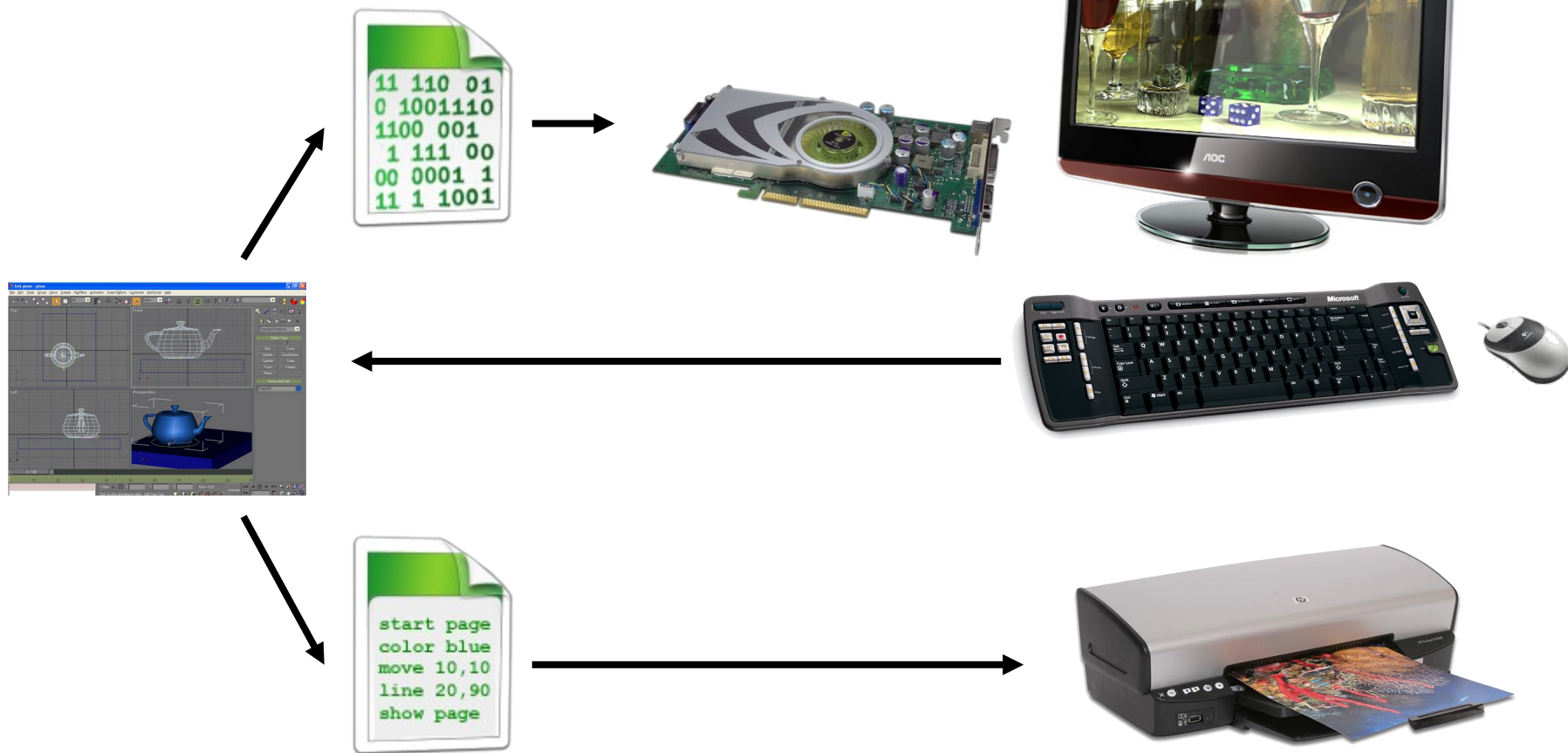
3. Диалогова работа с Изображение



Особености на Подхода

- ❖ Постъпково изграждане на изображението (интерактивност);
- ❖ Изображението не е само краен продукт, а е и свързващо звено между две последователни действия;
- ❖ Възможност за указване на елемент от изображението и разпознаване му, като елемент от структура от данни (модела);
- ❖ Средства за работа със структурни изображения;
- ❖ Поддържане на връзки между имена и вътрешно представяне на изображението;
- ❖ Присъствие на В/И устройство, което да произвежда изображения и да предоставя средства за указване на части от изображението.

Пример



Диалогова работа с Изображение

Човекът-оператор **НЕ** трябва да е
Програмист

*Това е съвременното състояние на развитие на системите
за геометрично моделиране.*

Графично Моделиране

Графично/Геометрично моделиране

Теории, методи и системи, насочени към създаване на информационно пълни представяния на двумерни/тримерни реални обекти, които дават възможност да се изчисли всяко добре определено геометрично свойство на обектите, които те описват.

Изводи от Особенностите при Подхода в Графичното Моделиране

Графично моделиране

Крайният продукт и фундаменталното свързващо звено в
Компютърната графика е **Изображението**.

Графично моделиране

Трябва да се поддържа йерархия от представяния поради:

- ❖ Специфичните особености на абстрактният модел, който трябва да бъде реализиран, като изображение;
- ❖ Възможностите на компютъра за представяне и обработка на графична информация.

Изводи

- ❖ Може да се поддържа само едно вътрешно представяне на изображението, обаче, операции с изображението са многообразни, и дефинирането им в термините само на едно вътрешно представяне усложнява реализацията;
- ❖ Дисплейните системи са твърде разнообразни, следователно изображение е добре да бъде дефинирано в термините на виртуални технически устройства;
- ❖ В абстрактният модел на оператора изображението се разглежда, като йерархия от съставни графични елементи;
- ❖ В средствата за манипулация с изображението трябва да бъде отразена в някаква разумна степен и системата от понятия на оператора;

- ❖ Йерархията от представяния е задължителна и между тях трябва да съществуват програмни или хардуерни конвертори, преобразуващи помежду им;
- ❖ Всяко представяне е нов, по-конкретен модел на определен аспект на абстрактният модел на оператора от гледна точка на възможностите за визуализиране.

Методология за Графично Моделиране

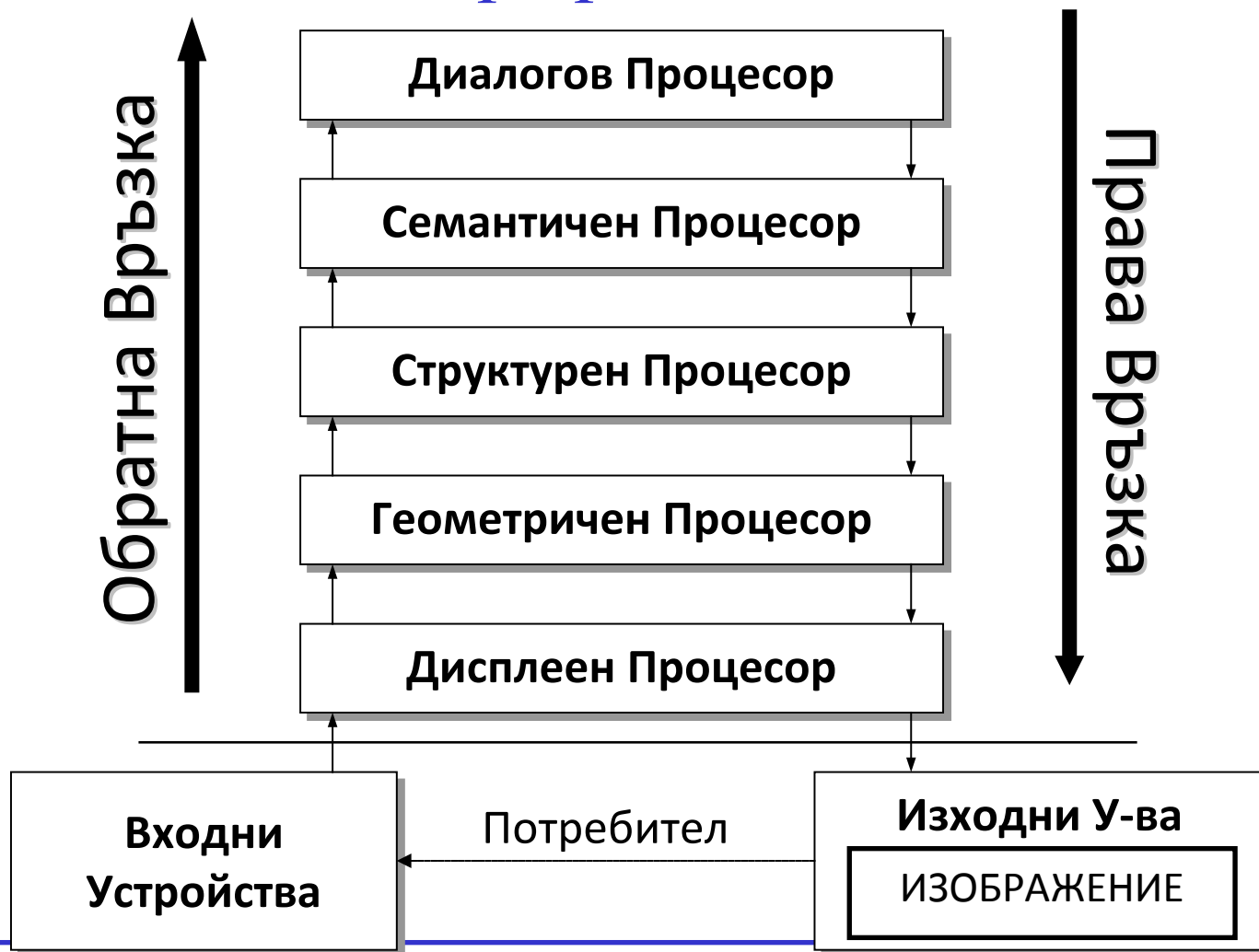
Методология за Графично Моделиране

Всяка система за работа с графична/геометрична информация представлява йерархична система от **Виртуални Машини**.

Системата трябва да поддържа информация за:

- ❖ Визуализационните характеристики на елементите, от които е съставено изображението;
- ❖ Геометричните характеристики на елементите, от които е съставено изображението, такива като форма, размери, местоположение и ориентация в пространството за визуализация;
- ❖ Структурните характеристики на изображението, т. е. йерархията от под изображения;
- ❖ Семантичните характеристики на изображението, т. е. евентуалните имена, чрез които операторът е именувал различни части от изображението, както и други невизуални характеристики, свързани отделните части на модела.

Модел на диалогова графична система



Методология за Графично Моделиране

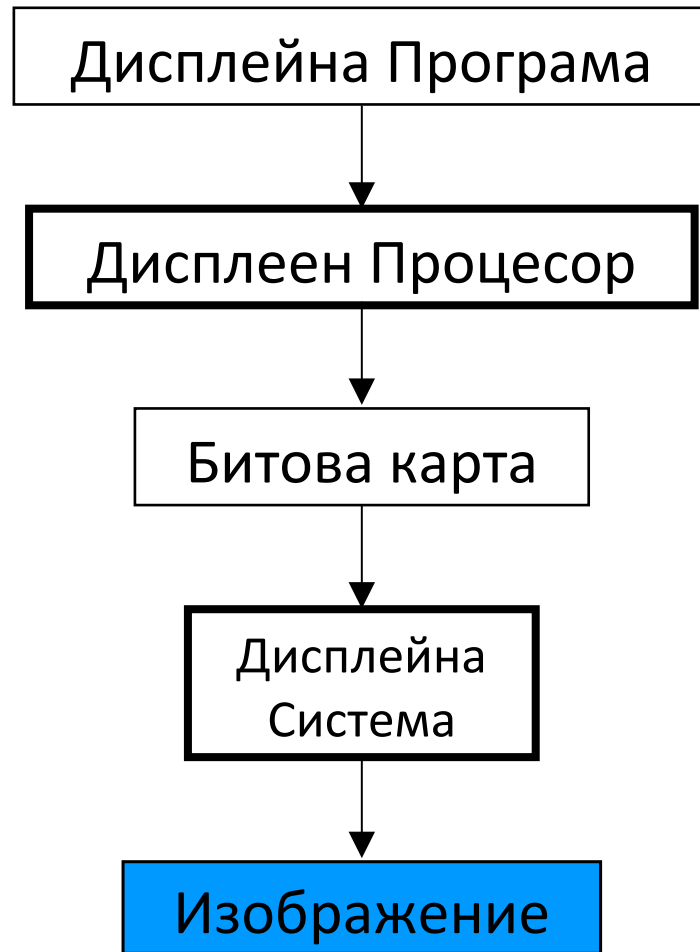
Съвкупността от тези виртуални машини и отношения между тях,
както и начина на реализацията им определят
Методологията за Графично Моделиране.

Дисплеен Процесор

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.

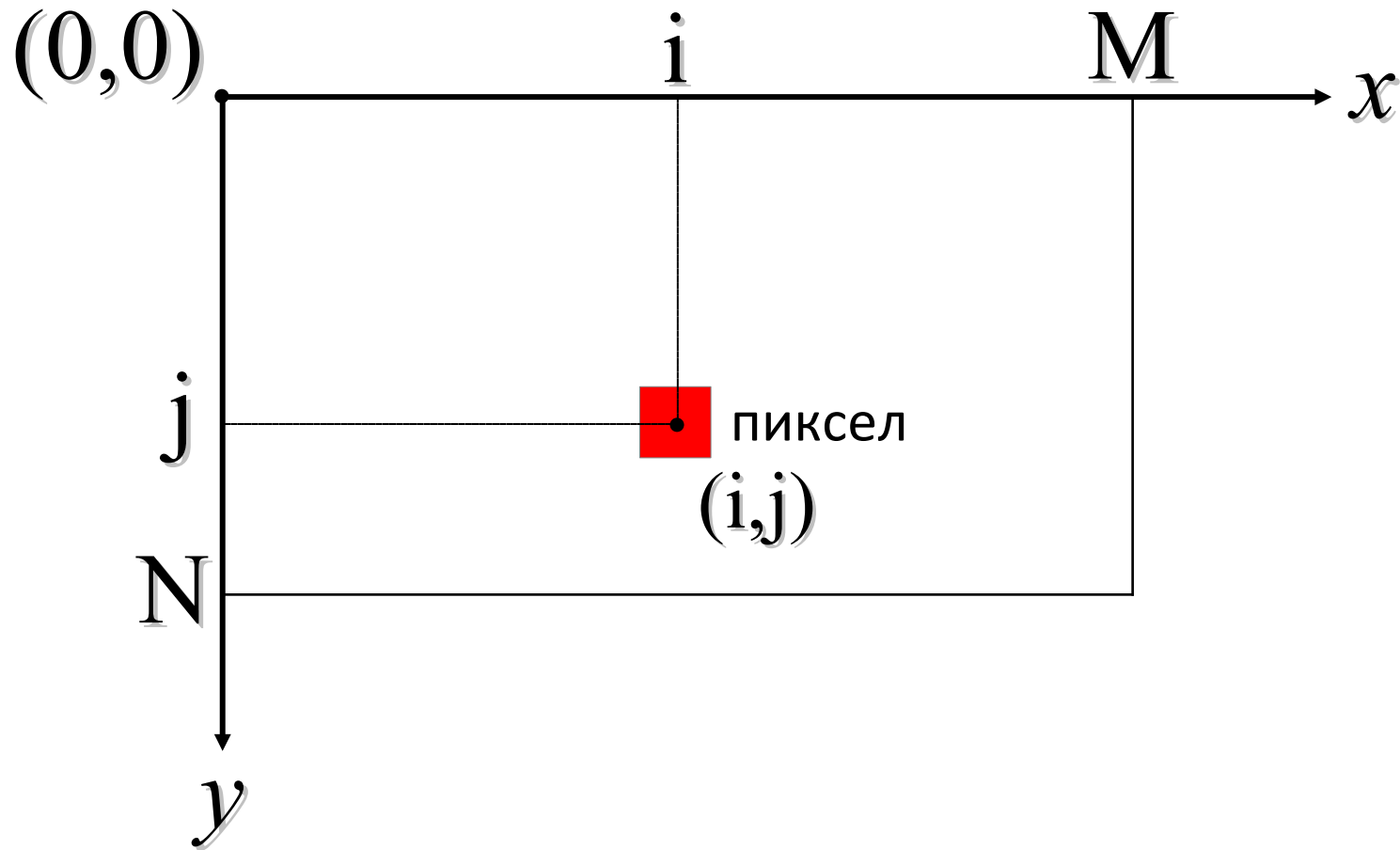
Екранна област (SD)

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

- ❖ 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

- ❖ Тъй като в някои от първите технологии за монитори, лъча обхожда пикселите на екрана от ляво на дясно и отгоре надолу, то мястото на всеки пиксел битовата карта се определя по формулата (тя се е запазила и в по-новите устройства):

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

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

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

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.

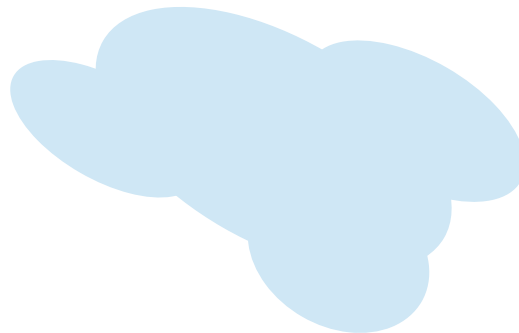
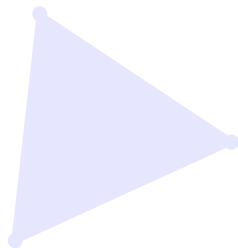
Най-често графичните примитиви биват:

- ❖ 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, инициирал 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

Геометричен процесор

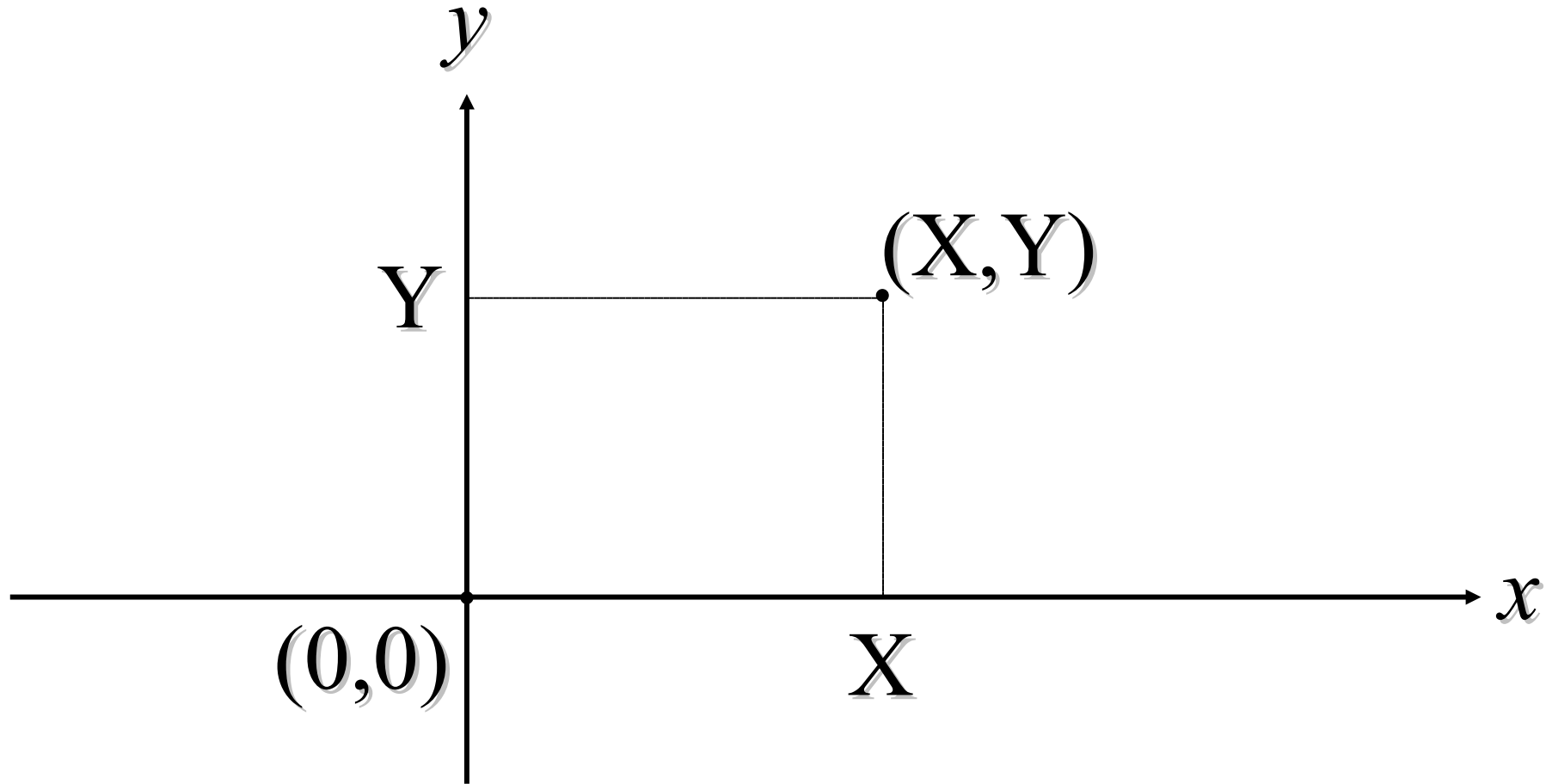
Основното предназначение на Геометричният процесор е реализиране на геометрични преобразувания с изображението.

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 $WD=R \times R$, където с R се обозначава множеството на реалните числа;
- ❖ В световната област обикновено се въвежда дясноориентирана декартова координатна система.

Схема на Световната област (WD)



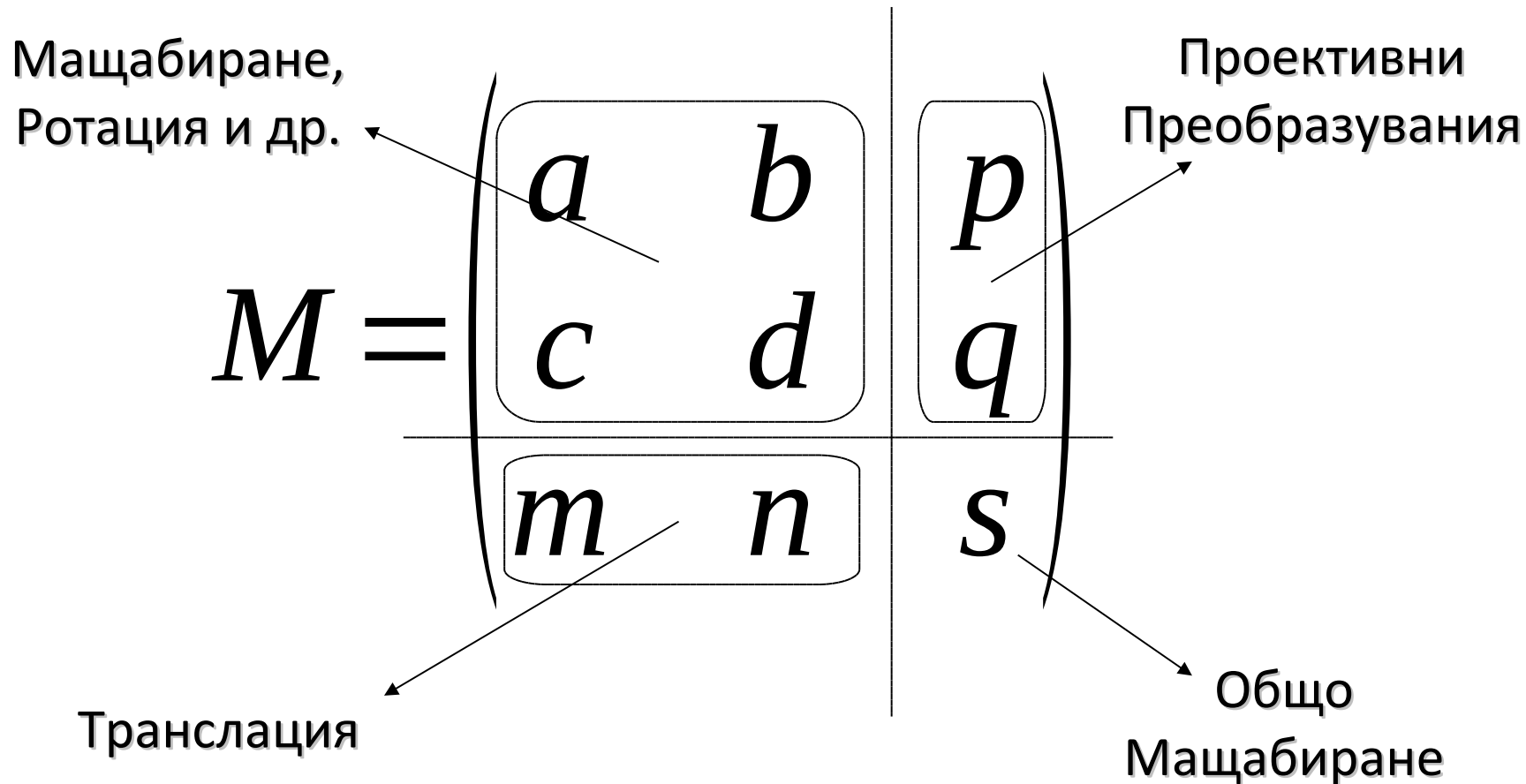
Световна област (WD)

- ❖ 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



Геометрични Преобразования

$$P = \begin{pmatrix} X & Y & H \end{pmatrix}$$

$$P \cdot M = P'$$

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

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

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 – Мащабиране

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

Пример – „Приплъзване/Наклон“ по x

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

Пример – „Приплъзване/Наклон“ по y

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

Пример – Огледално отр. по x

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

Пример – Огледално отр. по y

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

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

Структурен процесор

Работата с изображение базирано само на 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.

Структурният процесор 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(и) към структурен елемент;
- ❖ Отстраняване на елемент(и) от структурен елемент.

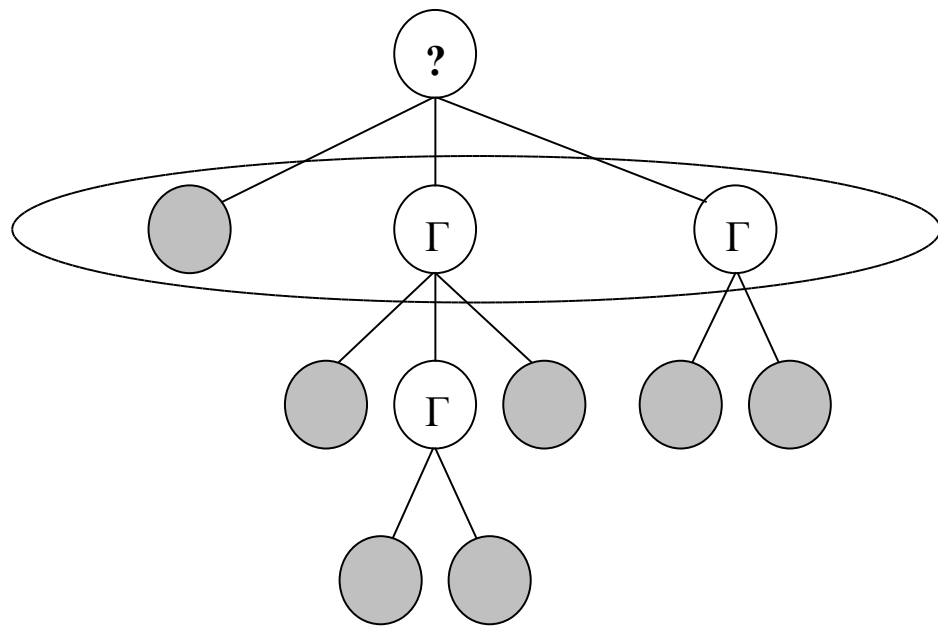
Йерархична структура

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



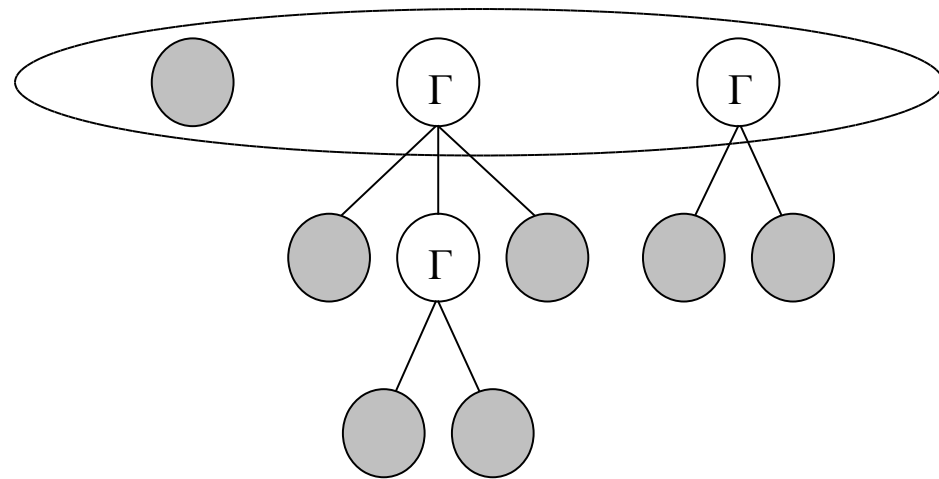
ДЪРВО



Сегмент



Елемент

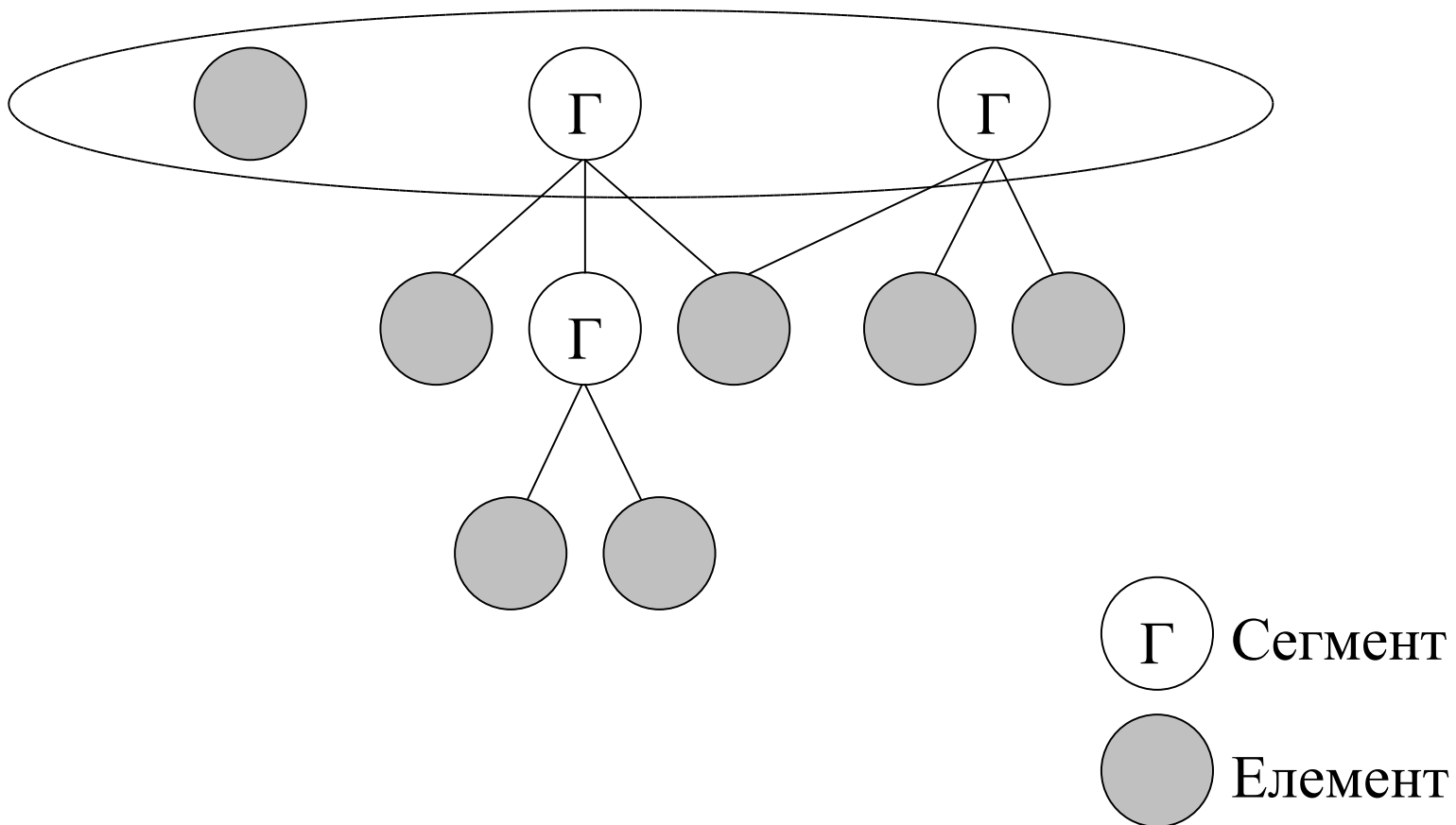


ГОРА

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

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

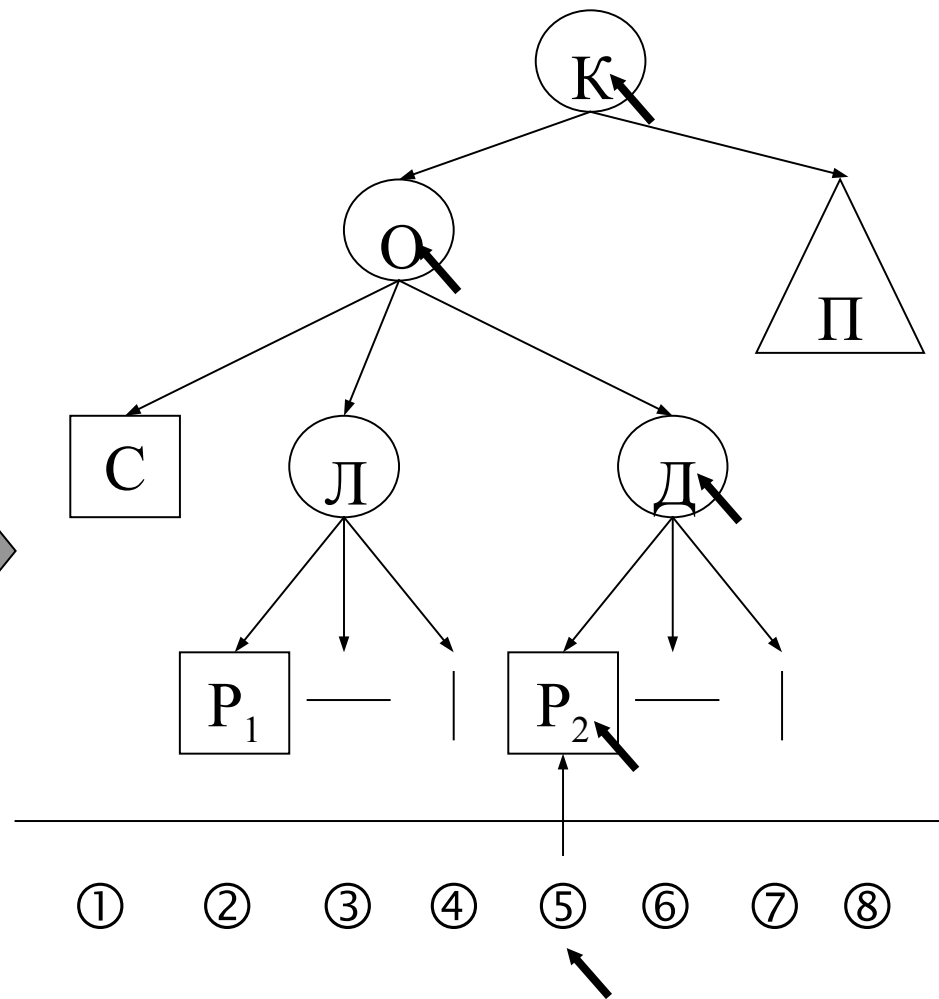
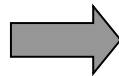
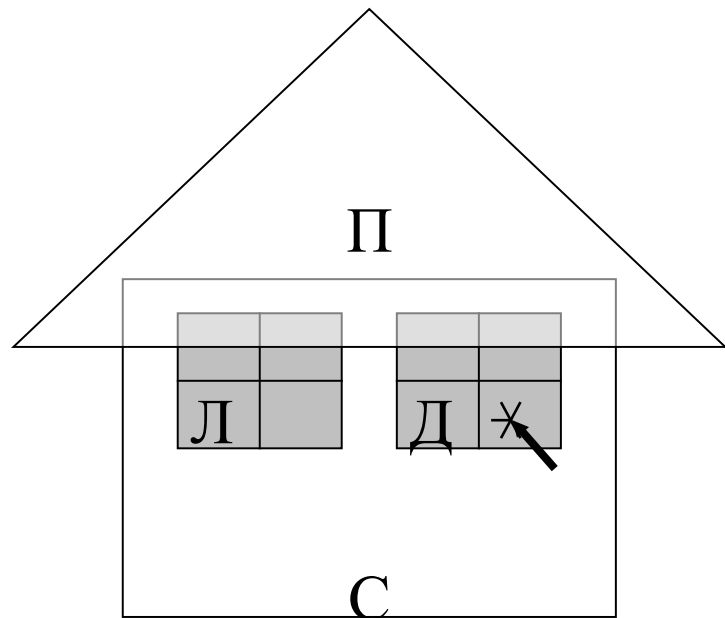
Параметрите определящи налагането на двата прозореца един върху друг е даннов обект, наречен **Рамка**.

Рамка

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

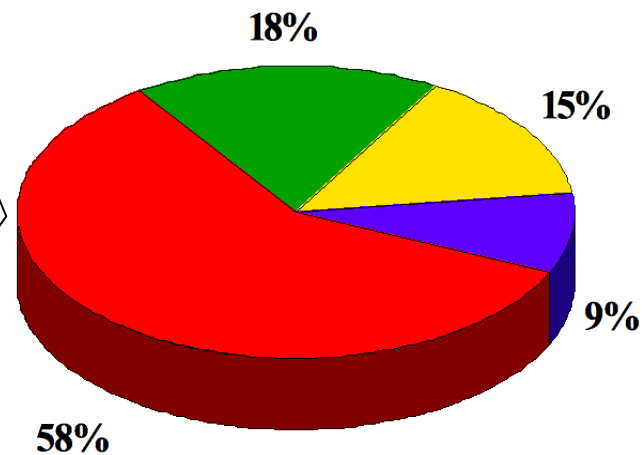
Семантичен процесор

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

Семантичният процесор прави и преобразования запазващи („визуалната“) семантиката на моделите.

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

Диалогов процесор

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

Диалогов процесор

За осъществяването на диалог е необходима
Правя 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.

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.

Принципно действие на Диалоговия процесор

$$\text{ДиалП}(e, i, M) \rightarrow i', M'$$

Където:

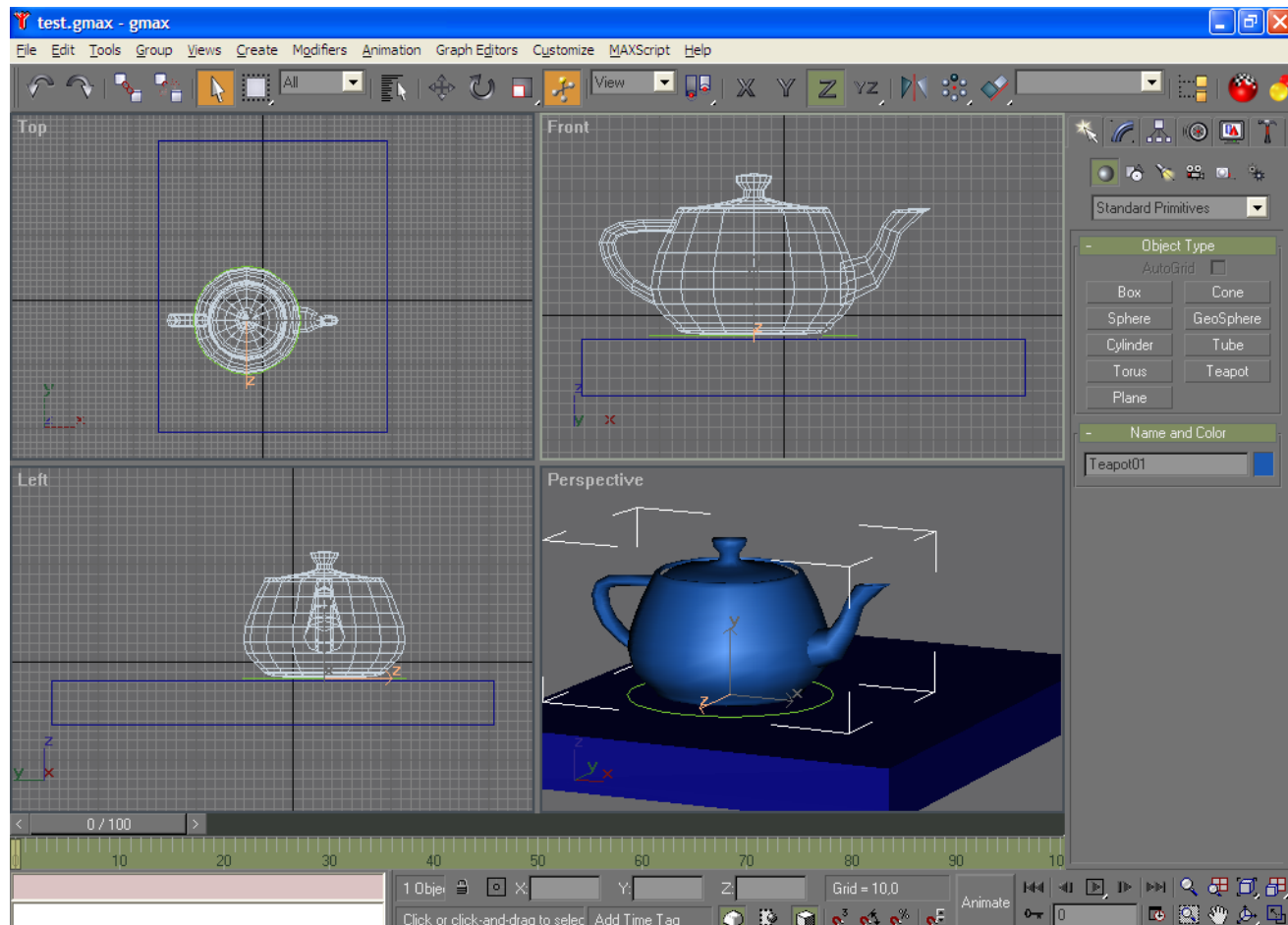
e = събитие

i = състояние на диалога

M = модел

Всяка конкретна реализация на
Графичен Потребителски Интерфейс
е една конкретна
специализирана система за работа с
Графична информация.

Пример



Въпроси?

apenev@uni-plovdiv.bg

