КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

КАФЕДРА ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ

Отчёт по учебной (ознакомительной) практике

*Создание приложения на языке программирования Java*

Выполнил студент*(ка)*

Гр.09-822*(2)*

*Шабаков Ильвар*

Проверил*(а)*

*Жуманиёзов А. Р.*

Казань 2018

**Тема:** создание приложений на языке программирования Java.

**Задание:** используя язык программирования Java создать приложение, решающее поставленную задачу.

**Реализация:** Задача заключалась в том, чтобы оперировать фигурами, созданными следующим образ. Создается интерфейс(IShape), в который включаются методы(square,length,shift,rot,symaxis,cross), от которого наследуются 2 типа фигур:незамкнутые(OpenFigure) и замкнутые(NGon). Вся суть задачи в том, чтобы переопределить методы для каждой из фигур. А в main'е мы, используя данные функции, можем управлять фигурами(совершать сдвиги на вектор/поворачивать фигуры/создавать симметрию/и в конце концов проверять на пересечение). Оперируем с такими фигурами, как Круг, Отрезок, Ломанная, Треугольник, Четырехугольник, Трапеция**.**

**Листинг кода**

*MAIN*

package com.company;import java.util.\*;public class Main{ public static void main(String[] args) { String buf; double S = 0, L = 0; Scanner in = new Scanner(System.*in*); int n = in.nextInt(); IShape[] s1 = new IShape[n]; String[] m = new String[n]; for (int i = 0; i < n; i++) { m[i] = in.next(); s1[i] = *Parse*(m[i]); if (s1[i] == null) { System.*out*.println("Фигура неизвестна"); i--; continue; } S += Math.*abs*(s1[i].Square()); L += Math.*abs*(s1[i].Length()); } System.*out*.println("Общая площадь= " + S); System.*out*.println("Средняя площадь= " + S / n); System.*out*.println("Общая длина= " + L); System.*out*.println("Средняя длина= " + L / n); IShape[] s2 = new IShape[n]; for (int i = 0; i < n; i++) { s2[i] = *Parse*(m[i]); for (int j = 0; j < n; j++) System.*out*.println("Суждение о том, пересекаются ли фигуры 1." + j + " и 2." + i + " " + s2[i].Cross(s1[j])); buf = in.next(); if (buf.equals("0")) { s2[i] = s2[i].Shift(new Point2D(new double[]{in.nextDouble(), in.nextDouble()})); } else if (buf.equals("1")) { s2[i] = s2[i].Rot(in.nextDouble()); } else if (buf.equals("2")) { s2[i] = s2[i].Symaxis(in.nextInt()); } System.*out*.println("Вот что вышло:"); System.*out*.println(s2[i]); System.*out*.println(); System.*out*.println("Суждение о том, пересекаются ли фигуры 1." + i + " и 2." + i + " " + s2[i].Cross(s1[i])); } } public static IShape Parse(String s) { Scanner in = new Scanner(System.*in*); if (s.equals("0")) return new Circle(new Point2D(new double[]{in.nextDouble(), in.nextDouble()}), in.nextDouble()); else if (s.equals("1")) return new Section(new Point2D(new double[]{in.nextDouble(), in.nextDouble()}), new Point2D(new double[]{in.nextDouble(), in.nextDouble()})); else if (s.equals("2")) { int n = in.nextInt(); Point2D[] array = new Point2D[n]; for (int i = 0; i < n; i++) array[i] = new Point2D(new double[]{in.nextDouble(), in.nextDouble()}); return new Polyline(array, n); } else if (s.equals("3")) { int n = in.nextInt(); Point2D[] array = new Point2D[n]; for (int i = 0; i < n; i++) array[i] = new Point2D(new double[]{in.nextDouble(), in.nextDouble()}); return new NGon(n, array); } else if (s.equals("4")) { Point2D[] array = new Point2D[3]; for (int i = 0; i < 3; i++) array[i] = new Point2D(new double[]{in.nextDouble(), in.nextDouble()}); return new TGon(array); } else if (s.equals("5")) { Point2D[] array = new Point2D[4]; for (int i = 0; i < 4; i++) array[i] = new Point2D(new double[]{in.nextDouble(), in.nextDouble()}); return new QGon(array); } else if (s.equals("6")) { Point2D[] array = new Point2D[4]; for (int i = 0; i < 4; i++) array[i] = new Point2D(new double[]{in.nextDouble(), in.nextDouble()}); return new Rectangle(array); } else if (s.equals("7")) { Point2D[] array = new Point2D[4]; for (int i = 0; i < 4; i++) array[i] = new Point2D(new double[]{in.nextDouble(), in.nextDouble()}); return new Trapeze(array); } else return null; }} *ISHAPE*

package com.company;public interface IShape{ double Square(); double Length(); public IShape Shift(Point2D a); public IShape Rot(double phi); public IShape Symaxis(int i); public boolean Cross(IShape i);}

*OPENFIGURE*

package com.company;import static com.company.Point.\*;public abstract class OpenFigure implements IShape{ public double Square() { return 0; } public abstract double Length(); public abstract OpenFigure Shift(Point2D a); public abstract OpenFigure Rot(double phi); public abstract OpenFigure Symaxis(int i); public abstract boolean Cross(IShape i);}

*NGON*

package com.company;public class NGon implements IShape { Point2D[] array; int n; NGon(int n, Point2D[] array) { this.array = array; this.n = n; } @Override public double Square() { double S = array[n - 1].GetX(0) \* array[0].GetX(1) - array[0].GetX(0) \* array[n - 1].GetX(1); for (int i = 0; i < n - 1; i++) S += array[i].GetX(0) \* array[i + 1].GetX(1) - array[i + 1].GetX(0) \* array[i].GetX(1); return S / 2; } @Override public double Length() { double S = Point.*Sub*(array[0], array[n - 1]).Abs(); for (int i = 0; i < n - 1; i++) { S += Point.*Sub*(array[i], array[i + 1]).Abs(); } return S; } @Override public IShape Shift(Point2D a) { NGon buf = new NGon(this.n, this.array); for (int i = 0; i < n; i++) buf.array[i] = (Point2D) Point.*Add*(buf.array[i], a); return buf; } @Override public IShape Rot(double phi) { NGon buf = new NGon(this.n, this.array); for (int i = 0; i < n; i++) buf.array[i] = (Point2D) buf.array[i].Rot(phi); return buf; } @Override public IShape Symaxis(int i) { NGon buf = new NGon(this.n, this.array); for (int j = 0; j < n; j++) buf.array[j] = (Point2D) buf.array[j].Simm(i); return buf; } @Override public boolean Cross(IShape a) { NGon ngon = (NGon) a; for (int i = 0; i < this.n - 1; i++) { for (int j = 0; j < ngon.n - 1; j++) { if (new Section(this.array[i], this.array[i + 1]).Cross(new Section(ngon.array[j], ngon.array[j + 1]))) return true; } if (new Section(this.array[i], this.array[i + 1]).Cross(new Section(ngon.array[ngon.n - 1], ngon.array[0]))) return true; } for (int j = 0; j < ngon.n - 1; j++) { if (new Section(this.array[this.n - 1], this.array[0]).Cross(new Section(ngon.array[j], ngon.array[j + 1]))) return true; } if (new Section(this.array[this.n - 1], this.array[0]).Cross(new Section(ngon.array[ngon.n - 1], ngon.array[0])) || this.IsBelongs(ngon.array[0]) || ngon.IsBelongs(array[0])) return true; return false; } public boolean IsBelongs(Point2D a) { for (int i = 0; i < n - 1; i++) if (new Section(array[i], array[i + 1]).WIP(a) != new Section(array[i], array[i + 1]).WIP(array[n - 1])) return false; if (new Section(array[n - 2], array[n - 1]).WIP(a) == new Section(array[n - 2], array[n - 1]).WIP(array[0]) && new Section(array[0], array[n - 1]).WIP(a) == new Section(array[0], array[n - 1]).WIP(array[1])) return true; else return false; } public String toString() { String s = "{"; for (Point2D buf : array) s += buf.toString() + ","; return s.substring(0, s.length() - 1) + "}"; }}

*POINT*

package com.company;public class Point{ protected int dim; protected double[] x; @Override public String toString() { StringBuilder P = new StringBuilder(); P.append("("); for (int i = 0; i < dim - 1; i++) { P.append(x[i]).append(";"); } P.append(x[dim - 1]).append(")"); return P.toString(); } public Point(int dim) { this.dim = dim; this.x = new double[dim]; } public Point(int dim, double[] x) { this.dim = dim; this.x = x.clone(); } public int GetDim() { return dim; } public double[] GetX() { return x; } public double GetX(int i) { return x[i]; } public void SetX(double[] x) { this.x = x; } public void SetX(int i, double x) { this.x[i] = x; } public double Abs() { double a = 0; for (double i : x) { a += i \* i; } return Math.*sqrt*(a); } public Point Add(Point B) { if (dim != B.GetDim()) { System.*out*.println("Точки несовместимы"); } double[] x = new double[dim]; for (int i = 0; i < dim; i++) { x[i] = this.x[i] + B.GetX(i); } return new Point(dim, x); } public static Point Add(Point A, Point B) { if (A.GetDim() != B.GetDim()) { System.*out*.println("Точки несовместимы"); } Point buf = A.clone(); for (int i = 0; i < A.GetDim(); i++) { buf.SetX(i, A.GetX(i) + B.GetX(i)); } return buf; } public Point Sub(Point B) { if (dim != B.GetDim()) { System.*out*.println("Точки несовместимы"); } double[] x = new double[dim]; for (int i = 0; i < dim; i++) { x[i] = this.x[i] - B.GetX(i); } return new Point(dim, x); } public static Point Sub(Point A, Point B) { if (A.GetDim() != B.GetDim()) { System.*out*.println("Точки несовместимы"); } double[] x = new double[A.GetDim()]; for (int i = 0; i < A.GetDim(); i++) { x[i] = A.GetX(i) - B.GetX(i); } return new Point(A.GetDim(), x); } public Point Mult(double r) { double[] x = new double[dim]; for (int i = 0; i < dim; i++) { x[i] = this.x[i] \* r; } return new Point(dim, x); } public static Point Mult(double r, Point A) { double[] x = new double[A.GetDim()]; for (int i = 0; i < A.GetDim(); i++) { x[i] = A.GetX(i) \* r; } return new Point(A.GetDim(), x); } public double Multi(Point B) { if (dim != B.GetDim()) { System.*out*.println("Точки несовместимы"); } double a = 0; for (int i = 0; i < dim; i++) { a += x[i] \* B.GetX(i); } return a; } public static double Multi(Point A, Point B) { if (A.GetDim() != B.GetDim()) { System.*out*.println("Точки несовместимы"); } double a = 0; for (int i = 0; i < A.GetDim(); i++) { a += A.GetX(i) \* B.GetX(i); } return a; } public Point Simm(int n) { Point buf = this.clone(); for (int i = 0; i < dim; i++) { buf.SetX(i, -this.x[i]); } buf.SetX(n, -this.x[n]); return buf; } public static double leng(Point A, Point B) { return (Point.*Sub*(A, B).Abs()); } public Point clone() { return new Point(this.dim, this.x); }}

*POINT2D*

package com.company;public class Point2D extends Point{ public Point2D() { super(2); } public Point2D(double[] x) { super(2, x); } public Point Rot(double a) { double[] x = new double[2]; x[0] = this.x[0] \* Math.*cos*(a) - this.x[1] \* Math.*sin*(a); x[1] = this.x[0] \* Math.*sin*(a) + this.x[1] \* Math.*cos*(a); return new Point2D(x); } public Point2D clone() { return new Point2D(this.x); }}

*POLYLINE*

package com.company;public class Polyline extends OpenFigure{ Point2D[] array; int n; Polyline(Point2D[] array, int n) { this.array = array; this.n = n; } public double Length() { double S = 0; for (int i = 0; i < n - 1; i++) S += Point2D.*leng*(array[i], array[i + 1]); return S; } public OpenFigure Shift(Point2D a) { Polyline buf = new Polyline(array, n); for (int i = 0; i < n; i++) buf.array[i] = (Point2D) buf.array[i].Add(a); return buf; } public OpenFigure Rot(double phi) { Polyline buf = new Polyline(array, n); for (int i = 0; i < n; i++) buf.array[i] = (Point2D) buf.array[i].Rot(phi); return buf; } public OpenFigure Symaxis(int i) { Polyline buf = new Polyline(array, n); for (int j = 0; j < n; j++) buf.array[j] = (Point2D) buf.array[j].Simm(i); return buf; } public boolean Cross(IShape a) { Polyline pol = (Polyline) a; for (int i = 0; i < this.n - 1; i++) for (int j = 0; j < pol.n - 1; j++) if (new Section(this.array[i], this.array[i + 1]).Cross(new Section(pol.array[j], pol.array[j + 1]))) return true; return false; }}

*SECTION*

package com.company;public class Section extends OpenFigure{ Point2D st, f; Section(Point2D st, Point2D f) { this.st = st; this.f = f; } public String toString() { return (st.x[0] + ";" + st.x[1]) + " " + (f.x[0] + ";" + f.x[1]); } public Point2D getF() { return f; } public void setF(Point2D f) { this.f = f; } public Point2D getSt() { return st; } public void setSt(Point2D st) { this.st = st; } @Override public double Length() { return Point.*Sub*(st, f).Abs(); } @Override public Section Shift(Point2D a) { Section buf = new Section(st, f); buf.st = (Point2D) buf.st.Add(a); buf.f = (Point2D) buf.f.Add(a); return buf; } @Override public OpenFigure Rot(double phi) { Section buf = new Section(st, f); buf.st = (Point2D) buf.st.Rot(phi); buf.f = (Point2D) buf.f.Rot(phi); return buf; } @Override public OpenFigure Symaxis(int i) { Section buf = new Section(st, f); buf.st = (Point2D) buf.st.Simm(i); buf.f = (Point2D) buf.f.Simm(i); return buf; } @Override public boolean Cross(IShape i) { Section sec = (Section) i; double k1 = (f.x[1] - st.x[1]) / (f.x[0] - st.x[0]), c1 = this.st.GetX(1) - this.st.GetX(0) \* k1; double k2 = (sec.f.x[1] - sec.st.x[1]) / (sec.f.x[0] - sec.st.x[0]), c2 = sec.st.GetX(1) - sec.st.GetX(0) \* k2; if (k1 != k2) { Point2D cross = new Point2D(new double[]{(c2 - c1) / (k1 - k2), (c2 - c1) / (k1 - k2) \* k1 + c1}); if (Point.*Sub*(this.st, cross).Abs() < this.Length() && Point.*Sub*(this.f, cross).Abs() < this.Length() && Point.*Sub*(sec.st, cross).Abs() < sec.Length() && Point.*Sub*(sec.f, cross).Abs() < sec.Length()) return true; } return false; } //Where Is Point public short WIP(Point2D a) { double k = (f.x[1] - st.x[1]) / (f.x[0] - st.x[0]), c = this.st.GetX(1) - this.st.GetX(0) \* k; if (a.GetX(1) > k \* a.GetX(0) + c) return 1; else if (a.GetX(1) < k \* a.GetX(0) + c) return -1; else return 0; }}

*QGON*

package com.company;import java.util.Scanner;public class QGon extends NGon { QGon(Point2D[] k) { super(4, k); } @Override public double Square() { double P=(Point.*Sub*(array[0], array[1]).Abs()+Point.*Sub*(array[1],array[2]).Abs()+Point.*Sub*(array[2], array[3]).Abs()+Point.*Sub*(array[3], array[0]).Abs())/2; double A=Point.*Sub*(array[0], array[1]).Abs(); double B=Point.*Sub*(array[1],array[2]).Abs(); double C=Point.*Sub*(array[2], array[3]).Abs(); double D=Point.*Sub*(array[3], array[0]).Abs(); /\*double angle1=Math.toDegrees(Math.atan2(array[3].GetX(1)-array[1].GetX(1),array[3].GetX(0)-array[1].GetX(0))); double angle2=Math.toDegrees(Math.atan2(array[2].GetX(1)-array[0].GetX(1),array[2].GetX(0)-array[0].GetX(0))); PROVERKA \*/ double angle3=Math.*toDegrees*(Math.*atan2*(A,B)); double angle4=Math.*toDegrees*(Math.*atan2*(C,D)); return Math.*sqrt*((P-A)\*(P-B)\*(P-C)\*(P-D)-(A\*B\*C\*D\*(Math.*cos*((angle3+angle4)/2))\*(Math.*cos*((angle3+angle4)/2)))); /\* SPOSOB NOMER 2 double d1=Point.Sub(array[1],array[3]).Abs(); double d2=Point.Sub(array[0],array[2]).Abs(); double angle=Math.toDegrees(Math.atan2(d1,d2)); double angle1=Math.atan2(array[3].GetX(1)-array[1].GetX(1),array[3].GetX(0)-array[1].GetX(0)); double angle2=Math.atan2(array[2].GetX(1)-array[0].GetX(1),array[2].GetX(0)-array[0].GetX(0)); double deltaangle=Math.toDegrees(angle1-angle2); return (d1\*d2\*Math.sin(deltaangle))/2; \*/ }}

*RECTANGLE*

package com.company;public class Rectangle extends QGon{ public Rectangle(Point2D[] array) { super(array); } public double Square() { return Point.*Sub*(array[0], array[1]).Abs() \* Point.*Sub*(array[0], array[3]).Abs(); }}

*TGON*

package com.company;public class TGon extends NGon{ TGon(Point2D[] array) { super(3, array); } public double Square() { return (array[0].GetX(0) \* (array[1].GetX(1) - array[2].GetX(1)) + array[1].GetX(0) \* (array[2].GetX(1) - array[0].GetX(1)) + array[2].GetX(0) \* (array[0].GetX(1) - array[1].GetX(1))) / 2; }}

*TRAPEZE*

package com.company;public class Trapeze extends QGon{ public Trapeze(Point2D[] array) { super(array); } @Override public double Square() { return (Point.*Sub*(array[0], array[1]).Abs() + Point.*Sub*(array[2], array[3]).Abs()) / 2 \* (array[1].GetX(1) - array[0].GetX(1)); }}

*CIRCLE*

package com.company;import java.util.Scanner;import static com.company.Point.\*;import static java.lang.Math.*PI*;public class Circle implements IShape{ private double R; private Point2D X; public Circle(Point2D O, double R) { this.X = O; this.R = R; } public void SetC(double m[]) { X.SetX(m); } public double Square() { return R \* R \* *PI*; } public double Length() { return 2 \* R \* *PI*; } public Point GetC() { return this.X; } public double GetR() { return R; } public void setR(double r) { R = r; } public IShape Shift(Point2D a) { return new Circle((Point2D) *Add*(X, a), this.R); } public IShape Rot(double b) { return new Circle((Point2D) X.Rot(b), this.R); } public IShape Symaxis(int N) { return new Circle((Point2D) X.Simm(N), this.R); } public String toString() { return "{" + X.toString() + ", R = " + R; } public boolean Cross(IShape a) { Circle Cir = (Circle) a; if (Cir.GetR() + this.GetR() >= Point.*Sub*(Cir.GetC(), this.GetC()).Abs()) { return true; } else return false; }}

*P.S. заполненный вариант отправьте мне назад на почту:* [*hometask123@rambler.ru*](mailto:hometask123@rambler.ru)

*Спасибо.*