


# Компьютерная геометрия и графика



---

## Лекция 1.

## Введение в компьютерную графику

Лектор: Юлия Александровна Болотова



---

# План работы в семестре



# Работа в семестре

---

[ftp://ftp.vt.tpu.ru/study/Bolotova/public/computer\\_graphics](ftp://ftp.vt.tpu.ru/study/Bolotova/public/computer_graphics)

- Лекции.
- Лабораторные работы.
- Самостоятельная работа.
- Реферат.
- 2 коллоквиума.
- Экзамен.



# Компьютерная графика (лк)

---

1. Растровая и векторная графика.
2. Цветовые модели. Принципы зрительного восприятия.
3. Растровые алгоритмы.

**Контрольная работа №1**



# Компьютерная геометрия

---

4. Двумерные и трехмерные преобразования.
5. Построение проекций трехмерных объектов.
6. Работа с OpenGL.

**Контрольная работа №2.**

**Защита рефератов.**

**Экзамен.**



# Лабораторные работы

---

1. Растровая и векторная графика (6 ч.).
2. Растровые алгоритмы (6 ч.).
3. Двумерная графика (4 ч.).
4. Трехмерная графика. Проецирование. (4 ч.)
5. Работа с OpenGL (12 ч.).

Для защиты л.р. необходимо представить работающую программу, печатный отчет и ответить на вопросы.

# Форма отчетности.

## Лабораторные работы

---

- Скрепленный печатный отчет, содержащий
  - титульный лист;
  - задание;
  - последовательный ход выполнения работы, проиллюстрированный фрагментами кода, алгоритмами и формулами;
  - результат выполнения программы;
  - **ВЫВОДЫ.**



# Реферат (15-20 стр.)

---

- Титульный лист.
- Содержание.
- Введение (постановка задачи, актуальность).
- Последовательное изложение предметной области.
- Заключение.
- Список литературы.





# Литература

---

- Порев В.Н., Блинова Т.А. Компьютерная графика / учебное пособие. – СПб.: БХВ-Петербург, 2006. - 520 с.
- Демин А.Ю., Кудинов А.В. Компьютерная графика / учебное пособие. – Томск: Изд.-во ТПУ, 2005. - 163 с.
- Шикин Е.В., Боресков А.В. Компьютерная графика. –М. «Диалог МИФИ», 2005. – 464с.
- Сайт: <http://compgraph.tpu.ru>
- Поляков А., Брусенцев В. Методы и алгоритмы компьютерной графики в примерах на Visual C++. – СПб.: БХВ-Петербург, 2003.-560с.
- Роджерс Д. Адамс Дж. Математические основы машинной графики: пер. с англ.–М.: Мир, 2001.-604 с.

# Развитие компьютерной графики



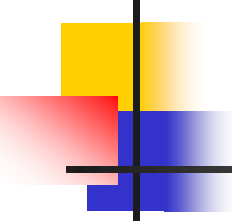
---

80% информации мы получаем  
посредством зрения.



Активное развитие способов человеко-  
машинного взаимодействия.

# Основные направления работы с изображениями



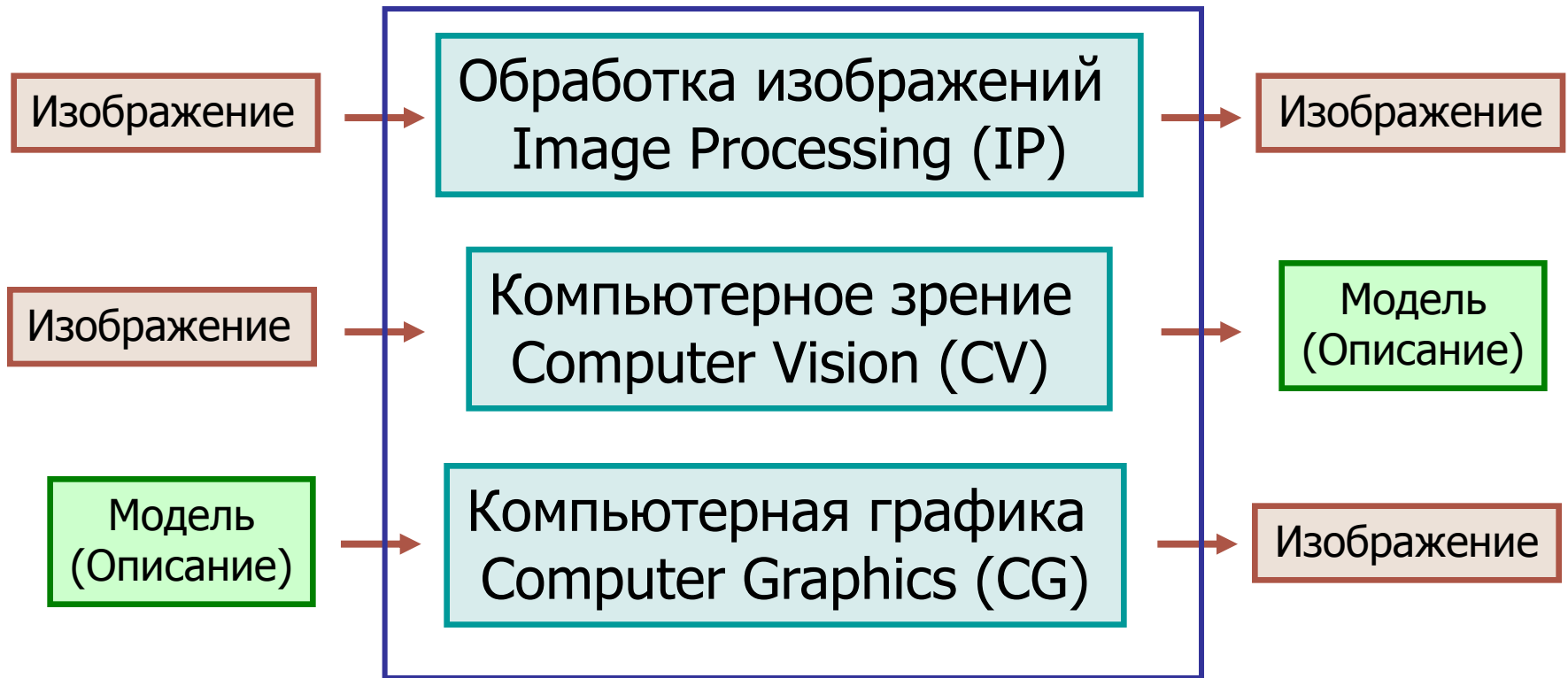
---

Основная функция компьютера - обработка информации (ОИ).

Основные направления работы с изображениями:

- обработка изображений;
- компьютерное зрение;
- компьютерная графика.

# Основные направления работы с изображениями



# Обработка изображений Image Processing

Изображение



Обработка изображений  
Image Processing



Изображение



Захват кадра из видеопотока



Выделение области для распознавания



Контрастирование области распознавания



Замена цвета для усиления контраста между номером и подложкой



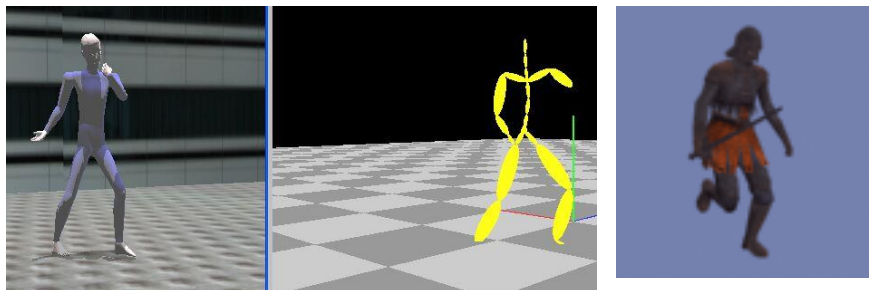
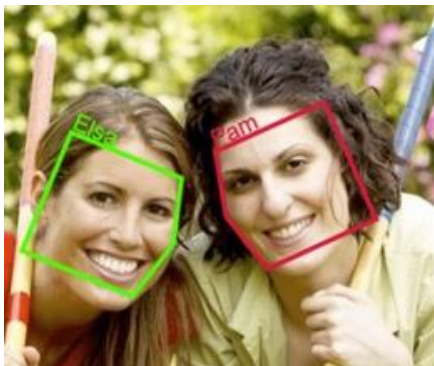
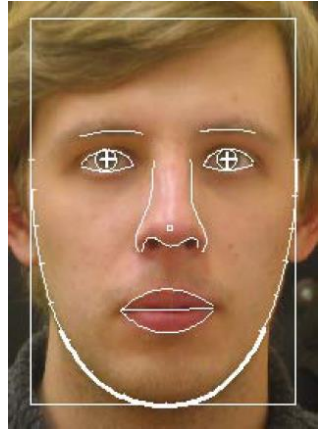
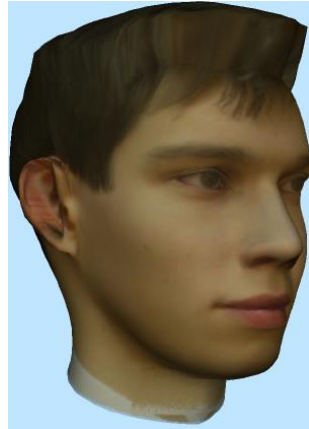


## Обработка изображений

---

- Рассматривает задачи, в которых входные и выходные данные являются изображениями.
  
- Применение:
  1. Повышение качества изображений, для улучшения восприятия: повышение контраста, коррекция цветов, уменьшение шума...
  2. Предварительный шаг в компьютерном зрении (выделение контуров объектов, информативных признаков: углы, границы).

# Компьютерное зрение Computer Vision



Захват движения и анимация персонажа



# Компьютерное зрение

---

- Изучает методы описания изображений, т.е. отвечает на вопрос:  
«Что мы видим на изображении?»
- Применяется:
  - зрение роботов,
  - системы видеонаблюдения,
  - распознавание текстов



# Компьютерная графика Computer Graphics

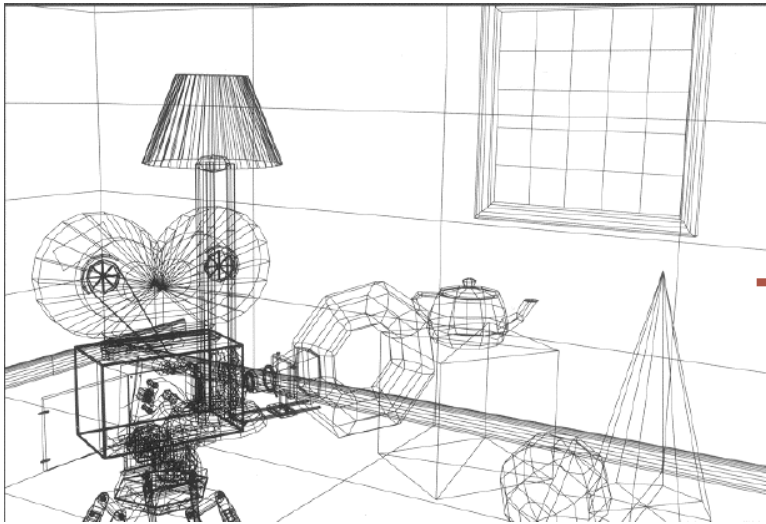
Модель  
(Описание)



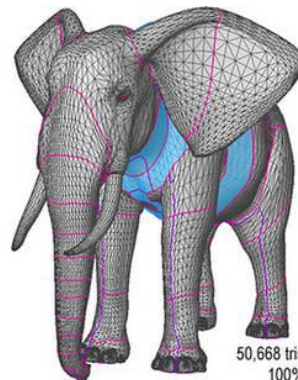
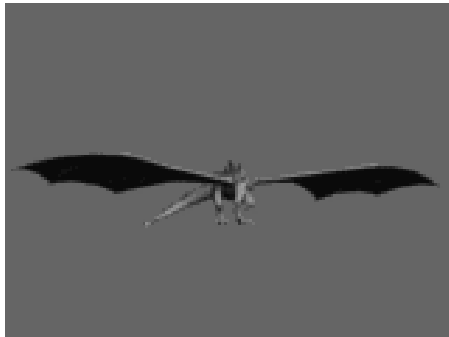
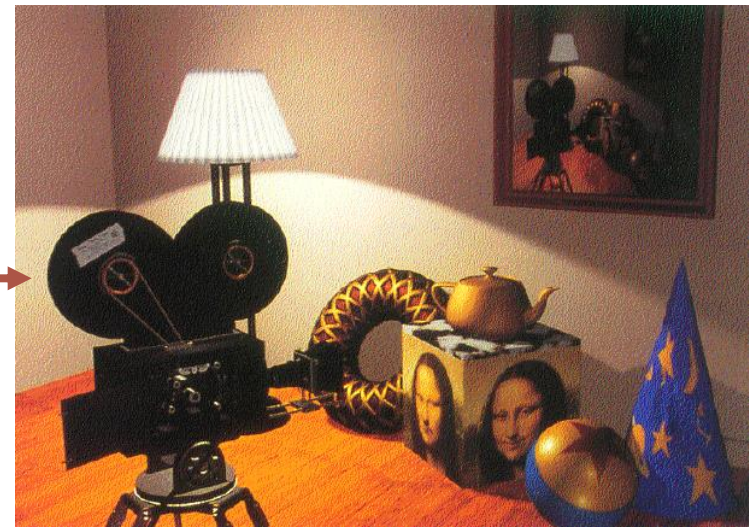
Компьютерная графика  
Computer Graphics



Изображение



Графический  
конвейер





# Компьютерная графика

---

- Изучает способы визуализации изображений на устройствах вывода.
- Применение:  
виртуальная реальность, игры,  
архитектура, моделирование различных  
процессов, САПР.

Сводная таблица основных направлений деятельности, связанных с изображениями

<b>Параметры</b>	<b>Обработка изображений</b>	<b>Компьютерное зрение</b>	<b>КГ</b>
Задача	Преобразование	Получение описания (модели)	Визуализация, создание изображения по модели
Назначение	Реставрация, улучшение качества, выделение признаков	Зрение роботов, распознавание текстов, системы безопасности	2D,3D моделирование процессов, системы виртуальной реальности, комп. игры, архитектура, реклама, ГИС, САПР
Входные данные	Изображение	Изображение	Модель
Выходные данные	Изображение	Модель	Изображение



# Основные определения КГ

---

- **Компьютерная графика (КГ)** – это раздел информатики, изучающий методы и средства построения, преобразования, хранения и вывода изображений различных геометрических объектов и сцен с помощью ЭВМ.
- **Интерактивная КГ** – раздел КГ, изучающий вопросы динамического управления со стороны пользователя содержанием изображения с помощью интерактивных устройств взаимодействия.



# Этапы построения изображений

---

**1. Моделирование** – применение методов математического аппарата для наиболее полного описания изображения или сцены. (*Компьютерная геометрия*).

**2. Визуализация** (отображение) – построение изображения на плоском экране дисплея путем преобразования модели объектов или сцен в статическое изображение или последовательность статических кадров (видео). (*Компьютерная графика*).

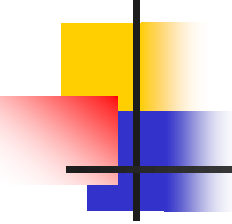


# История развития КГ

---

[http://www.youtube.com/watch?v=QB4\\_pDMUWmI](http://www.youtube.com/watch?v=QB4_pDMUWmI)

# История развития КГ. 1940-1970 гг.



---

- Время больших компьютеров (эра до персональных компьютеров).
- Особенности: пользователь не имел доступа к монитору, графика развивалась на математическом уровне и выводилась на принтере в виде текста, напоминающего на большом расстоянии изображение. Графопостроители появились в конце 60-х годов.





# 1950 г. Векторные дисплеи

- Массачусетский технологический университет машина Whirlwind-I («Вихрь I») в 1950 г.
- Для создания имитатора полетов для решения задач стабилизации поведения самолетов и отработки точности бомбометания.



# 1960 г. Цифровые дисплеи

---

- Первые компьютеры с мощностью, достаточной для выполнения задач цифровой обработки изображений появились в начале 60-х.

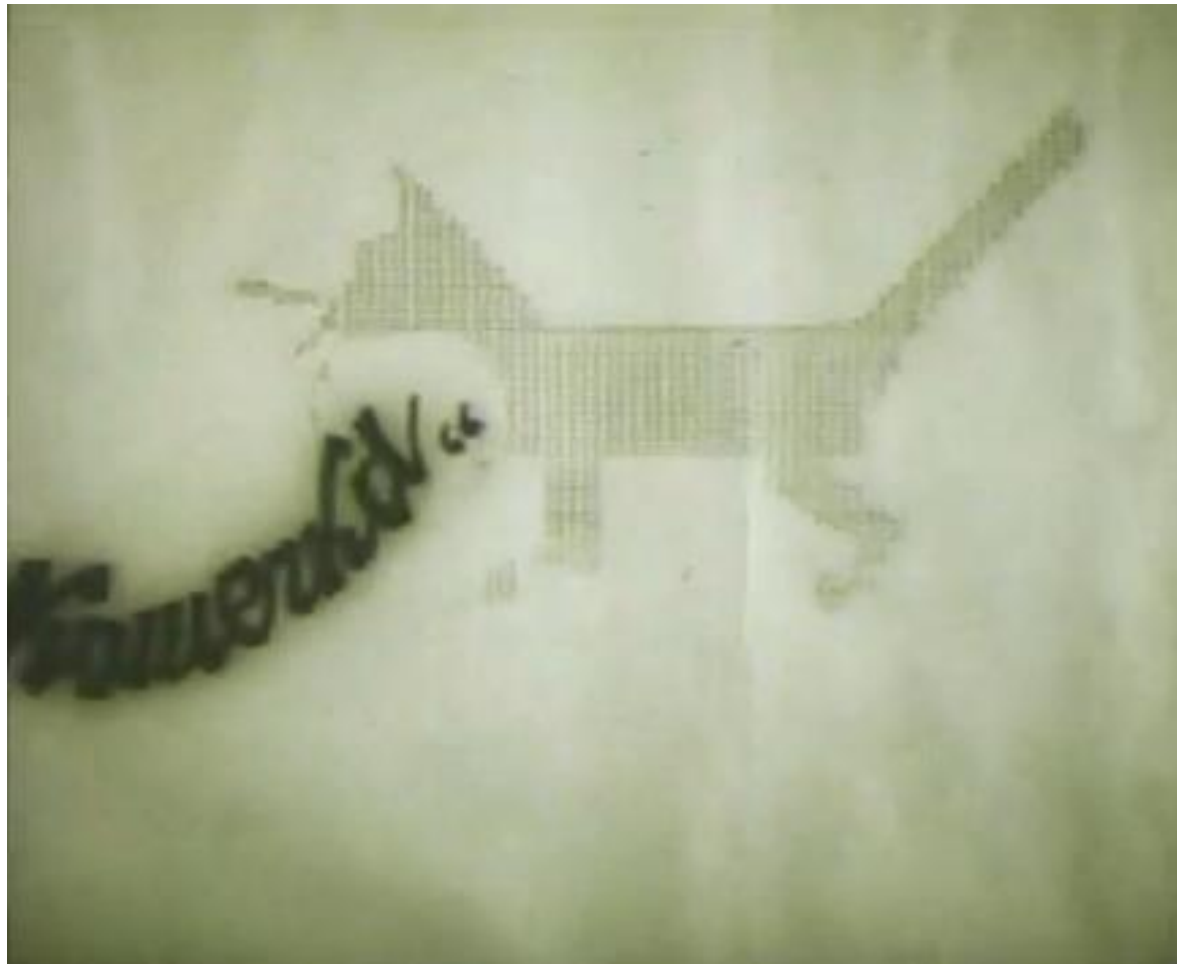


# История развития КГ.

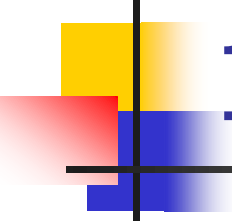
## 1968 г. Фильм «Кошечка»

- В МГУ группой под руководством Н.Н. Константинова была создана компьютерная математическая модель движения кошки.
- Машина БЭСМ-4, выполняя написанную программу решения дифференциальных уравнений, рисовала мультфильм «Кошечка», который для своего времени являлся прорывом. Для визуализации использовался алфавитно-цифровой принтер.

<http://mults.info/mults/?id=1342>



# История развития КГ. 1971-1985 гг.



---

- Появились персональные компьютеры, т.е. появился доступ пользователя к дисплеям. Роль графики резко возросла, но наблюдалось очень низкое быстродействие компьютера. Программы писались на ассемблере. Появилось цветное изображение.
- Особенности: этот период характеризовался зарождением реальной графики.



# Конец 80-х

---

- Программное обеспечение имелось для всех сфер применения: от комплексов управления до настольных издательств.
- Возникло новое направление рынка на развитие аппаратных и программных систем сканирования, автоматической оцифровки.

# История развития КГ

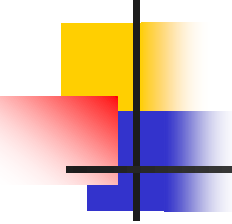
## 1986-1990 гг.

---

- Появление технологии Мультимедиа. К графике добавились обработка звука и видеоизображения, общение пользователя с компьютером расширилось.
- Особенности: появление диалога пользователя с персональным компьютером; появление анимации и возможности выводить цветное изображение.

# История развития КГ

## 1991-2008 гг.



---

- Появление графики нашего дня Virtual Reality.
- Появились датчики перемещения, благодаря которым компьютер меняет изображения при помощи сигналов посылаемых на него.
- Появление стереочков (монитор на каждый глаз), благодаря высокому быстродействию которых, производится имитация реального мира.
- Замедление развития этой технологии из-за опасения медиков, т.к. благодаря Virtual Reality можно очень сильно нарушить психику человека, благодаря мощному воздействию цвета на неё.





---

# Области применения компьютерной графики



# Области применения компьютерной графики

---

## **Научная графика**

Первые компьютеры использовались лишь для решения научных и производственных задач. Чтобы лучше понять полученные результаты, производили их графическую обработку, строили графики, диаграммы, чертежи рассчитанных конструкций.



# Современные ПО научной визуализации

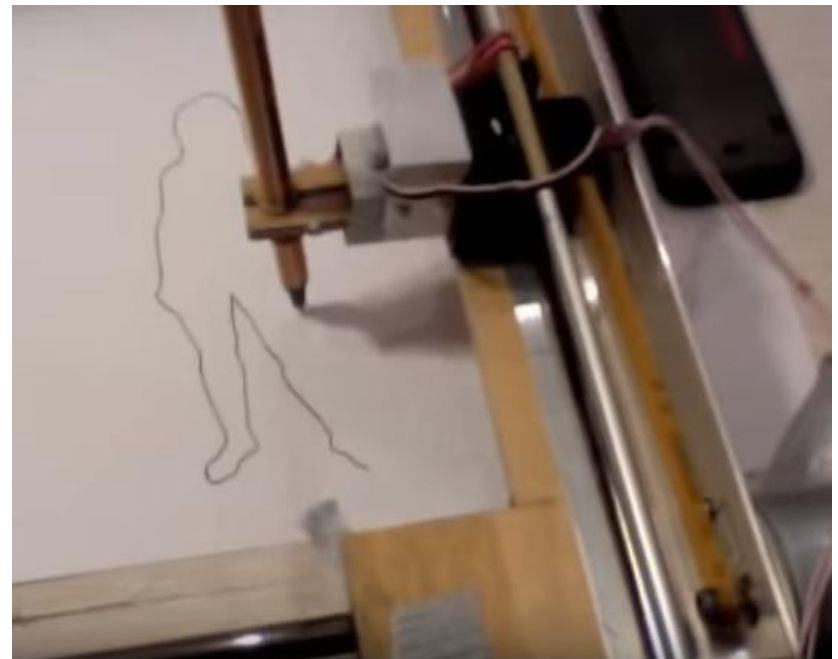
---

- Визуализация научных данных  
(построение графиков, моделирование  
различных процессов)
  - Maple,
  - Matlab,
  - MathCad.

# Области применения компьютерной графики

Затем появились специальные устройства - **графопостроители (плоттеры)** для вычерчивания чертежей и графиков чернильным пером на бумаге.

[http://www.youtube.com/watch?v=8ES61MbSmpc&feature=player\\_embedded](http://www.youtube.com/watch?v=8ES61MbSmpc&feature=player_embedded)





## Области применения компьютерной графики

---

**Деловая графика** - область компьютерной графики, предназначенная для наглядного представления различных показателей работы учреждений.

Плановые показатели, отчетная документация, статистические сводки - вот объекты, для которых с помощью деловой графики создаются иллюстративные материалы.

- Microsoft Excel;
- Visio.



# Области применения компьютерной графики

---

**Конструкторская графика** используется в работе инженеров-конструкторов, архитекторов, изобретателей новой техники.

Этот вид компьютерной графики является обязательным элементом **САПР** (систем автоматизации проектирования).

Средствами конструкторской графики можно получать как плоские изображения (проекции, сечения), так и пространственные трехмерные изображения.



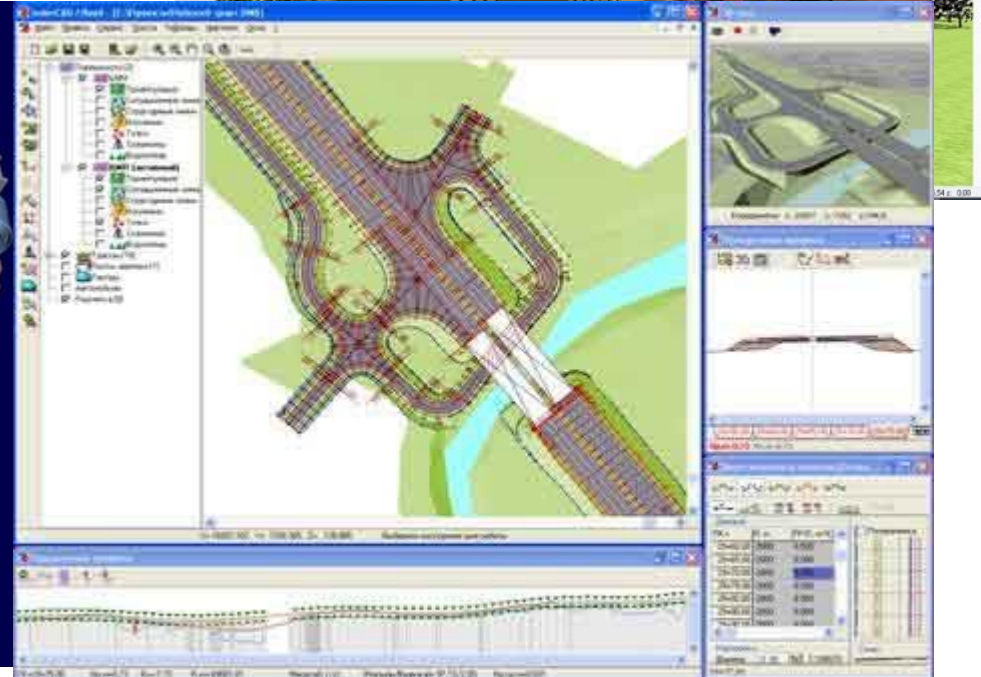
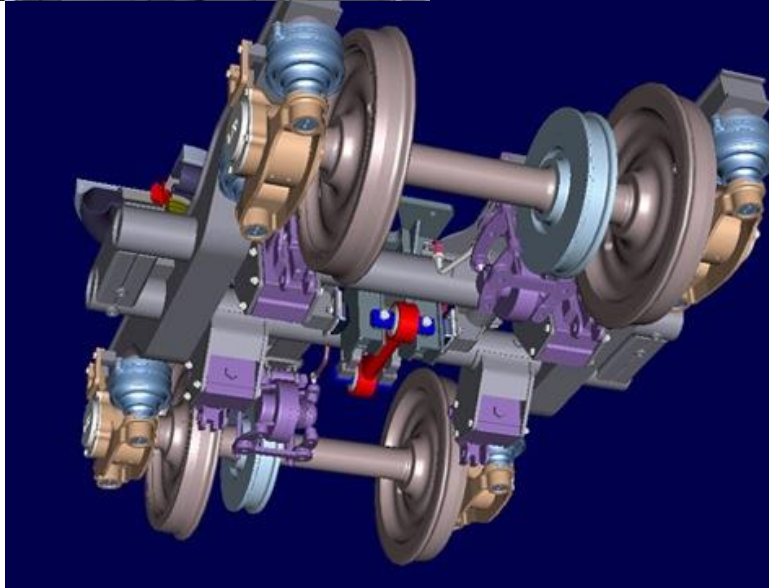
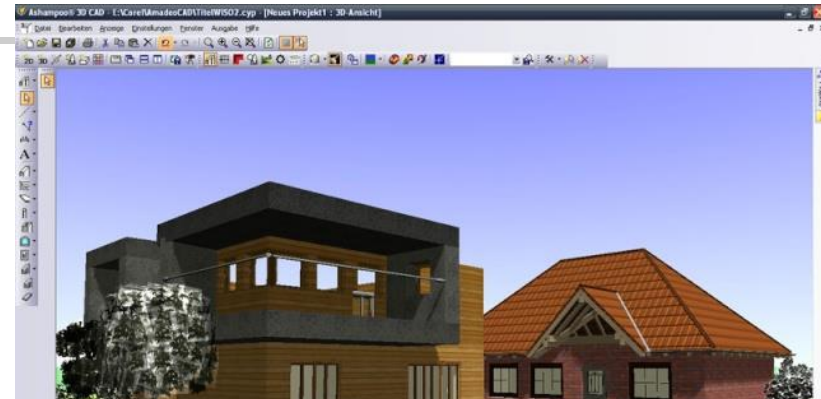
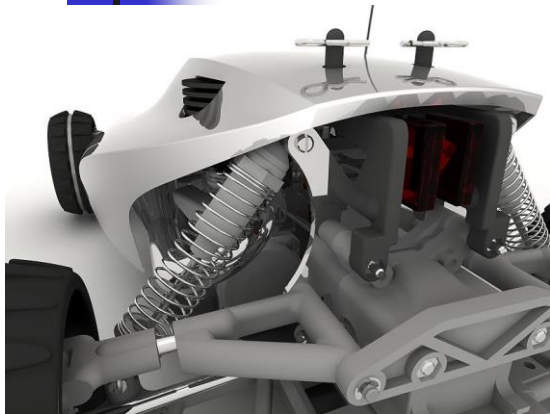
# Подгруппы САПР

---

Выделяют три основные подгруппы САПР:

- **машиностроительные САПР** (MCAD - Mechanical Computer Aided Design)
- **архитектурно-строительные САПР** (CAD/AEC - Architectural, Engineering, and Construction)
- **САПР печатных плат** (ECAD - Electronic CAD/EDA - Electronic Design Automation)

# Пример модели, разработанной в САПР







# Области применения КГиГ(2)

---

- Геометрическое проектирование и моделирование (построение чертежей, эскизов, объемных изображений)
  - CAD (Computer-Added Design)-системы: AutoCad,
  - 3DMax,
  - библиотеки OpenGL, DirectX.



## Области применения компьютерной графики

---

**Иллюстративная графика** - это произвольное рисование и черчение на экране компьютера.

Пакеты иллюстративной графики относятся к прикладному программному обеспечению общего назначения.

Простейшие программные средства иллюстративной графики называются графическими редакторами.

- Paint;
- Adobe Photoshop;
- Corel Draw.



## Области применения компьютерной графики

---

**Художественная и рекламная графика**, ставшая популярной во многом благодаря телевидению.

С помощью компьютера создаются рекламные ролики, мультфильмы, компьютерные игры, видеоуроки, видеопрезентации.

Графические пакеты для этих целей требуют больших ресурсов компьютера по быстродействию и памяти. Отличительной особенностью этих графических пакетов является возможность создания реалистических изображений и "движущихся картинок".



# Области применения КГиГ(З)

---

- Изобразительное искусство (реклама, создание фильмов, мультипликация, издательские системы)
  - Adobe Photoshop,
  - CorelDraw,
  - Adobe Illustrator,
  - Adobe Flash.



# Области применения компьютерной графики

---

**Компьютерная анимация** - это получение движущихся изображений на экране дисплея.

Художник создает на экране рисунке начального и конечного положения движущихся объектов, все промежуточные состояния рассчитывает и изображает компьютер, выполняя расчеты, опирающиеся на математическое описание данного вида движения.

Полученные рисунки, выводимые последовательно на экран с определенной частотой, создают иллюзию движения.

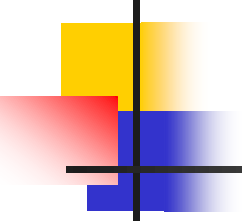
- Easy GIF Animator
- Adobe Flash



# Основные направления КГ(4)

---

- Виртуальная реальность (визуальное моделирование внешнего мира от компьютерных игр до различных тренажеров)
  - OpenGL, DirectX.



---

# Способы хранения изображений в памяти ЭВМ

“Цифровые изображения являются  
усеченной моделью картины  
реального мира”



# Графический формат

---

- Порядок (структура), согласно которому данные, описывающие изображение, записаны в файле:
  - растровый;
  - векторный;
  - фрактальный.



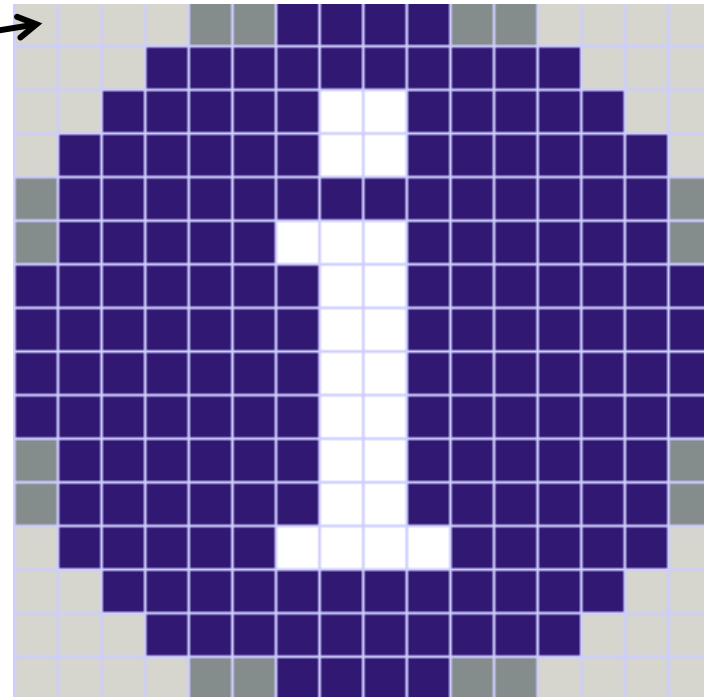
# Растровый способ представления изображений

В виде матрицы отдельных точек (пикселей)  
различных цветов или оттенков.

Каждый цвет  
обозначается  
определенной  
цифрой,  
записанной

в массиве bitmap (битовая карта)

```
15 15 15 15 12 12 08 08 08 ...  
15 15 15 08 08 08 08 08 08 ...  
15 15 08 08 08 08 08 08 08 ...  
.  
.  
.
```





# Растровый способ представления изображений

---

- Достоинство

- возможность получения фотореалистичного изображения высокого качества в различном цветовом диапазоне.

- Недостатки

- большой объем файла для хранения изображения и оперативной памяти для его обработки;
- трудно масштабировать.



# Фотография 10×15 с разрешением 300 dpi:

---

10 см = 3,9 дюйма; 15 см = 5,9 дюймов

По вертикали:  $3,9 * 300 = \mathbf{1170}$  точек

По горизонтали:  $5,9 * 300 = \mathbf{1770}$  точек

Общее число пикселей:

$$1170 * 1770 = \mathbf{2\ 070\ 900}$$

Если 1 цвет = 1 байт, то

$$2\ 070\ 900 * 3 \text{ байта} = \mathbf{5,92 \text{ Мб}}$$



# Сферы применения растровых изображений

---

- Хранение фотографий, видеозаписей.
- Создание изображений со большим количеством оттенков и цветовых переходов.

# Векторный способ представления изображений

Разбиение изображений на ряд графических примитивов – точки, дуги, прямые, кривые ломанные, и др.



**Отрезок** (координаты начала, координаты конца);

**Окружность** (центр окружности, радиус);

**Кривая Безье** (начальная точка, промежуточные точки конечная точка).

# Векторный способ представления изображений



---

## ■ Достоинства:

- Векторное изображение м.б. легко масштабировано без потери качества.
- Графические файлы имеют существенно меньший, по сравнению с растровым, объём.
- Максимально использует разрешающие возможности устройства вывода.



# Векторный способ представления изображений

---

- Недостатки:
  - Трудно получить фотореалистичное изображение.
  - При прорисовке необходимо каждый раз растеризовать изображение.
  - Программная зависимость вследствие отсутствия единого формата.



# Сферы применения векторных изображений

---

- Полиграфика (создание логотипов, красочных иллюстраций).
- САПР.
- Компьютерное моделирование.
- Дополненная/виртуальная реальность.





# Растровая и векторная графика

---



фотография



вектор

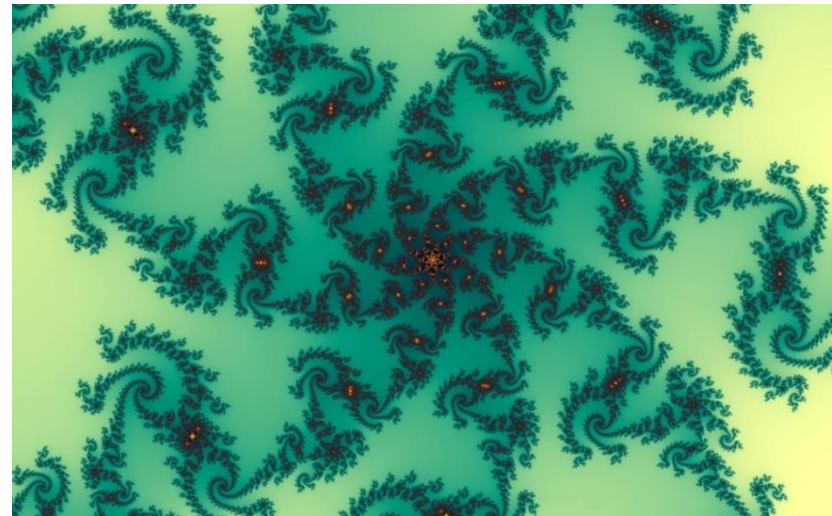


# Фрактальная графика

---

- **Фрактал** (лат. *fractus* — дроблённый, сломанный, разбитый) – геометрическая фигура, обладающая свойством самоподобия.

# Фрактальная графика





# Возникновение термина

---

Термин «фрактал» был введён Бенуа Мандельбротом в 1975 году и получил широкую популярность с выходом в 1977 году его книги  
*«Фрактальная геометрия природы»*.



# Виды фракталов

---

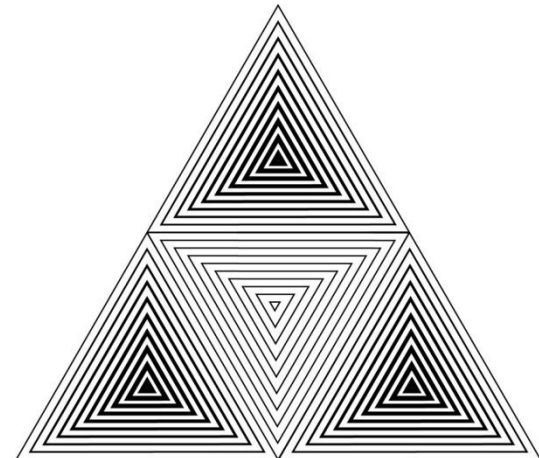
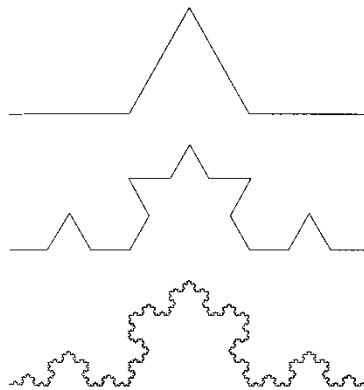
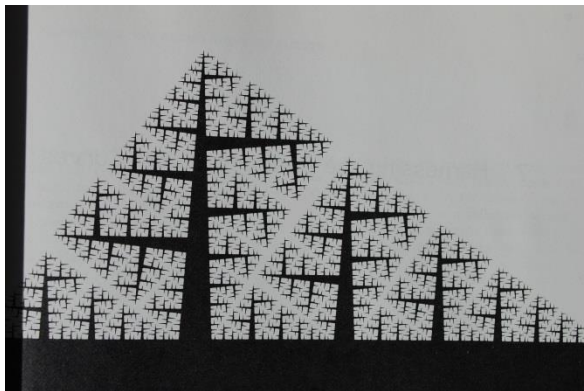
Геометрические фракталы;  
Алгебраические фракталы;  
Стохастические фракталы.



# Геометрические фракталы

---

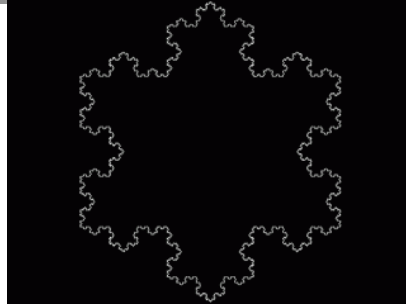
В двухмерном случае их получают с помощью ломаной, плоской фигуры или поверхности (в трехмерном случае), называемой генератором. За один шаг алгоритма каждый из отрезков, составляющих ломаную, заменяется на ломаную-генератор в соответствующем масштабе.



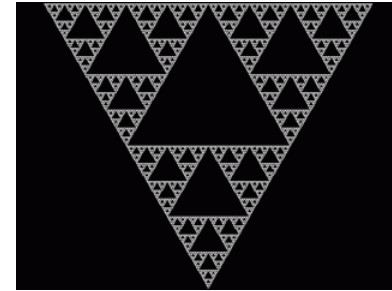
# Примеры геометрических фракталов



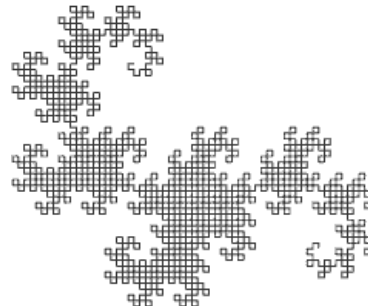
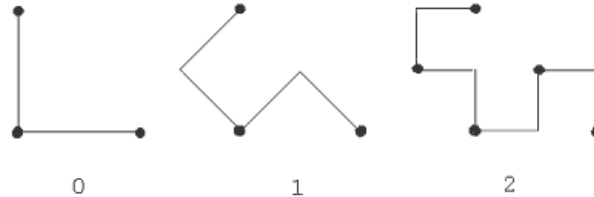
- Снежинка Коха



- Треугольник Серпинского



- Драконова ломаная



# Алгебраические фракталы

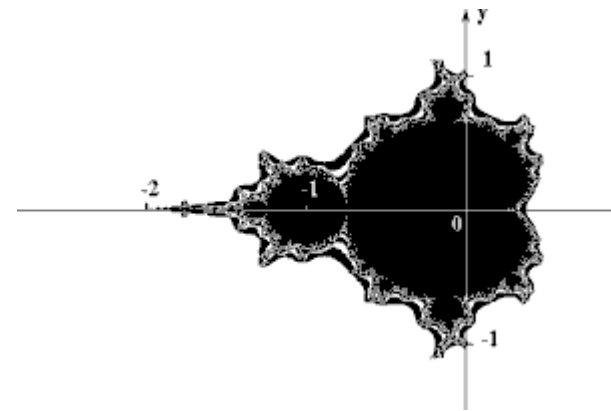
Описываются комплексной нелинейной функцией (многочленом)  $f(z)$ .

1. Возьмем какую-нибудь начальную точку  $z_0$  на комплексной плоскости.

2. Рассмотрим бесконечную последовательность чисел на комплексной плоскости, каждое следующее из которых получается из предыдущего:  $z_0, z_1 = f(z_0), z_2 = f(z_1), \dots, z_{n+1} = f(z_n)$ .

В зависимости от начальной точки  $z_0$  такая последовательность может:

- стремиться к бесконечности при  $n \rightarrow \infty$ ;
- сходиться к какой-то конечной точке;
- циклически принимать ряд фиксированных значений.







# Стохастические фракталы

---

Получаются в том случае, если в итерационном процессе хаотически менять какие-либо его параметры. При этом получаются объекты очень похожие на природные - несимметричные деревья, изрезанные береговые линии и т.д.

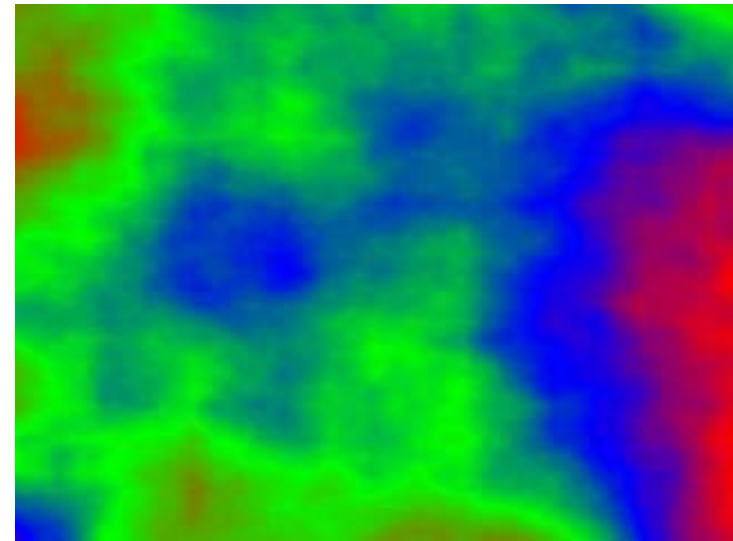
Двумерные стохастические фракталы используются при моделировании рельефа местности и поверхности моря.



# Стохастический фрактал. «Плазма»

- Возьмем прямоугольник и для каждого его угла определим цвет.
- Далее находим центральную точку прямоугольника и раскрашиваем ее в цвет равный среднему арифметическому цветов по углам прямоугольника плюс некоторое случайное число. Чем больше случайное число - тем более "рваным" будет рисунок.
- Если мы теперь скажем, что цвет точки это высота над уровнем моря - получим вместо плазмы - горный массив.

Так моделируются горы в большинстве программ: строится карта высот «плазма», к ней применяются различные фильтры, накладываем текстуру и, горы готовы.



# Классификация ПО компьютерной графики (1)



**Редакторы растровой графики:** Adobe Photoshop, Corel PhotoPaint, Paint

**Редакторы векторной графики:** Adobe Illustrator, Corel Draw, Macromedia FreeHand

**Настольные издательские системы (верстка документов):** Adobe PageMaker, QuarkXPress

**Смешанные системы и имитаторы рисования:** Corel Xara, Fractal Design Painter

# Классификация ПО компьютерной графики (2)

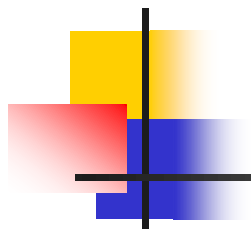


**Векторизаторы:** Adobe StreamLine

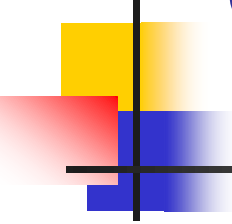
**Средства 3D графики, анимации и САПР:** 3D Studio Max, Ray Dream Studio, AutoCAD 3D

**Графические библиотеки и стандарты:**  
OpenGL, DirectX

**Средства web-дизайнера:** MSFrontPage, GIFConstructionSet



# Основные характеристики растровых изображений



# Основные характеристики растровых изображений

---

- **Разрешение** – число пикселей, отводимых как по горизонтали, так и по вертикали на единицу длины.
- **Глубина цвета** – число бит, используемых для представления каждого пикселя изображения



# Разрешение растровых изображений

---

- **Разрешение** – количество пикселей на единицу длины:

принтеры – **lpi** (кол-во линий на дюйм);

сканеры – **dpi** (кол-во точек на дюйм);

мониторы – **ppi** (кол-во точек на дюйм).

- **Размер растра** - количество пикселей по горизонтали и вертикали.



## Разрешение оригинала - dpi

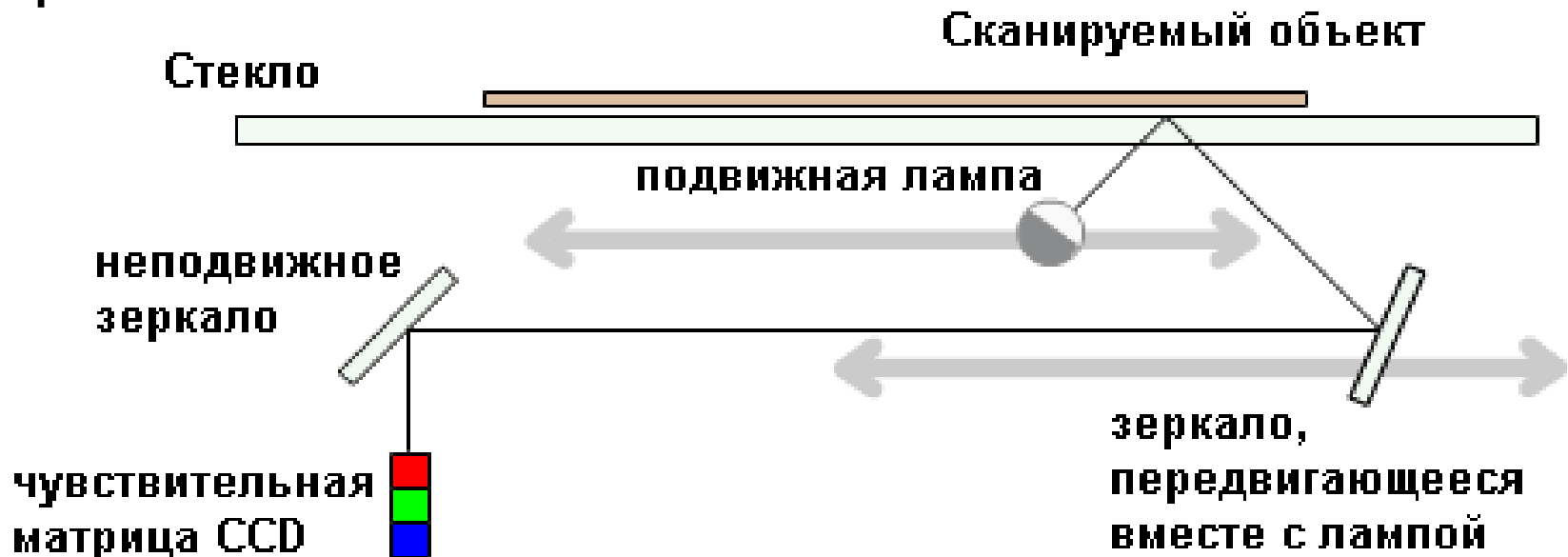
---

- Используется при вводе изображения в компьютер (сканеры), измеряется в точках на дюйм (dots per inch - **dpi**).



# Разрешение оригинала - dpi

- В процессе сканирования светочувствительный элемент сканера измеряет оптическую плотность сканируемого оригинала по всей площади в точках с **заданным интервалом** вдоль и поперек оригинала.





# Разрешение оригинала - dpi

---

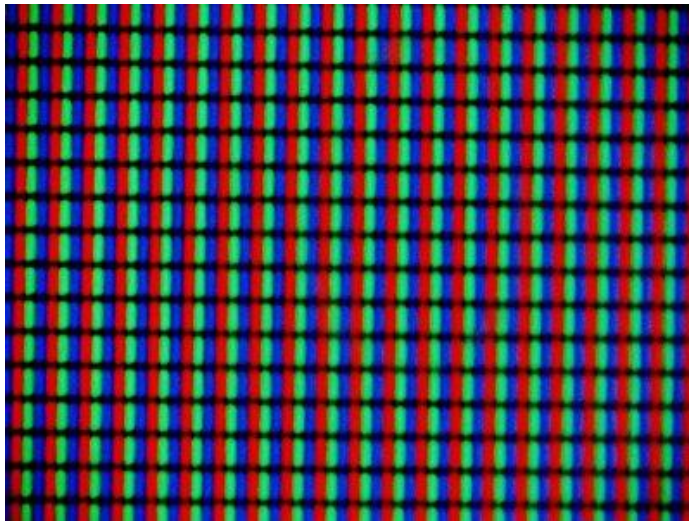
- В результате получается прямоугольная таблица, каждая ячейка которой соответствует измеренному значению цвета.
- Каждая ячейка таблицы называется точкой (**dot**), а вся таблица – растровым изображением.



# Разрешение экранного изображения - ppi

---

- Экран монитора покрыт прямоугольной сеткой из точек люминофора. Каждой точке изображения ставится в соответствие точка люминофора, называемая пикселем (pixel).
- Разрешение измеряется в единицах **ppi**  
(**pixel per inch**)





## Разрешение печатного изображения

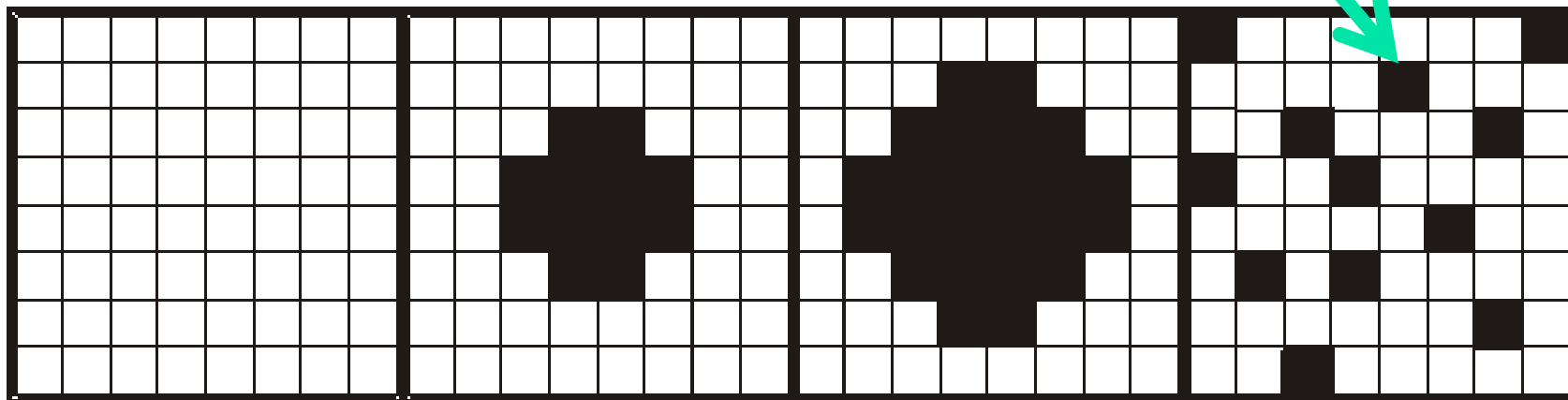
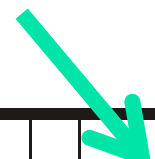
---

- При растривании на оригинал как бы накладывается сетка линий, ячейки которой образуют элемент растра.
- Частота сетки растра измеряется числом линий на дюйм (lines per inch – **lpi**) и называется линиатурой.

# Как регулировать яркость при печати?

- Растррирование с амплитудной модуляцией.
- Растррирование с частотной модуляцией.

более качественно



ячейка растра

АМ-растр  
18,75 %

АМ-растр  
50 %

ЧМ-растр  
18,75 %



# Основные характеристики растра. Глубина цвета

---

- **Глубина цвета** - количество бит, используемых для кодирования одного пикселя.
- **Цветовой диапазон** - максимальное количество цветов.

Как вычислить цветовой диапазон?



# Цветовой диапазон

---

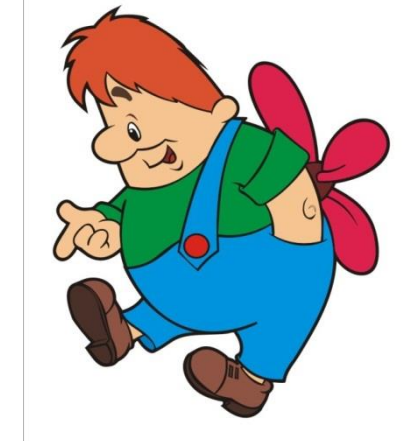
- Цветовой диапазон равен  $2^n$ , где  $n$  – количество битов, отводимое для хранения значения цвета одного пикселя.

# Типы изображений

## в зависимости от глубины цвета

- **бинарные** – 1 бит/пиксель.
- в **оттенках/градациях серого** (256 цветов) – 8 бит/пиксель;
- **цветные** – от 2 бит/пиксель и больше.
  - High color (16 бит/пиксель – 65536 цветов)
  - True color (24 бита/пиксель – 16,7 млн. цветов)

Человек способен различать 350 000 цветов.





# Цветные изображения

- Полноцветные

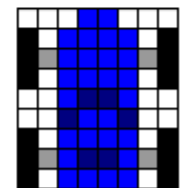
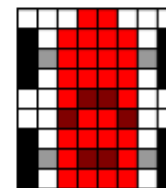


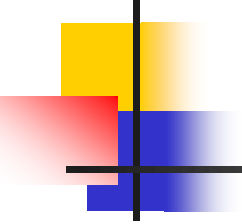
- С индексированной палитрой

0	0	0	1	1	0	0	0
3	0	1	1	1	1	0	3
3	4	1	1	1	1	4	3
3	0	1	1	1	1	0	3
0	0	1	2	2	1	0	0
0	0	2	1	1	2	0	0
3	0	1	1	1	1	0	3
3	4	1	2	2	1	4	3
3	0	1	1	1	1	0	3

0	
1	Red
2	Dark Red
3	Black
4	Grey

0	
1	Blue
2	Dark Blue
3	Black
4	Grey





---

# Векторная графика. Построение кривых

# Представление кривых

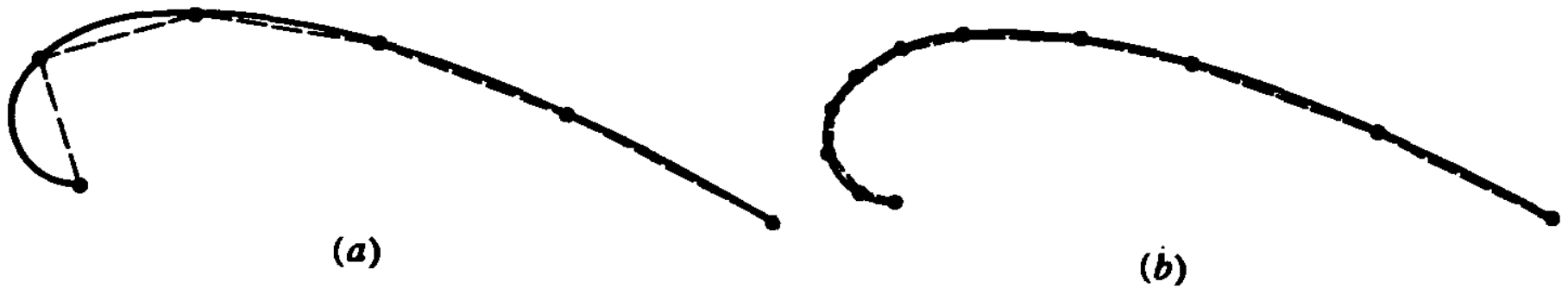


Рис. 4-1 Точечное представление кривых. (a) Равномерная плотность точек вдоль кривой; (b) плотность точек возрастает с уменьшением радиуса кривизны.

## Представление кривых на плоскости

- явный способ (explicit curves)

$$y = f(x) \qquad y = \sin(x)$$

- неявный способ (implicit)

$$f(x, y) = 0 \qquad x^2 + y^2 - R^2 = 0$$

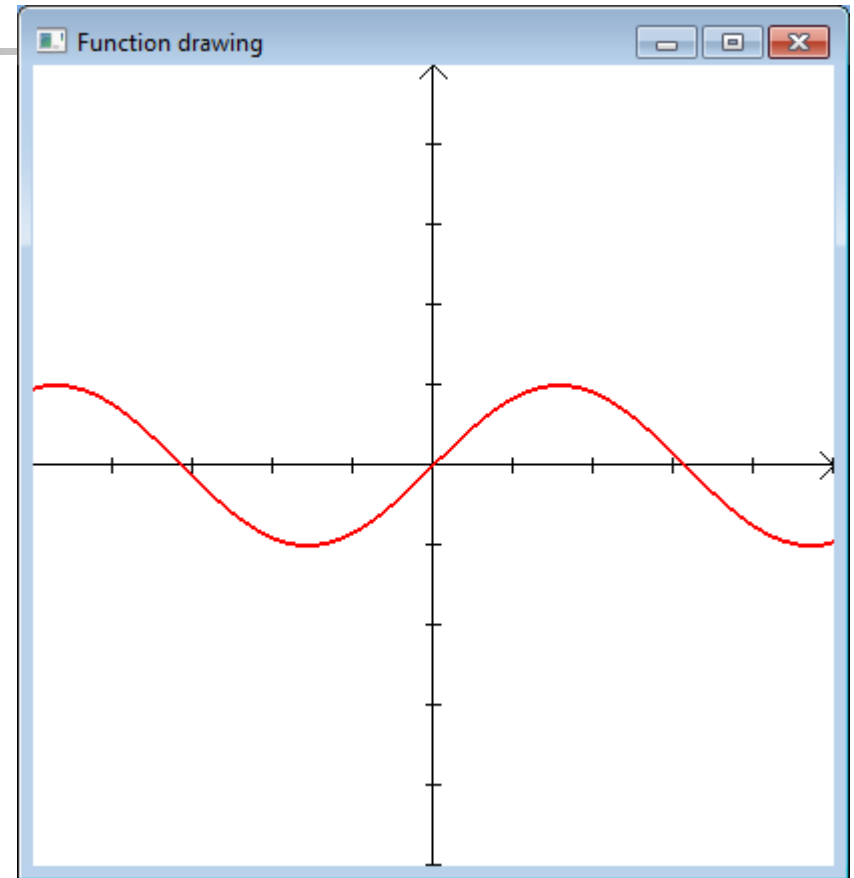
- Параметрический способ (parametric curves)

$$\begin{cases} x = f_x(t) \\ y = f_y(t) \end{cases}, t \in [a, b] \qquad \begin{cases} x = \cos(t) \\ y = \sin(t) \end{cases}, t \in [0, 2\pi)$$

## Явные кривые

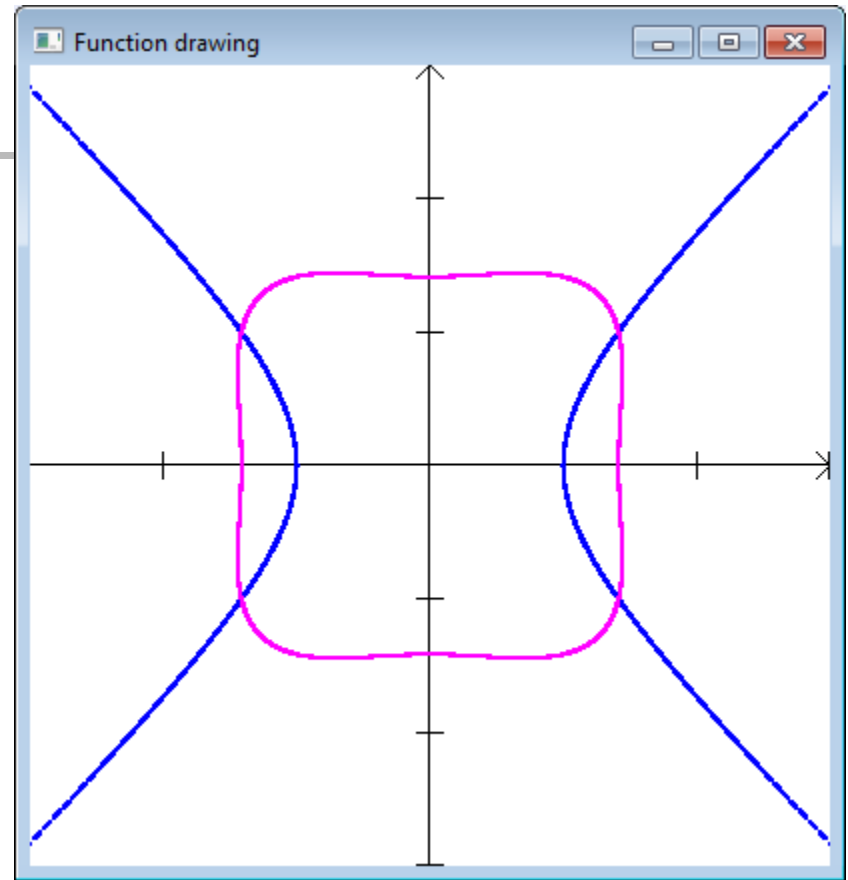
```
x = MinX;  
MoveTo(x, F(x));  
while (x <= MaxX)  
{  
    LineTo(x, F(x));  
    x = x + Step;  
}
```

  $y = \sin(x)$



# Неявные кривые

```
y = MinY;  
while (y <= MaxY)  
{  
  x = MinX;  
  while (x <= MaxX)  
  {  
    f = F(x, y);  
  
    if (f > -EPS && f < EPS)  
      SetPixel(x, y);  
  
    x = x + Step;  
  }  
  y = y + StepY;  
}
```

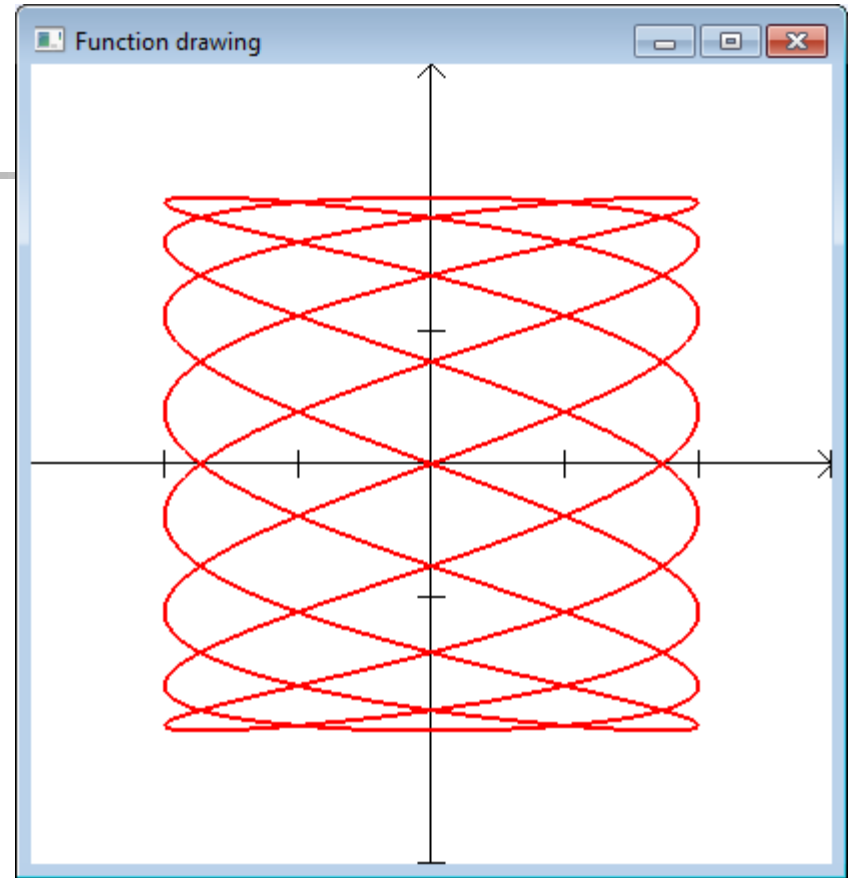


—  $x^2 - y^2 - 1 = 0$

—  $x^4 + y^4 - x^2 - y^2 - 2 = 0$

## Параметрические кривые

```
t = A;  
x = Fx(t);  
y = Fy(t);  
MoveTo(x, y);  
while (t <= B)  
{  
    x = Fx(t);  
    y = Fy(t);  
    LineTo(x, y);  
    t = t + Step;  
}
```



$$\begin{cases} f_x(t) = 2 \cdot \sin(8 \cdot t) \\ f_y(t) = 2 \cdot \sin(3 \cdot t) \end{cases}$$



# Определения

---

Сначала дадим несколько общих определений.

- ❑ *Слайн* — кривая, удовлетворяющая некоторым критериям гладкости.
- ❑ *Базовые (опорные) точки* — набор точек, на основе которых выполняется построение кривой.
- ❑ *Интерполяция* — построение кривой, точно проходящей через набор базовых точек.
- ❑ *Аппроксимация* — сглаживание, приближение, т. е. построение гладкой кривой, проходящей не через набор базовых точек, а вблизи них.
- ❑ *Экстраполяция* — построение линии за пределами интервала, заданного набором базовых точек.



# Линейная интерполяция

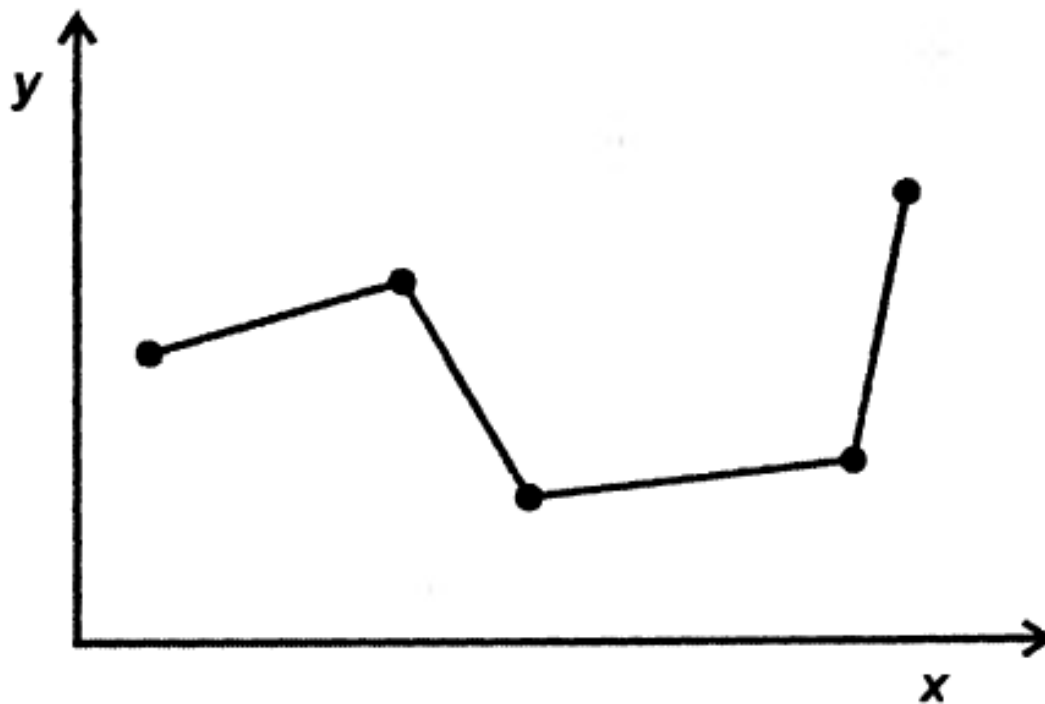


Рис. 8.1. Линейная интерполяция

Проста, но не очень точно определяет положение объекта на интервалах между заданными точками.



# Критерий гладкости

---

Критерием гладкости является существование производных функции, описывающей кривую. Какого порядка существует производная — такого порядка и гладкость. Обычно достаточно гладкой считается функция, если она имеет производную первого или второго порядка.

# Сплайновые кривые.

## Кривые Безье

---

- **Кривые Безье** или **Кривые Бернштейна-Безье** были разработаны в 60-х годах XX века независимо друг от друга Пьером Безье компании «Рено» и Полем де Кастельжо из компании «Ситроен», где применялись для проектирования кузовов автомобилей.
- Кривая Безье является частным случаем многочленов **Бернштейна**, описанных советским математиком в 1912 году.



# Кривые Безье (Pierre Bézier): линейные

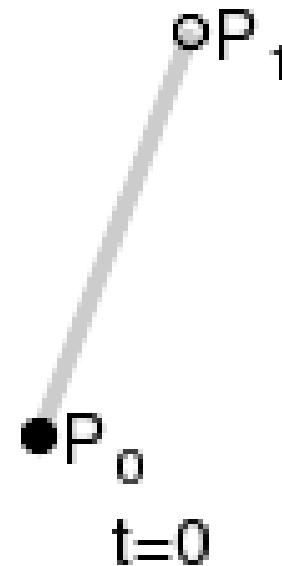
---

## ■ Линейные кривые Безье

- Линейная интерполяция между концевыми точками

$$B = P_0 \cdot (1-t) + P_1 \cdot t$$

$$t \in [0, 1]$$



# Кривые Безье: квадратичные

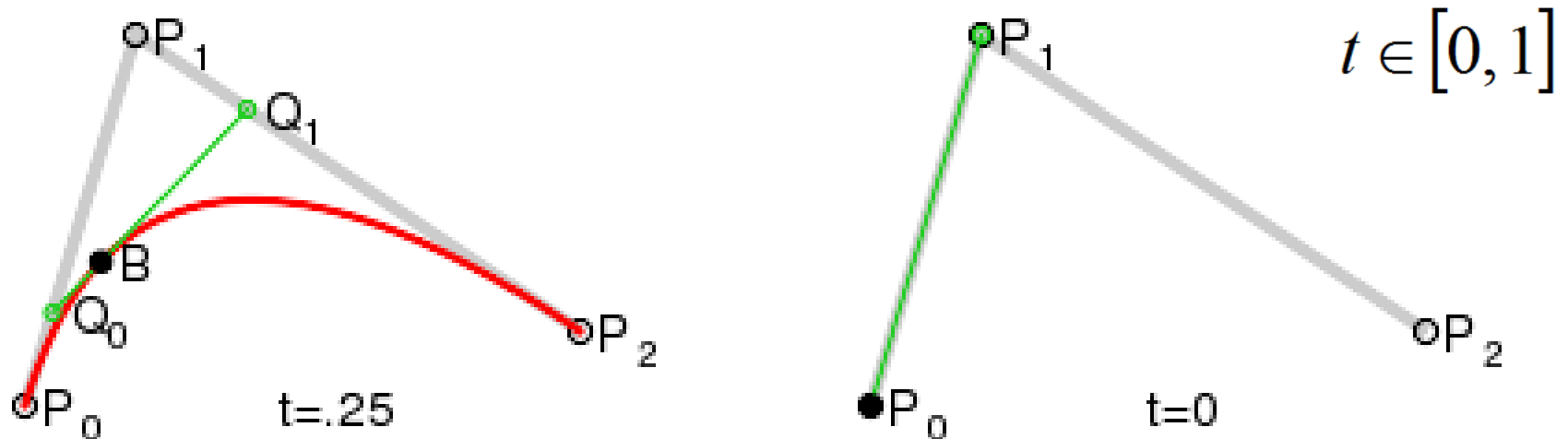
## ■ Квадратичные кривые Безье

- Композиция нескольких линейных кривых:

$$Q_1 = P_0 \cdot (1-t) + P_1 \cdot t$$

$$Q_2 = P_1 \cdot (1-t) + P_2 \cdot t$$

$$B = Q_0 \cdot (1-t) + Q_1 \cdot t = P_0 \cdot (1-t)^2 + 2 \cdot P_1 \cdot (1-t) \cdot t + P_2 \cdot t^2$$



# Кривые Безье: кубические

## ■ Кубические кривые Безье

$$Q_0 = P_0 \cdot (1-t) + P_1 \cdot t$$

$$Q_1 = P_1 \cdot (1-t) + P_2 \cdot t$$

$$Q_2 = P_2 \cdot (1-t) + P_3 \cdot t$$

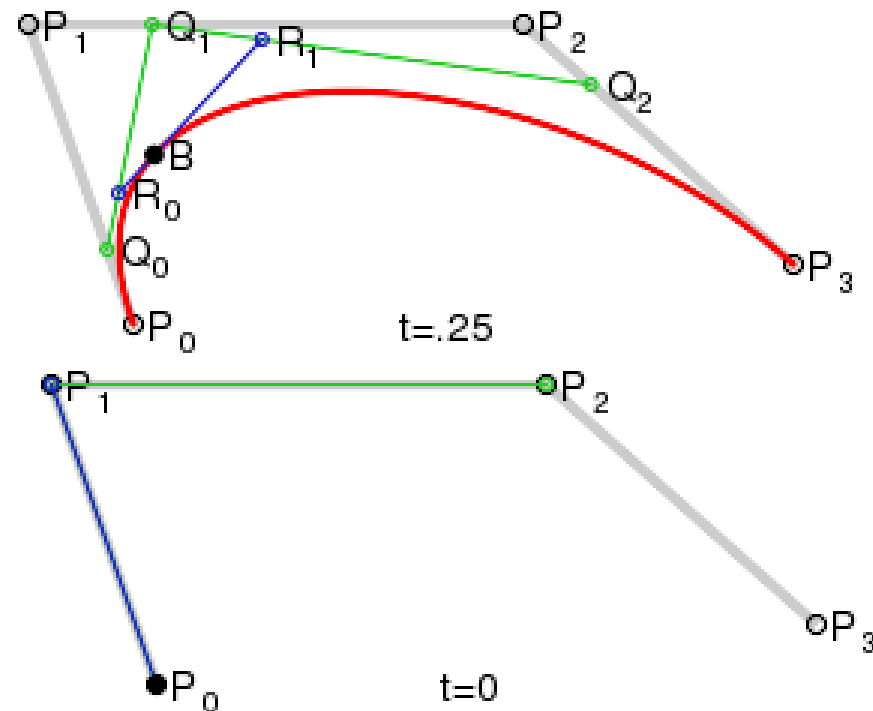
$$R_0 = Q_0 \cdot (1-t) + Q_1 \cdot t$$

$$R_1 = Q_1 \cdot (1-t) + Q_2 \cdot t$$

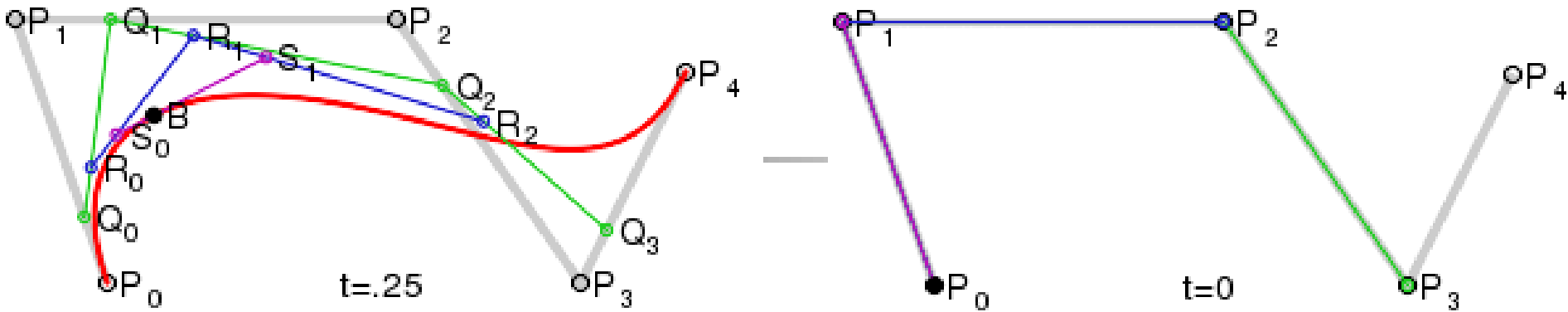
$$B = R_0 \cdot (1-t) + R_1 \cdot t =$$

$$= P_0 \cdot (1-t)^3 + 3 \cdot P_1 \cdot (1-t)^2 \cdot t + 3 \cdot P_2 \cdot (1-t) \cdot t^2 + P_3 \cdot t^3$$

$$t \in [0, 1]$$



# Кривые Безье: старшие степени



- В общем случае:

$$B(t) = \sum_{i=0}^n P_i \cdot \mathbf{b}_{i,n}(t)$$

$$\mathbf{b}_{i,n}(t) = C_n^i \cdot t^i \cdot (1-t)^{n-i}$$

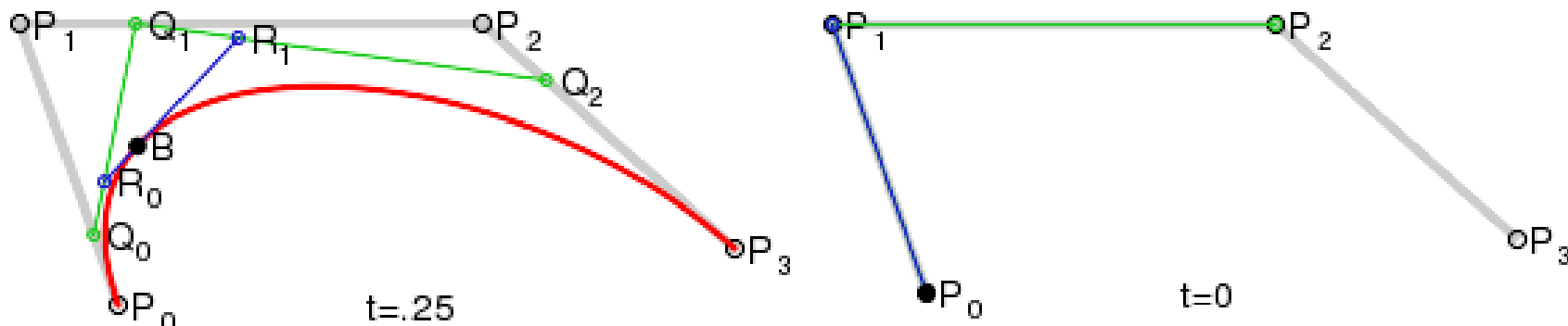
полином Бернштейна

$$C_n^i = \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

число Сочетаний

# Алгоритм де Кастельжо построения кривых Безье

- Задана кривая Безье с опорными точками . Соединив последовательно опорные точки с первой по последнюю, получаем ломаную линию (**серые отрезки**).
- Разделяем каждый полученный отрезок этой ломаной в соотношении  $t : (1-t)$  и соединяем полученные точки. В результате получаем ломаную линию с количеством отрезков, меньшим на один, чем исходная ломаная линия (**зеленые и синий отрезки**).
- Повторяем процесс до тех пор, пока не получим единственную точку. Эта точка и будет являться точкой на заданной кривой Безье с параметром (**черная точка**).





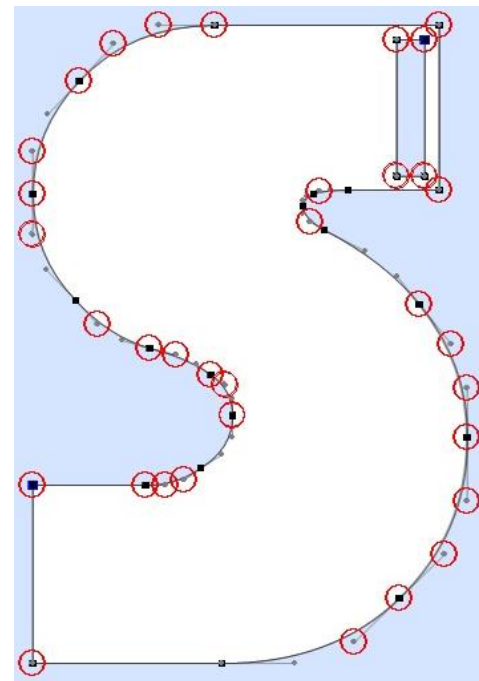
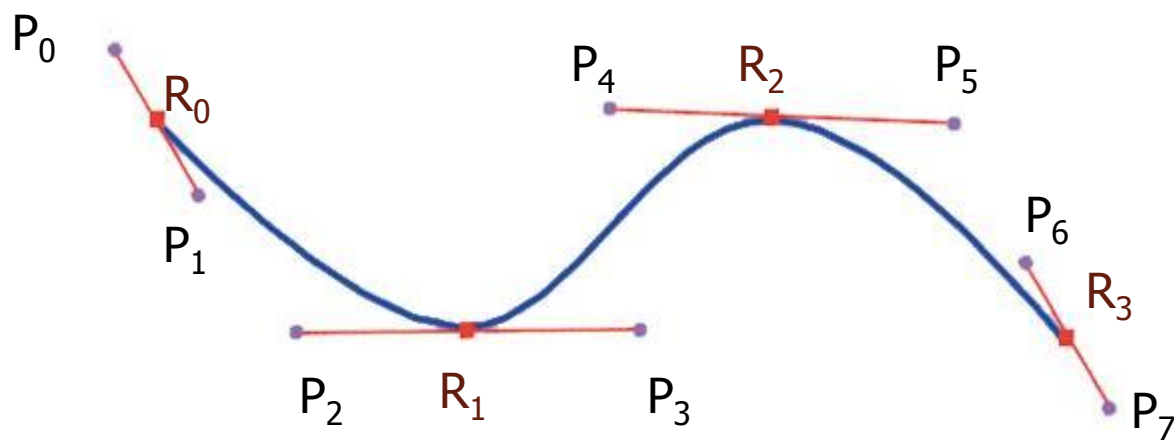
# Кубические кривые Безье: матричная запись

$$\begin{aligned} B(t) &= P_0 \cdot (1-t)^3 + 3 \cdot P_1 \cdot (1-t)^2 \cdot t + 3 \cdot P_2 \cdot (1-t) \cdot t^2 + P_3 \cdot t^3 = \\ &= t^3 \cdot (-P_0 + 3 \cdot P_1 - 3 \cdot P_2 + P_3) + t^2 \cdot (3 \cdot P_0 - 6 \cdot P_1 + 3 \cdot P_2) + \\ &\quad t \cdot (-3 \cdot P_0 + 3 \cdot P_1) + P_0 = \end{aligned}$$

$$= \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$B(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot M_B \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

# Сопряжение кривых Безье



# Сплайновые кривые Catmull-Rom

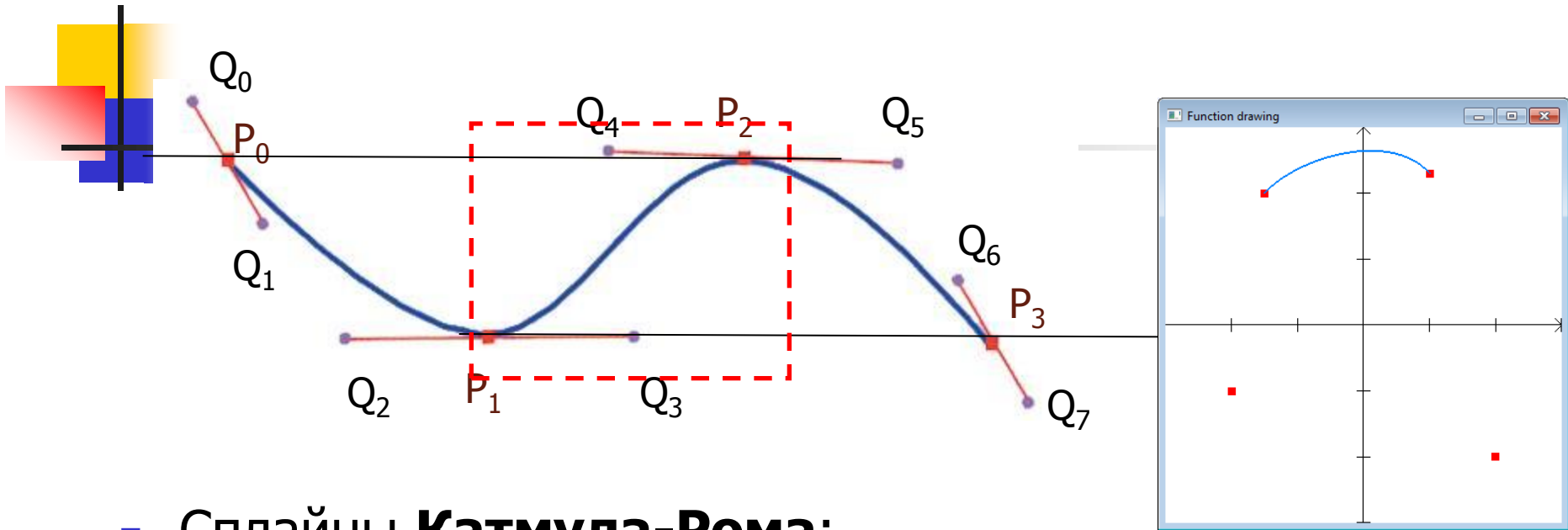
$$R(t) = \frac{1}{2} \left( -t(1-t)^2 P_0 + (2 - 5t^2 + 3t^3) P_1 + t(1 + 4t - 3t^2) P_2 - t^2(1-t) P_3 \right),$$

$$0 \leq t \leq 1.$$

Свойства составной сплайновой кривой Catmull — Rom:

- проходит точно через опорные точки;
- является геометрически непрерывной;
- набор базовых функций однозначно определяет кривую, т. е. нет возможности регулировать ее форму.

# Интерполяция с помощью кривых Безье

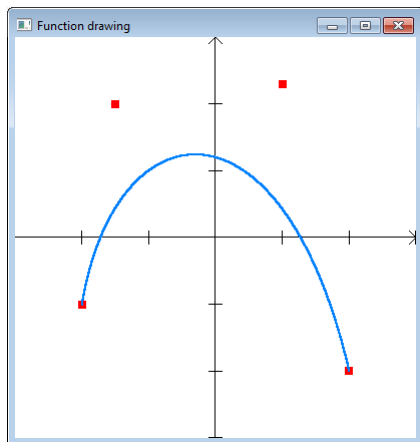


- Сплайны **Катмула-Рома**:

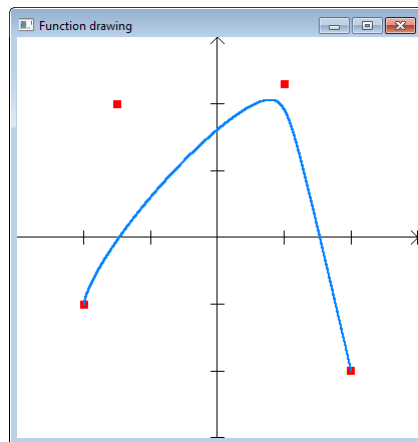
$$P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \frac{1}{2} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

# Рациональные кривые Безье (rational)

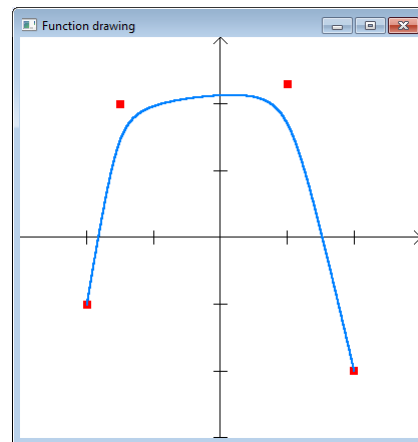
$$B(t) = \frac{\sum_{i=0}^n w_i P_i \cdot \mathbf{b}_{i,n}(t)}{\sum_{i=0}^n w_i \cdot \mathbf{b}_{i,n}(t)}, \quad 0 \leq t \leq 1$$



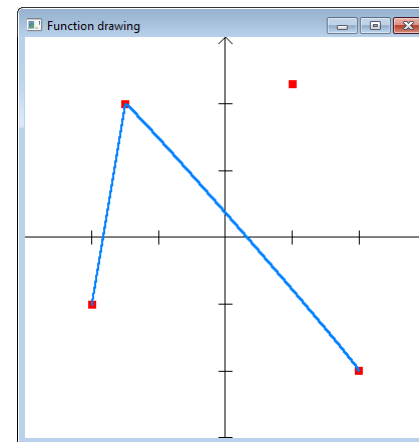
$$w=(1, 1, 1, 1)$$



$$w=(1, 1, 10, 1)$$



$$w=(1, 30, 30, 1)$$



$$w=(1, 1000, 1, 1)$$



# В-сплайны (B-splines)

---

$$P(t) = \sum_{i=1}^{n+1} P_i \cdot N_{i,k}(t), \quad t_{\min} \leq t \leq t_{\max}, \quad 2 \leq k \leq n+1$$

$$N_{i,1} = \begin{cases} 1, & \text{если } x_i \leq t \leq x_{i+1} \\ 0, & \text{иначе} \end{cases}$$

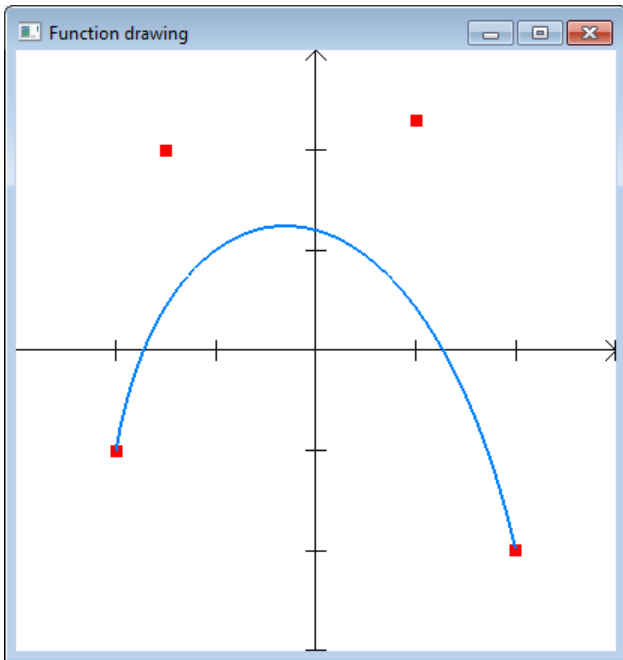
$$N_{i,k} = \frac{(t - x_i) \cdot N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t) \cdot N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}},$$

полагаем  $\frac{0}{0} = 1$  - узловой вектор

$$x = [x_1 \quad x_2 \quad x_3 \quad \dots \quad x_{n+k+1}]$$

# Примеры B-сплайнов (кривая Безье)

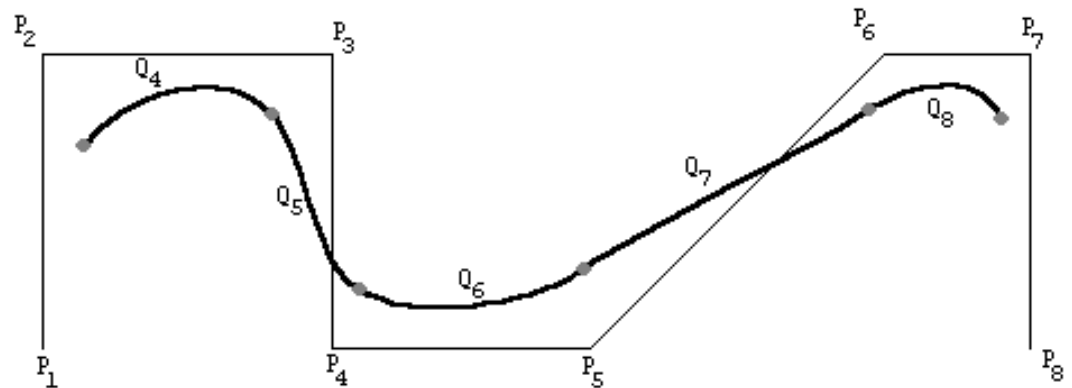
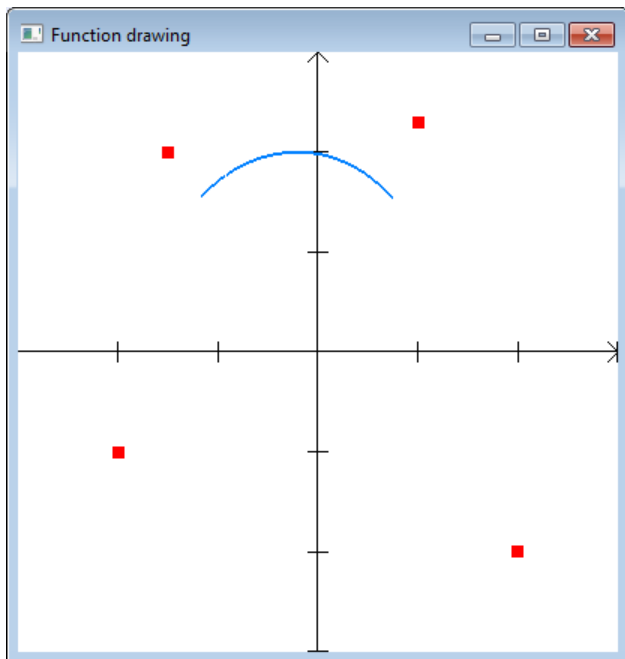
- Кубическая кривая Безье:



$$x = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1]$$

# Примеры B-сплайнов (uniform)

$$P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \frac{1}{6} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_i \\ P_{i+1} \\ P_{i+2} \\ P_{i+3} \end{bmatrix}$$

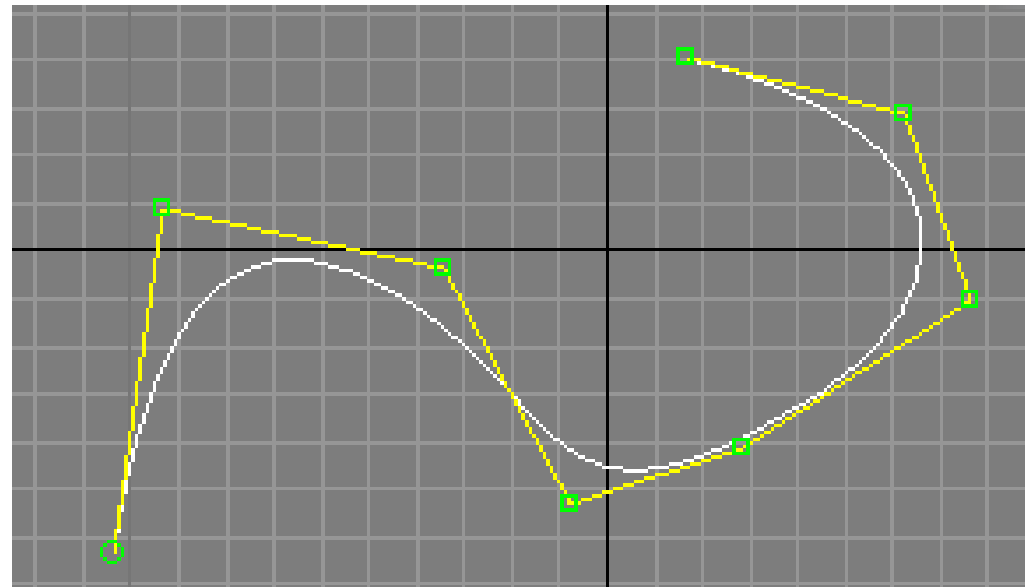




# Примеры В-сплайнов (nonuniform rational)

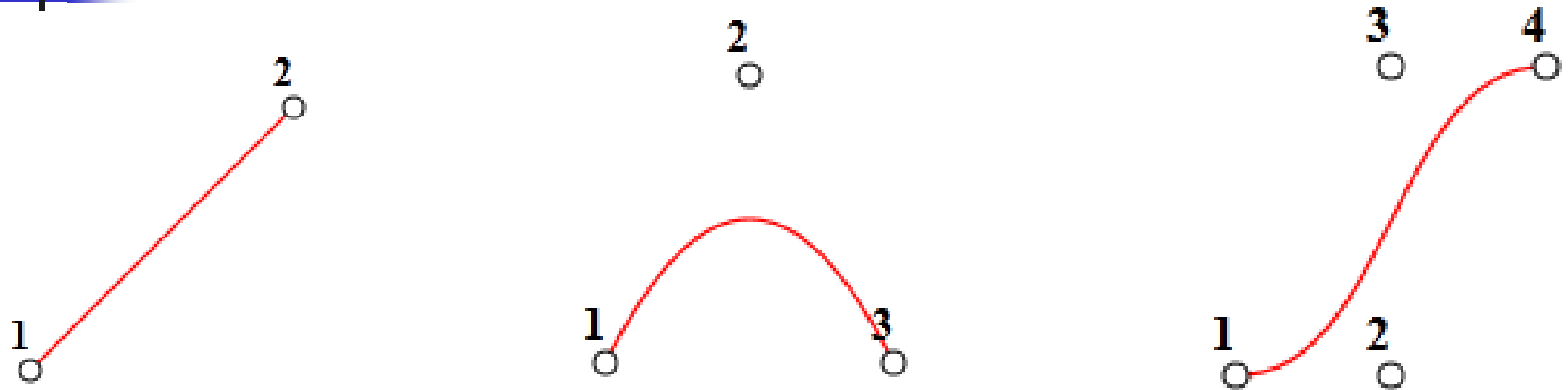
## ■ NURBS

$$P(t) = \frac{\sum_{i=1}^{n+1} w_i P_i \cdot N_{i,k}(t)}{\sum_{i=1}^{n+1} w_i \cdot N_{i,k}(t)}$$



# Кривые Безье –

## для рисования плавных изгибов



Для 2-х точек:  $P = (1 - t)P_1 + tP_2$

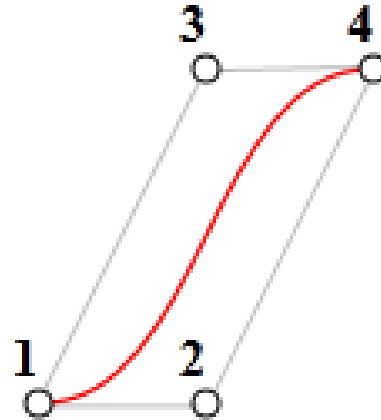
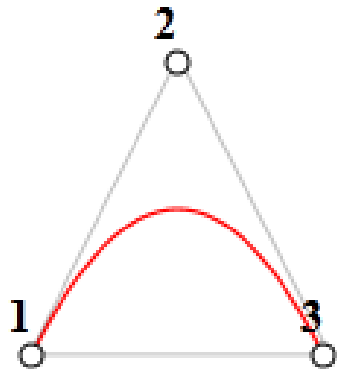
Для 3-х точек:  $P = (1 - t)^2 P_1 + 2(1 - t)tP_2 + t^2P_3$

Для 4-х точек:  $P = (1 - t)^3 P_1 + 3(1 - t)^2 tP_2 + 3(1 - t)t^2 P_3 + t^3 P_4$

$P$  - координаты текущей точки,  $P_i$  - исходных точек,  
 $t$  - шаг  $[0..1]$ , влияет на точность построения.

# Свойства кривых Безье (1)

- **Степень кривой равна числу точек минус один.** На рисунках выше, соответственно, получаются для двух точек — линейная кривая (прямая), для трёх точек — квадратическая кривая (парабола), для четырёх — кубическая.
- **Кривая всегда находится внутри выпуклой оболочки, образованной опорными точками:**



## Свойства кривых Безье (2)

- Кривая натянута по касательным.
- Соединяя несколько кривых, можно получить объект любой формы.





# Лабораторная работа №1

---

«Разработка собственного векторного формата»

# Пример логотипа





# Формулировка задания

---

- Создание формата хранения векторного рисунка (можно в формате XML)(базовые элементы: точка, линия, прямоугольник, эллипс, кривая Безье, текст, толщина и цвет пера).
- Написание парсера, анализирующего предложенный формат и изображающий рисунки 2-х логотипов, согласно вашему варианту. (язык программирования - C#)
- Сохранение рисунков логотипов в растровом формате.
- Анализ плюсов и минусов векторного и растрового способа хранения изображений (в выводах).



# Рисование примитивов

<u><a href="#">DrawArc(Pen, Rectangle, Single, Single)</a></u>	Рисует дугу, которая является частью эллипса, заданного структурой <u><a href="#">Rectangle</a></u> .
<u><a href="#">DrawBezier(Pen, Point, Point, Point, Point)</a></u>	Рисует кривую Безье, определяемую четырьмя структурами <u><a href="#">Point</a></u> .
<u><a href="#">DrawBeziers(Pen, Point[])</a></u>	Рисует последовательность кривых Безье из массива структур <u><a href="#">Point</a></u> .
<u><a href="#">DrawClosedCurve(Pen, Point[])</a></u>	Строит замкнутую фундаментальную кривую, определяемую массивом структур <u><a href="#">Point</a></u> .
<u><a href="#">DrawEllipse(Pen, Rectangle)</a></u>	Рисует эллипс, определяемый ограничивающей структурой <u><a href="#">Rectangle</a></u> .
<u><a href="#">DrawLine(Pen, Point, Point)</a></u>	Проводит линию, соединяющую две структуры <u><a href="#">Point</a></u> .
<u><a href="#">DrawLines(Pen, PointF[])</a></u>	Рисует набор сегментов линии, которые соединяют массив структур <u><a href="#">PointF</a></u> .
<u><a href="#">DrawPath</a></u>	Рисует объект <u><a href="#">GraphicsPath</a></u> .



# Рисование сложных фигур с заполнением

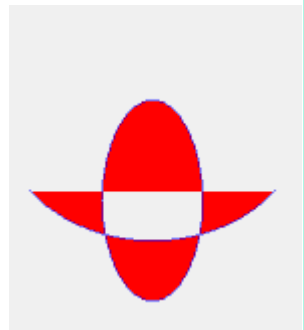
- Класс **GraphicsPath** Представляет последовательность соединенных линий и кривых. Позволяет закрашивать созданную фигуру. Не наследуется.
- **Пространство имен:** System.Drawing.Drawing2D

```
GraphicsPath path = new GraphicsPath();

path.AddArc(0, 0, 150, 120, 30, 120); // Add an open figure.
path.AddEllipse(50, 50, 50, 100); // Add an intrinsically closed figure.

Pen pen = new Pen(Color.FromArgb(128, 0, 0, 255), 1);
SolidBrush brush = new SolidBrush(Color.Red);

// The fill mode is FillMode.Alternate by default.
e.Graphics.FillPath(brush, path);
e.Graphics.DrawPath(pen, path);
```





# Где посмотреть информацию

---

- [https://msdn.microsoft.com/ru-ru/library/System.Drawing\(v=vs.100\).aspx](https://msdn.microsoft.com/ru-ru/library/System.Drawing(v=vs.100).aspx)
- [https://msdn.microsoft.com/ru-ru/library/system.drawing.drawing2d\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.drawing.drawing2d(v=vs.110).aspx)