

## Теоретический минимум к курсу “Компьютерная графика” (2016)

1. Дайте определение цифровому цветному изображению.
2. Дайте определение понятию цвет.
3. Сформулируйте закон аддитивности Грассмана.
4. Опишите принцип работы алгоритма “серый мир”.
5. Опишите принцип работы операции свёртки.
6. Опишите принцип работы медианного фильтра.
7. Сформулируйте метод скользящего окна и критерий обнаружения АО
8. Сформулируйте алгоритм выделения связанных компонент методом последовательного сканирования.
9. Сформулируйте алгоритм кластеризации k-средних.
10. Сформулируйте и опишите формулами метод минимизации эмпирического риска
11. Дайте определение понятиям точности, полноты, кривой точности-полноты и AUC.
12. Опишите алгоритм построения дескриптора изображения на основе нормализованной гистограммы ориентированных градиентов
13. Сформулируйте алгоритм Брезенхема для прямых.
14. Дайте определение сплайну Безье и перечислите его свойства.
15. Сформулируйте алгоритм Де Костельжо для визуализации сплайнов Безье.
16. Перечислите основные этапы графического конвейера.
17. Опишите отличия модельной, мировой, видовой системы координат.
18. Дайте определение октодереву и опишите алгоритм его построения из воксельной модели.
19. Дайте определение понятию конструктивной геометрии.
20. Дайте определение индексированному по вершинам представлению.
21. Опишите различие между первичными и вторичными источниками света.
22. Дайте определение ДФО (двулучевой функции отражения)?
23. Напишите формулу модели освещения Фонга, поясните обозначенные переменные.
24. Дайте определение понятия шейдер.
25. Опишите отличие VBO от VAO.
26. Укажите, в какой момент компилируются шейдеры.
27. Опишите суть метода излучательности.
28. Дайте определение обратной трассировки лучей, перечислите её плюсы и минусы.
29. Перечислите основные шаги рекурсивной трассировки лучей.
30. Опишите идею регулярной сетки как ускоряющей структуры в трассировке лучей.
31. Опишите метод удаления нелицевых граней, укажите тип сцен, для которых он применим.
32. Дайте определение буфера глубины.
33. Дайте определение альфа-канала и поясните за его назначение.
34. Опишите схему работы метода Монте-Карло для вычисления объема (или площади в 2D) сфер

### Ответы - теормин:

#### 1) Дайте определение цифровому цветному изображению.

Цифровое цветное изображение  $X \in \mathbb{R}^{n \times m \times c}$  – трёхмерный массив, в котором для каждого пиксела с координатами  $(x, y)$  хранится описание цвета;  $n, m$  – высота и ширина изображения;  $c$  – число каналов (длина вектора, описывающего цвет пиксела). Обычно каждая компонента дискретизируется в пределах  $[0, 255]$  в 8 бит.

#### 2) Дайте определение понятию цвет

Цвет – это результат взаимодействия **света**, **сцены** и нашей **зрительной системы**.

#### 3) Эмпирический закон о линейности человеческого зрения (закон аддитивности Грассмана)

1. Если наблюдатель задает цвет лучей 1 и 2 как  $(R_1, G_1, B_1)$  и  $(R_2, G_2, B_2)$ , тогда, если мы сложим источники 1 и 2, то мы их можем воспроизвести как  $(R_1+R_2, G_1+G_2, B_1+B_2)$
2. Соответствие цветов выполняется на всех уровнях яркости, т.е., если  $C_1 = (R_1, G_1, B_1)$ , то  $kC_1 = (kR_1, kG_1, kB_1)$

#### 4) Опишите принцип работы алгоритма “серый мир”.

- Средний уровень («серый») по каждому каналу должен быть одинаков для всех каналов
- Если цветовой баланс нарушен, тогда «серый» в этом канале больше «серого» других каналов

- Вычислим коэффициенты усиления так, чтобы среднее в каждом канале стало одинаковым:

$$\bar{R} = \frac{1}{N} \sum R(x, y); \quad \bar{G} = \frac{1}{N} \sum G(x, y); \quad \bar{B} = \frac{1}{N} \sum B(x, y);$$

$$Avg = \frac{\bar{R} + \bar{G} + \bar{B}}{3};$$

$$R' = R \frac{Avg}{\bar{R}}; \quad G' = G \frac{Avg}{\bar{G}}; \quad B' = B \frac{Avg}{\bar{B}}$$

### 5) Опишите принцип работы операции свёртки.

Пусть  $f$  – изображение,  $g$  – ядро. Свертка изображения  $f$  с помощью  $g$  обозначается как  $f * g$  и называется:

$$(f * g)[m, n] = \sum_{k, l} f[m - k, n - l] g[k, l]$$

### 6) Опишите принцип работы медианного фильтра.

*Медианный фильтр* — один из видов цифровых фильтров, широко используемый в обработке изображений **для уменьшения уровня шума**.

Принцип работы: выбор медианы из выборки пикселей в окрестности данного

Пример:

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 10 | 15 | 20 | 10 | 15 | 20 |
| 23 | 90 | 27 | 23 | 27 | 27 |
| 33 | 31 | 30 | 33 | 31 | 30 |

сортировка:

10 15 20 23 **27** 30 31 33 90

перемещение:

### 7) Сформулируйте метод скользящего окна и критерий обнаружения АО (области покрытия) + IoU (11 из 2017)

(“окно” – рамка, ограничивающая часть изображения)

- Возьмём «окно» фиксированного или нескольких размеров
- Сканируем изображение «окном»
- Каждое положение окна – гипотеза, которую нужно классифицировать

**Критерий обнаружения АО** (Area of overlap):

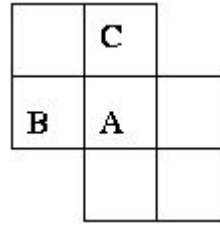
- 1) Нам необходима эталонная коллекция с разметкой
- 2) Если отношение площади области **пересечения эталонной и предсказанной области** к площади **объединения эталонной и предсказанной области** больше чем пороговое значение, то считается, что предсказанная область определена правильно

Критерий: пересечение / объединение > порога

8) Сформулируйте алгоритм выделения связанных компонент методом последовательного сканирования.

!!! Связная область — множество пикселей, у каждого пикселя которого есть хотя бы один сосед, принадлежащий данному множеству.

Последовательно сканируем бинарное изображение сверху вниз, слева направо:



```
if A = 0
    do nothing

else if (not B labeled) and (not C labeled)
    increment (label numbering) and (label A)
else if B xor C labeled
    copy label to A
else if B and C labeled
    if (B label) = (C label)
        copy label to A
    else
        copy either B label or C label to A
        record equivalence of labels
```

После того, как мы обошли всю картинку, нам нужно совершить еще один обход, чтобы провести переразметку с учетом эквивалентности областей

9) Сформулируйте алгоритм кластеризации k-средних.

**Алгоритм кластеризации k-средних** (количество кластеров k — задано):

1. Произвольный выбор k объектов как исходных центров кластеров.
2. Присваивание каждого объекта кластеру (тому, который "ближе" по выбранной метрике)
3. Пересчет кластерных центров
4. Повторение пунктов 2-3 до тех пор, пока на какой-либо итерации кластеры перестанут изменяться

10) Сформулируйте и опишите формулами метод минимизации эмпирического риска

$X^m = \{x_1, \dots, x_m\}$  - обучающая выборка

**Эмпирический риск(ошибка обучения):**

$$R_{emp}(f, X^m) = (1/m) \sum_{i=1}^m L(f(x_i), y_i), \quad L - \text{функция потерь}$$

**Метод минимизации эмпирического риска:**

$$f_0 = \operatorname{argmin}_{f \in F} R_{emp}(f, X^m)$$

**Идея:** подбираем параметры  $\mathbf{t}$  решающего правила  $\mathbf{f}$  таким образом, чтобы эмпирический риск  $R_{emp}$  был минимален

11) Дайте определения: точности, полноты, кривой точности-полноты и AUC.

**Точность (Precision)** - доля истинных объектов основного класса среди всех объектов, классифицированных как основной класс.

**Полнота (Recall)** - доля правильно распознанных объектов основного класса среди всех объектов основного класса из тестовой выборки.

**Кривая точности-полноты** - зависимость между точностью и полнотой в зависимости от параметров.

**AUC (Area Under Curve)** - площадь под графиком — объективный показатель качества, позволяет сравнивать разные кривые.

- F-мера / Интегральный показатель F / F-measure:

$$F = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

## 12) Алгоритм построения дескриптора изображения на основе нормализованной гистограммы ориентированных градиентов:

1) Ищем яркость каждого пикселя.

2) Находим модуль градиента яркости и его направление для каждого пикселя

3) Разбиваем изображение на прямоугольники (клетки) и вычисляем гистограмму градиентов в каждой клетке.

Для начала разобьем всю область изменения направления градиента (например,  $[-\pi, \pi]$ ) на некоторое (обычно не очень большое, порядка 8-32) число сегментов. Выберем некоторую клетку и создадим для неё массив размера, равного количеству сегментов. Для каждого пикселя клетки посчитаем, в какой сегмент попадает направление градиента в этом пикселе и прибавим значение модуля градиента в соответствующую ячейку массива. В результате мы получили гистограмму ориентированных градиентов пикселей выбранной клетки. Повторим процедуру получения гистограммы для всех клеток.

4) Нормализуем гистограммы и объединяем полученные массивы.

## 13) Сформулируйте алгоритм Брезенхема для прямых.

**Алгоритм Брезенхема** - это алгоритм, определяющий, какие точки двумерного раstra нужно закрасить, чтобы получить близкое приближение прямой линии между двумя заданными точками

//plot(x, y) - рисует точку

// int-ы из-за неделимости пикселей

**function** line(x0, x1, y0, y1) // есть две заданные точки

    int deltax := abs(x1 - x0) // расстояние между x

    int deltax := abs(y1 - y0) // расстояние между y

    real error := 0 // ошибка, которая является индикатором правильности принятия решения

    real deltaerr := deltax / deltax // тангенс угла наклона

    int y := y0 // начинаем от y\_0

**for** x **from** x0 **to** x1 // бежим по x

        plot(x,y) // рисуем точку

        error := error + deltaerr // пересчитываем ошибку // цель - понять какая строка ближе всего к линии для каждого столбца(поэтому и бежим по x)

**if** error >= 0.5

            y := y - 1

        error := error - 1.0

## 14) Дайте определение сплайну Безье и перечислите его свойства.

**/\*Определение Сплайна:**

**В компьютерной графике:**

**Составная кривая, сформированная полиномиальными участками (звеньями), которые удовлетворяют заданным условиям непрерывности (гладкости) на границах участков**

*В общем случае:*

Функции для интерполяции или сглаживания данных\*/

Кривая Безье — параметрическая **кривая**, задаваемая выражением

$$\mathbf{B}(t) = \sum_{i=0}^n \mathbf{P}_i \mathbf{b}_{i,n}(t), \quad 0 \leq t \leq 1$$

где  $\mathbf{P}_i$  — функция компонент векторов опорных вершин (координаты опорных точек), а  $\mathbf{b}_{i,n}(t)$  — базисные функции кривой Безье, называемые также **полиномами Бернштейна**.

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

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

где число сочетаний из  $n$  по  $i$ , где

$n$  - степень полинома,  $i$  - порядковый номер опорной вершины.

**Свойства сплайнов Безье:**

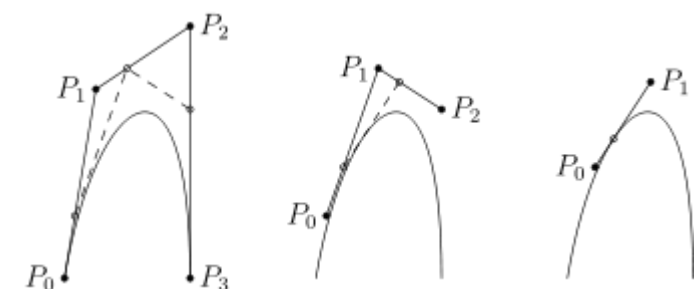
- Проходят через первую и последнюю точку
- Касательная в первой и последней точке проходит вдоль линии, соединяющей крайнюю и ближайшую контрольную точку
- Кривая лежит полностью внутри выпуклой оболочки контрольных точек

**15) Сформулируйте алгоритм Де Кастельжо для визуализации сплайнов Безье.**

Геометрическая интерпретация **алгоритма де Кастельжо** проста:

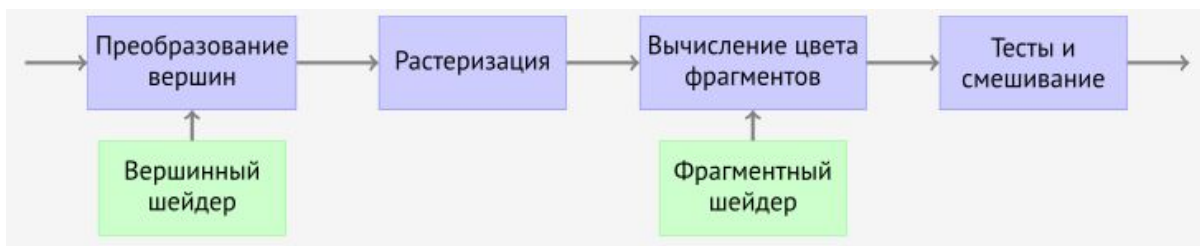
- Задана кривая Безье с опорными точками  $\mathbf{P}_{\{0\}}, \dots, \mathbf{P}_{\{n\}}$ . Соединив последовательно опорные точки с первой по последнюю, получаем ломаную линию.
- Разделяем каждый полученный отрезок этой ломаной в соотношении  $t:(1-t)$  и соединяем полученные точки. В результате получаем ломаную линию с количеством отрезков, меньшим на один, чем исходная ломаная линия.
- Повторяем процесс до тех пор, пока не получим единственную точку. Эта точка и будет являться точкой на заданной кривой Безье с параметром  $t$ .

Следующая иллюстрация демонстрирует этот процесс для кубической кривой Безье:



(демки)

**16) Основные этапы графического конвейера:**



17) Опишите отличия модельной, мировой, видовой системы координат.

**Модельная система координат** - система координат, определенная так, чтобы было удобно определять и ориентировать один объект

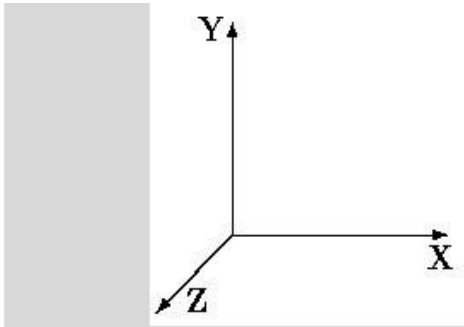


Рис. 5.1. Правосторонняя система координат (мировая)

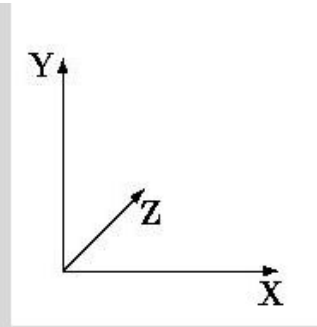


Рис. 5.2. Левосторонняя система координат (видовая)

**В мировой системе координат** располагаются все модели

Далее, для вывода на экран, производится преобразование к системе координат с камерой в центре СК, смотрящей по оси Z - это и есть **видовая система координат**.

18) Дайте определение октодереву и опишите алгоритм его построения из воксельной модели.  
(что запомнится из этого )

Одной из новейших перспективных технологий, позволяющей делать эффективную детализацию воксельных объектов, является **разреженное воксельное октодерево**.

В числе её преимуществ: значительная экономия памяти, естественная генерация уровней детализации и высокая скорость обработки в **рейкастинге**.

**Первый узел дерева** — корень, является кубом, содержащим весь объект целиком. Каждый узел или имеет 8 кубов-потомков, или не имеет никаких потомков. В результате всех подразбиений получается регулярная трёхмерная сетка вокселей.

Трёхмерный объект подразделяется на восемь кубов (октантов). Если какой-либо из октантов является **однородным**, т.е. он располагается либо целиком внутри объекта, либо целиком снаружи, разбиение заканчивается.

Иначе, если октант **не является однородным**, т.е. октант пересекается граничной поверхностью объекта, он разбивается далее на восемь подоктантов.

**Процесс разбиения завершается**, когда все листовые узлы октодеревя станут однородными, возможно с некоторой погрешностью

**Октодерево** — иерархический вариант воксельной модели, рекурсивное разбиение пространства на восемь октант.



## 4 Описание частных алгоритмов

### 4.1 Построение октодера

Построение октодера происходит по известным четырём картам глубин, т.е. уже обработанным данным сканирования.

#### 1. Если делиться нельзя

1.1. Найти для каждой из четырёх плоскостей свои координаты и глубину в соответствии со своим положением.

1.2. Предположить, что воксел существует.

1.3. Проверить на каждой из карт глубин

1.3.1. Если глубина в своих координатах больше или равна собственной, сделать вывод о несуществовании вокселя.

#### 2. Иначе

2.1. Построить каждого из восьми потомков.

2.2. Если каждый из них не существует, сделать вывод о собственном несуществовании.

19) Дайте определение понятию конструктивной геометрии.

**Конструктивная блочная геометрия** (*Constructive Solid Geometry, CSG*) технология, используемая в моделировании твёрдых тел и трехмерной графики. Она позволяет создать сложную сцену или объект с помощью битовых операций для комбинирования нескольких иных объектов.

Простейшие тела, используемые в конструктивной блочной геометрии — **примитивы** (англ. *primitives*), тела с простой формой: куб, цилиндр, призма, пирамида, сфера, конус.

Построение более сложного объекта происходит путём применения к описаниям объектов булевых (двоичных) операций на множествах — объединение, пересечение и разность.

20) Дайте определение индексированному по вершинам представлению.

**Индексированное по вершинам представление** - тип граничного представления (содержит не только каркас, но и грани).

- Выделение координат вершин в отдельную структуру
- С гранями ассоциируются не координаты вершин, а индексы в массиве координат вершин.

21) Опишите различие между первичными и вторичными источниками света.

Тело, испускающее свет, называется **первичным** источником света (или просто источником света). Тело, которое только отражает (преломляет?) падающие на него лучи, называется **вторичным** источником света.

22) Двулучевая Функция Отражения (ДФО):

фактически отношение падающего и исходящего света с учетом направления падения.

**ДФО** - отношение яркости поверхности к ее освещенности.

Яркость поверхности - количество света, излучаемого поверхностью

Освещенность поверхности - количество света, падающего на поверхность

### 23) Модель освещения Фонга:

$$L_o = k_a L_a + L_i (k_d (\vec{s} \cdot \vec{n}) + k_s (\vec{r} \cdot \vec{v})^{k_e})$$

$L_o$  - излучение данной поверхности

$k_a$  - коэффициент **фонового** освещения (ambient)

$L_a$  - **фоновое** излучение

$L_i$  - излучение i-го источника света

$k_d$  - коэффициент **диффузного** освещения (diffuse)

$s$  - направление на источник света

$n$  - нормаль к поверхности

$k_s$  - коэффициент **зеркального** освещения (specular)

$r$  - направление отражения луча от поверхности

$v$  - направление на наблюдателя

$k_e$  - коэффициент блеска, свойство материала

Расчёт освещения по Фонгу требует вычисления цветовой интенсивности трёх компонент освещения: фоновой (ambient), рассеянной (diffuse) и глянцевых бликов (specular).

### 24) Дайте определение понятия шейдер.

**Шейдер** - часть шейдерной программы, которая заменяет собой часть графического конвейера видеокарты. Каждый шейдер должен выполнить свою работу: **записать какие-то данные и передать их дальше по графическому конвейеру.**

**Шейдерная программа** - это программа, состоящая из шейдеров (вершинного и фрагментного, возможны и другие) и выполняющаяся на GPU.

Определение из вики:

**Шейдер** — компьютерная программа, предназначенная для исполнения процессорами видеокарты (GPU).

### 25) Опишите отличие VBO от VAO.

**VBO (Vertex Buffer Objects)** – технология, позволяющая хранить координаты вершин совместно с их атрибутами в видеопамяти.

**VAO (Vertex Arrays Object)** — технология OpenGL, которая определяет, какую часть VBO следует использовать в последующих командах и как эти данные нужно интерпретировать.

### 26) Укажите, в какой момент компилируются шейдеры.

Шейдеры (в OpenGL) компилируются во **время исполнения программы.**

### 27) Опишите суть метода излучательности.

**Основная идея излучательности** – сохранение светимости поверхностей по мере продвижения света от источников (из лекций)

Сцену можно представить как набор поверхностей, обменивающихся лучистой энергией.

**Метод излучательности описывает баланс энергетического равновесия в замкнутой системе.** Предполагается что поверхности идеально диффузные, т. е. после отражения падающий луч пропадает.



Излучательность отдельной поверхности включает самоизлучение и отраженный или пропущенный свет.

## Основная идея метода излучательности

---

- Разбить геометрические модели на маленькие кусочки (патчи)
  - Каждый кусочек освещен равномерно
- Все поверхности диффузны
- Тогда для данной длины волны полное излучение патча будет задаваться скаляром (самосвечение + светимость)
- Алгоритм излучательности создает набор линейных уравнений, связывающих исходящую энергию каждого патча с приходящей энергией на каждый патч
- Результатом процесса является количество излучения каждого патча после достижения равновесия

28) Дайте определение обратной трассировки лучей, перечислите её плюсы и минусы.

**Обратная трассировка лучей** - метод построения изображения, основанный на отслеживании пути каждого луча, попавшего в глаз наблюдателя или объектив камеры **в направлении, противоположном распространению света**

(+) Корректно обрабатывает тени, отражения, преломление, полупрозрачные объекты

(+) Может работать с неполигональными объектами

(-) Большое время работы

(-) Нет возможность вычисления вторичного освещения.

29) Перечислите основные шаги рекурсивной трассировки лучей.

### Рекурсивная трассировка лучей

А. генерируем луч через каждый пиксель изображения

Б. далее необходимо найти пересечение луча с поверхностью сцены

В. необходимо найти точное положение пересечения с объектом

Г. получение цвета пикселя, для лучшего результата бросаем дополнительные лучи (пункт А)

Д. записываем полученный цвет в пиксель и переходим к следующему

30) Идея регулярной сетки как ускоряющей структуры в трассировке лучей:

**Вóксел** — элемент объёмного изображения, содержащий значение элемента раstra в трёхмерном пространстве.

Начинаем с описывающего параллелепипеда, треугольники разбиваются по вокселям. При трассировке лучей луч проходит только через часть вокселей. В данных вокселях проверяется пересечение луча с объектами, которые расположены в данных вокселях (Алгоритм Брезенхема в 3D, останов — найдено пересечение в текущем вокселе).

Простой алгоритм, быстро строится, выгоден для небольших сцен.

### 31) Метод удаления нелицевых граней. Тип сцен, для которых он применим.

Для каждой грани вычисляется внешняя нормаль. Если нормаль смотрит в сторону наблюдателя (наблюдатель находится в положительном полупространстве по отношению к плоскости, проходящей через соответствующую грань), то её оставляем, иначе — удаляем.

**Применимость:** работает только для сплошных тел, удаляет только для сцен с одним выпуклым сплошным объектом, иначе должен использоваться в сочетании с другими методами (чтобы, например, не потерять фотореалистичность, вторичное освещение). Нужно помнить о правой тройке векторов

### 32) Дайте определение буфера глубины.

**Буфер глубины** - двумерный массив, каждый элемент которого соответствует пикселю на экране и равен расстоянию вдоль направления проецирования от картинной плоскости до соответствующей точки пространства.

### 33) Дайте определение альфа-канала и поясните за его назначение.

**Альфа-канал** - дополнительный канал, который может быть добавлен к пикселю. Содержит значение от 0 до 1, кодирует информацию о покрытии пикселя (**0** - нет покрытия, **прозрачный**; **1** - полное покрытие, **непрозрачный**; иначе - частичное покрытие, полупрозрачный).

### 34) Опишите схему работы метода Монте-Карло для вычисления объема (или площади в 2D) сфер

Пусть у нас есть какая-нибудь фигура на плоскости, площадь которой нам необходимо найти. Ограничим ее другой фигурой, площадь которой мы можем легко вычислить. Например, прямоугольником со сторонами, параллельными координатным осям. И **пусть про любую точку прямоугольника мы можем быстро узнать, попадает эта точка внутрь фигуры, площадь которой мы ищем, или нет. Тогда будем бросать на плоскость точки случайным образом в пределах ограничивающей фигуры, причем координаты x и y должны быть распределены равномерно по соответствующим отрезкам. Будем считать количество точек, попавших в фигуру, и общее количество брошенных точек. Поделив количество точек, попавших в фигуру, на общее количество, мы можем найти, какую часть прямоугольника занимает фигура. Количество брошенных точек зависит от точности, которую мы хотим получить (чем больше точек, тем точнее).** (для наглядности посмотрите ниже скрин из лекций

- Методы Монте-Карло решают численно задачу с помощью множественного вычисления случайной величины

- Пример:  $\int_a^b f(x)dx$

- Возьмем случ. величину  $u(x)$

- Плотность  $\phi(x) = \frac{1}{b-a}$

- Мат. ожидание  $Ef(u) = \int_a^b f(x)\phi(x)dx$

- $\int_a^b f(x)dx = (b-a)Ef(u) \Rightarrow$

- $\int_a^b f(x)dx \approx \frac{b-a}{N} \sum_{i=1}^N f(u_i)$

