|  |  |
| --- | --- |
| <abstract>  Polygon | |
| angle1: Double |  |
| angle2: Double |  |
| <abstract>  Polygon(area: Double) |  |

|  |  |
| --- | --- |
| DefinedPoints | |
| angle1: Double |  |
| angle2: Double |  |
| boolean check(angle1: double, angle2) |  |

|  |  |
| --- | --- |
| Rectangle → Polygon | |
| a: Double |  |
| b: Double |  |
| rectangle(area: double) |  |

|  |  |
| --- | --- |
| Triangle → Polygon | |
| a: Double |  |
| b: Double |  |
| Triangle(area: double) |  |

|  |  |
| --- | --- |
| Ellipse --Polygon | |
| a: Double |  |
| b: Double |  |
| Ellipse (area: double) |  |

|  |  |
| --- | --- |
| <interface> DefinedPoints | |
| a: Double |  |
| b: Double |  |
| boolean check(angle1: double, angle2) |  |

package week1;

import java.awt.List;

import java.util.ArrayList;

//been feeling the math ever since the eclipse

abstract class Polygon {

double angle1;

double angle2;

Polygon(double a, double b) {

angle1 = a;

angle2 = b;

}

abstract double area();

//make area abstract without making it, its own abstract class

}

interface DefinedPoints {

double angle1 = 4;

double angle2 = 3;

boolean check( double angle1, double angle2);

}

//Circles are PIr^2 but an ellipse cn have multiple points/poles generally 2 to 4, so you multiply PIxy

class Ellipse extends Polygon {

Ellipse(double a, double b) {

super(a, b);

}

// override

double area() {

System.out.print("Area of an Ellipse: ");

//We're going to act like the poles of the ellipse are angles

return Math.PI\*angle1 \* angle2;

}

}

//perfect squares would imply a = b

class Rectangle extends Polygon {

Rectangle(double a, double b) {

super(a, b);

}

// override area

double area() {

System.out.print("Area of a Rectangle: ");

return angle1 \* angle2;

}

}

class Triangle extends Polygon {

Triangle(double a, double b) {

super(a, b);

}

// override area

double area() {

System.out.print("Area of a Triangle: ");

return angle1 \* angle2 / 2;

}

}

//would also make this public if it were in its own file

//Special case of a triangle requires previously defined points

class IsoscelesTriangle implements DefinedPoints{

public boolean check(double angle1, double angle2) {

if (angle1 == angle2) return true;

else return false;

}

}

class AbstractAreas {

public static void main(String args[]) {

ArrayList<Polygon> PolygonList = new ArrayList<Polygon>();

PolygonList.add(new Rectangle(9, 5));

PolygonList.add(new Ellipse(8, 9));

PolygonList.add(new Triangle(7, 8));

for (Polygon f : PolygonList) {

System.out.println(f.area());

}

IsoscelesTriangle IT = new IsoscelesTriangle();

System.out.println(IT.check(3, 4));

System.out.print(IT.check(4, 4));

}

}

