

СИСТЕМА КОМПЬЮТЕРНОЙ ГЕОМЕТРИИ

*Баранкевич Алина Александровна, студентка 1-го курса
кафедры «Математические методы в строительстве»
Белорусский национальный технический университет, г. Минск
(Научный руководитель — Дыдышко П.А., ассистент)*

Работа посвящена построению компьютерной модели объектов аналитической геометрии с целью последующего построения системы компьютерной геометрии. Целью работы является создание набора базовых объектов и функций системы, которые позволили бы решать задачи аналитической геометрии на плоскости.

Для реализации системы был выбран язык программирования Julia, как наиболее подходящий по ряду факторов:

- Простота в освоении
- Высокая выразительность языка, приближенная к математической
- Интерактивность
- Визуализируемость
- Высокая производительность

Базовыми объектами аналитической геометрии на плоскости являются точки и векторы. Далее с их помощью создаются более сложные объекты: прямые, отрезки, многоугольники и т.д.

В данной работе будут рассмотрены точки, векторы, прямые, а также некоторые операции над ними.

В аналитической геометрии точки и векторы задаются набором координат:

```
struct GPoint
    x::Float64
    y::Float64
end
```

```
struct Gvector
    x::Float64
    y::Float64
end
```

Теперь определим основные операции над точками и векторами:

```

Base.:*(a::Real, v::GVector) = GVector(a*v.x, a*v.y)
Base.:+(u::GVector, v::GVector) = GVector(u.x+v.x,
u.y+v.y)
Base.:+(p::GPoint, v::GVector) = GPoint(p.x+v.x, p.y+v.y)
Base.:-(m::GPoint, n::GPoint) = GVector(m.x-n.x, m.y-n.y)
Base.:*(u::GVector, v::GVector) = u.x*v.x + u.y*v.y
norm(v::GVector) = sqrt(v*v)
angle(u::GVector, v::GVector) = acos(u*v/norm(u)/norm(v))

```

Теперь с помощью смоделированных объектов решим задачу нахождения площади и периметра треугольника, заданного координатами вершин:

```

perimeter(A::GPoint, B::GPoint, C::GPoint) =
    norm(B - A) + norm(C - B) + norm(A - C)
function area(A::GPoint, B::GPoint, C::GPoint)
    u = B - A
    v = C - A
    sqrt(abs(u.x*v.y - u.y*v.x))
end

```

И задачу определения взаимного расположения прямых:

```

struct GLine
    p::GPoint # точка на прямой
    v::GVector # направляющий вектор
end

function position(l1::GLine, l2::GLine; eps = 1e-5)
    if angle(l1.v, l2.v) > eps
        "Пересекаются"
    elseif angle(l1.v, l1.p - l2.p) < eps ||
        π - angle(l1.v, l1.p - l2.p) < eps
        "Совпадают"
    else
        "Параллельны"
    end
end

```

Литература:

1. Документация к языку программирования Julia [Электронный ресурс]. – Режим доступа: <https://docs.julialang.org>. – Дата доступа: 15.05.2022