

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВПО «МГИУ»)  
КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

## КУРСОВАЯ РАБОТА

по дисциплине «Методы хранения и обработки информации»  
на тему «Вычисляется сумма углов, пол которыми рёбра выпуклой  
оболочки пересекают заданную прямую. Вычисляет сумму площадей  
граней, все вершины которых-хорошие точки.»

Группа

2362

Студент

А.Ю. Дьячук

Руководитель работы  
к.ф.-м.н., доцент

Е.А. Роганов

Москва 2012

## Аннотация

Курсовая работа посвящена дополнению проектов «Выпуклая оболочка» и «Изображение проекции полиэдра». В первом из этих проектов решалась задача вычисляет сумму углов, под которыми рёбра выпуклой оболочки пересекают заданную прямую.

## Оглавление

1.	Введение . . . . .	3
2.	Первая модификация проекта «Выпуклая оболочка» . . . . .	3
3.	Модификация проекта «Изображение проекции полиэдра» . . . . .	6

# 1. Введение

В данной курсовой работе рассматриваются модификации двух эталонных проектов «Выпуклая оболочка» и «Изображение проекции полиэдра», реализованных на объектно-ориентированном языке программирования высокого уровня Ruby.

Целями работы являются:

- в проект «Выпуклая оболочка» добавить вычисление суммы внутренних углов выпуклой оболочки;
- «научить» программу определять и выводить сумму углов, под которыми рёбра выпуклой оболочки пересекают заданную прямую в проекте «Выпуклая оболочка»;
- добавить в проект «Изображение проекции полиэдра» вывод суммы площадей граней, всех вершин

Для того чтобы выполнить полученные задания, необходимо было изучить особенности языка Ruby, подробно разобрать каждый эталонный проект и применить полученные знания в области информатики, компьютерной математики и аналитической геометрии на плоскости и в пространстве.

## 2. Первая модификация проекта «Выпуклая оболочка»

### Постановка задачи

Модифицируйте эталонный проект таким образом, чтобы вычислять сумму углов, под которыми рёбра выпуклой оболочки пересекают заданную прямую.

После запуска программа предлагает пользователю ввести последовательно координаты вершин выпуклой оболочки. Введённая точка индуктивно добавляется в выпуклую оболочку. Нам же необходимо вместе со значениями периметра и площади выпуклой оболочки выводить сумму острых внутренних углов. Это задание не требует серьёзной модификации исходного кода, все функции будут вписываться в файл проекта `convex.rb`.

### Решение

Для выполнения задания необходимо вычислять внутренние углы, используя координаты вершин выпуклой оболочки. В языке Ruby имеется примесь `Math`, которая содержит методы вычисления простейших тригонометрических функций, в частности, в данном проекте используется `Math.acos`.

Из аналитической геометрии известно, что угол между векторами  $\overline{BA}$  и  $\overline{BC}$  можно найти по формуле

$$\cos(B) = \frac{\overline{BA} \cdot \overline{BC}}{|\overline{BA}| \cdot |\overline{BC}|}.$$

Для точек  $A(X_1; Y_1)$  и  $B(X_2; Y_2)$  координаты вектора

$$\overline{AB} = \{X_2 - X_1; Y_2 - Y_1\}. \quad (1)$$

Если  $\overline{BA}\{X_1; Y_1\}$  и  $\overline{BC}\{X_2; Y_2\}$ , то скалярное произведение этих векторов  $\overline{BA} \cdot \overline{BC} = X_1 X_2 + Y_1 Y_2$ , а длина вектора  $\sqrt{\overline{BA}^2} = |\overline{BA}| = \sqrt{X_1^2 + Y_1^2}$ .

Окончательная формула для угла В между векторами  $\overline{BA}$  и  $\overline{BC}$  выглядит так:

$$\cos(B) = \frac{X_1 X_2 + Y_1 Y_2}{\sqrt{X_1^2 + Y_1^2} \cdot \sqrt{X_2^2 + Y_2^2}}. \quad (2)$$

Метод `Math.acos` возвращает угол в радианах. Так как решено выводить ответ в градусах, воспользуемся формулой  $\alpha^\circ = \alpha \times \frac{180}{\pi}$ .

## Модификация кода

```
def get_sum
  return 0.0 unless self.respond_to? :points
  pts = self.points
  p pts
  s = 0.0
  pts.size.times do |i|
    p1, p2 = pts[i-1], pts[i] #находим вершины ребра
    m = cross_point $points[0], $points[1], p1, p2
    if m
      cp = (m == p1) ? p2 : p1
      cl = (m == $points[0]) ? $points[1] : $points[0]
      s += angle(cp, m, cl).abs unless cp == cl
    end
  end
end
s
end
```

находит уравнение прямой, заданной двумя точками

```
def equation_from_segment(p1, p2)
  # Ax + By + C = 0
  x1, y1 = p1.x, p1.y
  x2, y2 = p2.x, p2.y

  a = y1 - y2
  b = x2 - x1
  c = x1 * y2 - x2 * y1
  [a, b, c]
end
```

точка пересечения линии АВ и отрезка PQ

```
def cross_point(a, b, p, q)
  a1, b1, c1 = equation_from_segment(a, b)
  a2, b2, c2 = equation_from_segment(p, q)

  d = (a1*b2 - a2*b1)
  # Если прямые не пересекаются - завершаем работу, отрезки тоже не будут пересекаться
```

```

return false if d == 0

# Находим координаты пересечения прямых
x = -(c1*b2-c2*b1).to_f / d
y = -(a1*c2-a2*c1).to_f / d

# Точка пересечения прямых должна принадлежать отрезку PQ
m = R2Point.new(x, y)
return false unless m.inside?(p, q)
  m
end

```

ищем угол B

```

def angle (a, b, c)
  # координаты векторов
  ba_x = a.x - b.x
  ba_y = a.y - b.y
  bc_x = c.x - b.x
  bc_y = c.y - b.y

  dot_product = ba_x*bc_x + ba_y*bc_y # скалярное произведение
  module_ba = ba_x**2 + ba_y**2
  module_bc = bc_x**2 + bc_y**2
  cos = dot_product / Math.sqrt(module_ba * module_bc)
  cos *= -1 if cos < 0
return Math.acos(cos)
end

```

Увидеть работу этой модификации можно на рис. 1.



Рис. 1. Ввод координат

Увидеть работу этой модификации можно на рис. 2.

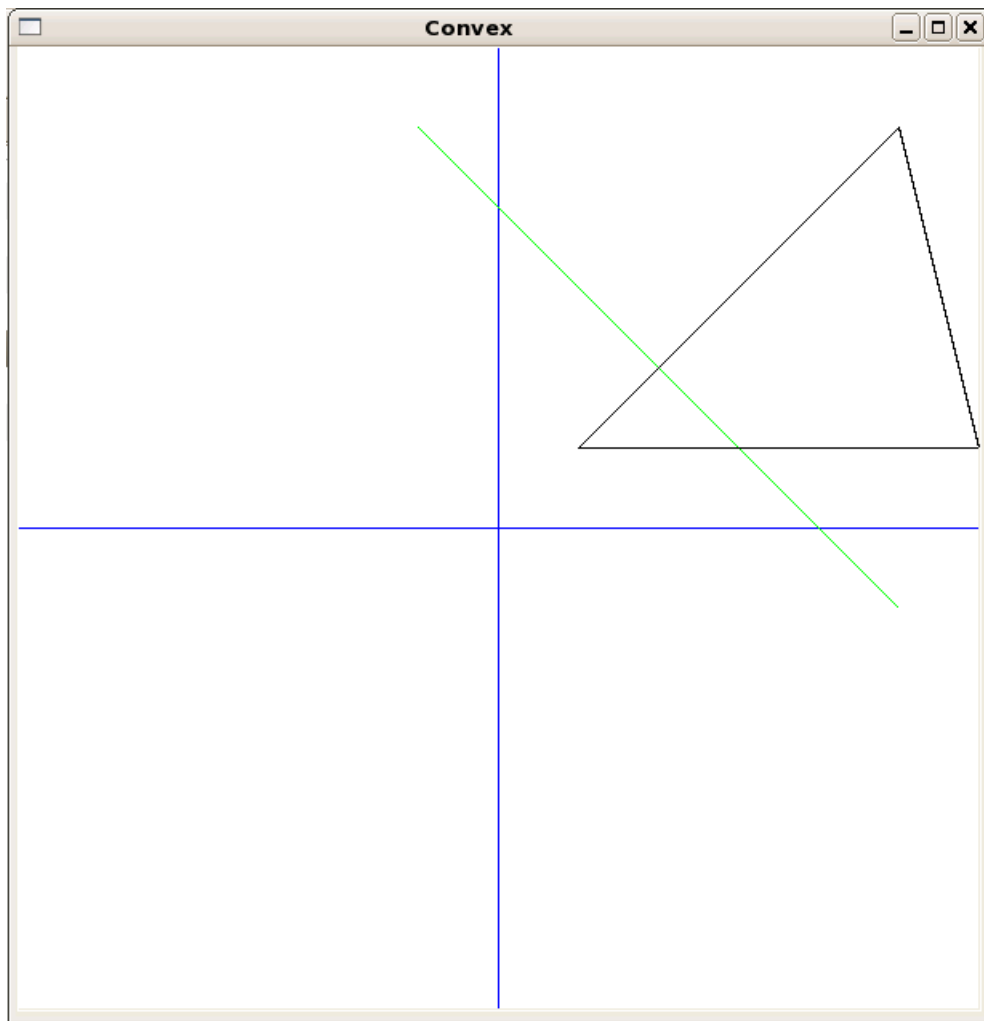


Рис. 2. В результате ввода координат получаем следующее изображение

### 3. Модификация проекта «Изображение проекции полиэдра»

#### Постановка задачи

Модифицируйте эталонный проект таким образом, чтобы определялась и печаталась следующая характеристика полиэдра: сумма площадей граней, все вершины которых-хорошие точки.

После запуска программы, программа вычисляет площадь хороших полигонов. Это задание не требует серьёзной модификации исходного кода, все функции будут вписываться в файл проекта `polyedr.rb`.

#### Решение

Для выполнения задания необходимо определить хорошую точку в пространстве, если она строго находится вне сферы  $x^2 + y^2 + z^2 = 1$ . И вычислить сумму площадей граней, все вершины которые-хорошие точки.

## Модификация кода

Создадим метод модуля.

```
def length
  Math.sqrt(@x*@x + @y*@y + @z*@z)
end
```

Создадим метод хорошей вершины. Вершина называется хорошей если она лежит за пределами сферы.

```
def good?
  @x*@x + @y*@y + @z*@z > 1#
end
```

Создадим метод площади грани.

```
def area#метод площади грани
  result = 0.0

  for i in 1 ... (vertexes.size - 1)
    result += triangle_area(vertexes[0], vertexes[i], vertexes[i + 1])
  end

  return 0.5*result
end
```

Создадим метод который вычисляет площадь треугольника.

```
def triangle_area(a, b, c)
  ((b - a).v(c - a)).length
end
```

Создадим метод который вычисляет площадь грани,если она является хорошей.

```
def good_area
  result = 0.0

  facets.each do |face|
    result += face.area if face.good?
  end

  return result
end
```

Увидеть работу этой модификации можно на рис. 3.

```
geany_run_script.sh
=====
Начало работы с полиэдром '1'
Изображение полиэдра '1' заняло 0,051466337 сек.
Площадь хороших полигонов 54721,35954999579.
Hit "Return" to continue ->
=====
Начало работы с полиэдром '2'
Изображение полиэдра '2' заняло 0,00450301 сек.
Площадь хороших полигонов 49999,99999999999.
Hit "Return" to continue ->
=====
Начало работы с полиэдром 'ссс'
Изображение полиэдра 'ссс' заняло 0,005047076 сек.
Площадь хороших полигонов 40000,00000000001.
Hit "Return" to continue ->
=====
Начало работы с полиэдром 'cube'
Изображение полиэдра 'cube' заняло 0,007267484 сек.
Площадь хороших полигонов 240000,0.
Hit "Return" to continue -> █
```

Рис. 3. Начало работы с полиэдром 'cube'

Увидеть работу этой модификации можно на рис. 4.

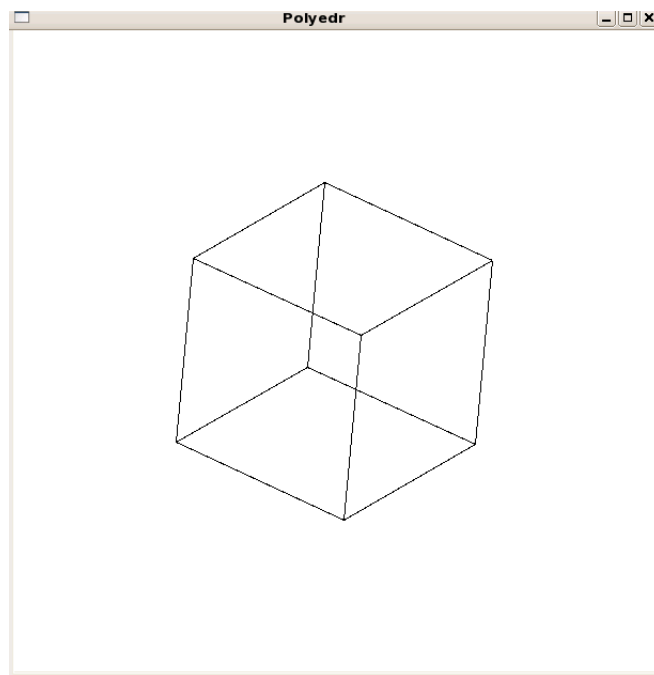


Рис. 4. Изображение куба



## Список литературы и интернет-ресурсов

- [1] <http://ru.wikipedia.org/wiki/LaTeX> — Википедия (свободная энциклопедия) о системе  $\text{\LaTeX}$ .
- [2] <http://www.mgena.chat.ru/latex/indru.html> — Система  $\text{\TeX}$ / $\text{\LaTeX}$ (конспект).
- [3] <http://edu.msiu.ru/files/571-convex.zip> — Исходные файлы эталонного проекта «Выпуклая оболочка». Доступны только при авторизации на образовательном портале МГИУ.
- [4] <http://edu.msiu.ru/files/1821-polyedr.zip> — Исходные файлы эталонного проекта «Изображение проекции полиэдра». Доступны только при авторизации на образовательном портале МГИУ.
- [5] М.Я. Выгодский. *Справочник по высшей математике*. — М., АСТ:Астрель, 2008.