abstract class Shape {

def perimeter: Double

def area: Double

}

class Circle(val radius: Double) extends Shape {

def diameter = 2 \* radius

def perimeter = math.Pi \* diameter

def area = math.Pi \* radius \* radius

}

trait Triangle extends Shape {

def a: Double

def b: Double

def c: Double

def ab: Double

def ac: Double

def bc: Double

}

final class TwoSidesAngleTriangle (val ab: Double,

val ac: Double,

val a: Double)

extends Shape

with Triangle {

def bc = math.sqrt(

(ab \* ab) + (ac \* ac) - 2 \* ab \* ac \* math.cos(a))

def b = math.atan(ac / ab)

def c = math.atan(ab / ac)

def perimeter = ab + ac + bc

def area = ab \* ac \* math.sin(a) / 2

}

final class TwoAnglesSideTriangle (val a: Double,

val b: Double,

val ab: Double)

extends Shape

with Triangle {

def c = 180-a-b

def bc = ab\*(math.sin(a)/math.sin(c))

def ac = ab\*(math.sin(b)/math.sin(c))

def perimeter = ab + ac + bc

def area = ab \* ac \* math.sin(a) / 2

}

final class ThreeSidesTriangle (val ab: Double,

val ac: Double,

val bc: Double)

extends Shape

with Triangle {

def a=math.acos((ab \* ab) + (ac \* ac) - (bc \* bc))/(2\*ab\*ac)

def b=math.acos((ab \* ab) + (bc \* bc) - (ac \* ac))/(2\*ab\*bc)

def c=180-(a+b)

def perimeter = ab + ac + bc

def area = ab \* ac \* math.sin(a) / 2

}

abstract class Quadrilateral extends Shape {

def cd: Double

def bd: Double

def ab: Double

def ac: Double

def bc: Double

def ad: Double

def h: Double

}

abstract class Trapezium extends Quadrilateral {

}

class Parallelogram extends Quadrilateral {

def cd = ab

def bd = ac

def area = ab \* h

def perimeter = (ab + ac) \* 2

}

class Rhombus (val ad: Double,

val bc: Double)

extends Parallelogram {

def ab = math.sqrt(ad \* ad + bc \* bc)

override area = ad \* bc / 2

def h = area / ab

def ac = ab

override def perimeter = ab \* 4

}

class Rectangle(val ab: Double,

val ac: Double)

extends Parallelogram {

def bc = math.sqrt(ab \* ab + ac \* ac)

def ad = bc

override def perimeter = 2 \* (ab + ac)

override def area = ab \* ac

def h = area / ab

}

class Square(val side: Double)

extends Rectangle(side, side) {

override def perimeter = 4 \* side

override def bc = math.sqrt(2) \* side

}

val r1 = new Rectangle(3, 4)

// Периметр

r1.perimeter

// Площадь

r1.area

// Диагональ

r1.bc

val r2 = new TwoSidesAngleTriangle(10, 12, 30)

// Периметр

r2.perimeter

// Площадь

r2.area

r2.a

r2.b

r2.c

r2.ab

r2.ac

r2.bc

val r3 = new TwoAnglesSideTriangle(60, 30, 10)

// Периметр

r3.perimeter

// Площадь

r3.area

r3.a

r3.b

r3.c

r3.ab

r3.ac

r3.bc

val r4 = new TwoSidesAngleTriangle(10, 12, 15)

// Периметр

r4.perimeter

// Площадь

r4.area

r4.a

r4.b

r4.c

r4.ab

r4.ac

r4.bc

val s = new Square(5)

s.perimeter

s.area

s.ab

s.ac

s.bd

s.cd

s.bc

val s1 = new Rhombus(5, 7)

s1.perimeter

s1.area

s1.ab

s1.ac

s1.bd

s1.cd

s1.bc