

Рекомендуемый интерфейс для решения задач вычислительной геометрии

```
const double EPS = 1e-9;

typedef double Angle;

struct Point {
    int x, y;
};

struct PolarPoint {
    //phi in [0..2 * PI)
    double r, phi;
};

struct Vector {
    int x, y;
};

struct Segment {
    Point a, b;
};

struct Ray {
    Point a, b;
};

struct Line {
    Point a, b;
};

struct Triangle {
    Point a, b, c;
};

struct Circle {
    Point c;
    int r;
};

struct Polygon {
    vector<Point> p;
};

Point createPoint(int x, int y);
Vector createVector(Point a, Point b);
Segment createSegment(Point a, Point b);
Ray createRay(Point a, Point b);
Line createLine(Point a, Point b);
Triangle createTriangle(Point a, Point b, Point c);
Circle createCircle(Point c, int r);
Polygon createPolygon(vector<Point> p);
```

```
void printPoint(Point p);
void printVector(Vector v);

double vectorLength(Vector a);
double pointDistance(Point a, Point b);
double distance(Segment s, Point a);
double distance(Ray ray, Point a);
double distance(Line line, Point a);
double segmentDistance(Segment a, Segment b);

Vector sum(Vector a, Vector b);
Vector mult(Vector a, int b);
Vector normalize(Vector a);

int dotProduct(Vector a, Vector b);
int crossProduct(Vector a, Vector b);

Point polarToCartesian(PolarPoint a);
PolarPoint cartesianToPolar(Point a);

bool isIntersected(Segment a, Segment b);
bool isIntersected(Ray a, Ray b);
bool isIntersected(Line a, Line b);

bool isBelonged(Segment s, Point a);
bool isBelonged(Ray ray, Point a);
bool isBelonged(Line line, Point a);

bool isInside(Triangle t, Point a);
bool isInside(Circle c, Point a);
bool isInside(Polygon p, Point a);
bool isConvex(Polygon p);

bool isOrthogonal(Vector a, Vector b);

double triangleArea(Triangle a);
double circleArea(Circle a);
double polygonArea(Polygon a);

double getAngle(Vector a, Vector b);

bool equal(double a, double b);
bool less(double a, double b);
bool greater(double a, double b);
```