# Лабораторная работа 4, задание t-6 (вариант 3)

## Условие

Написать программу для выполнения действий поворота и сдвига геометрических фигур.

## Код

{-

ПИ19-3 Деменчук Георгий t-6

Написать программу для выполнения

действий поворота и сдвига геометрических фигур

-}

-- Вводим кастомный тип фигур

data BaseFigure = Circle Float Float Float

| Rectangle Float Float Float Float

| Triangle Float Float Float Float Float Float deriving (Eq, Show, Read)

-- Получение площади фигуры

getFigureArea :: BaseFigure -> Float

getFigureArea (Circle \_ \_ r) = pi \* r \* r

getFigureArea (Rectangle x1 y1 x2 y2) = abs(x2-x1)\*abs(y2-y1)

getFigureArea (Triangle x1 y1 x2 y2 x3 y3) = (1/2)\*abs((x2-x1)\*(y3-y1) - (x3-x1)\*(y2-y1))

-- Сдвиг фигуры на заданную величину по x и y

setFigureMovement :: BaseFigure -> Float -> Float -> BaseFigure

setFigureMovement (Circle x y r) dx dy = (Circle (x+dx) (y+dy) r)

setFigureMovement (Triangle x1 y1 x2 y2 x3 y3) dx dy = (Triangle (x1+dx) (y1+dy) (x2+dx) (y2+dy) (x3+dx) (y3+dy))

setFigureMovement (Rectangle x1 y1 x2 y2) dx dy = (Rectangle (x1+dx) (y1+dy) (x2+dx) (y2+dy))

-- Поиск новых координат точки для поворота

xnew :: Float -> Float -> Float -> Float -> Float -> Float

xnew x y x0 y0 alpha = (x-x0)\*cos(alpha) - (y-y0)\*sin(alpha) + x0

ynew :: Float -> Float -> Float -> Float -> Float -> Float

ynew x y x0 y0 alpha = (x-x0)\*sin(alpha) + (y-y0)\*cos(alpha) + y0

-- Поиск центра фигуры

findRectangleCentre :: Float -> Float -> Float

findRectangleCentre x1 x2 = (x2-x1)/2 + x1

findTriangleCentre :: Float -> Float -> Float -> Float

findTriangleCentre x1 x2 x3 = (x1+x2+x3)/3

-- Поворот фигуры на заданный угол alpha

setFigureRotate :: BaseFigure -> Float -> BaseFigure

setFigureRotate (Rectangle x1 y1 x2 y2) alpha = (Rectangle x1new y1new x2new y2new)

where x1new = xnew x1 y1 (findRectangleCentre x1 x2) (findRectangleCentre y1 y2) alpha

y1new = ynew x1 y1 (findRectangleCentre x1 x2) (findRectangleCentre y1 y2) alpha

x2new = xnew x2 y2 (findRectangleCentre x1 x2) (findRectangleCentre y1 y2) alpha

y2new = ynew x2 y2 (findRectangleCentre x1 x2) (findRectangleCentre y1 y2) alpha

setFigureRotate (Circle x y r) alpha = (Circle x y r)

setFigureRotate (Triangle x1 y1 x2 y2 x3 y3) alpha = (Triangle x1new y1new x2new y2new x3new y3new)

where x1new = xnew x1 y1 (findTriangleCentre x1 x2 x3) (findTriangleCentre y1 y2 y3) alpha

y1new = ynew x1 y1 (findTriangleCentre x1 x2 x3) (findTriangleCentre y1 y2 y3) alpha

x2new = xnew x2 y2 (findTriangleCentre x1 x2 x3) (findTriangleCentre y1 y2 y3) alpha

y2new = ynew x2 y2 (findTriangleCentre x1 x2 x3) (findTriangleCentre y1 y2 y3) alpha

x3new = xnew x3 y3 (findTriangleCentre x1 x2 x3) (findTriangleCentre y1 y2 y3) alpha

y3new = ynew x3 y3 (findTriangleCentre x1 x2 x3) (findTriangleCentre y1 y2 y3) alpha

--Main-метод

main :: IO ()

main = do

-- Задаем переменные для тестовых данных

-- Тестовый круг

let testCircle = Circle 4 6 10

print("testCircle ",testCircle)

print("---")

-- Тестовый треугольник

let testTriangle = Triangle 1 1 6 4 8 2

print("testTriangle ", testTriangle)

print("---")

-- Тестовый прямоугольник

let testRectangle = Rectangle 2 2 5 5

print("testRectangle ", testRectangle)

print("---")

-- Тестирование перемещения фигуры

print("--- setFigureMovement 2 2 ---")

print(setFigureMovement testCircle 2 2)

print(setFigureMovement testTriangle 2 2)

print(setFigureMovement testRectangle 2 2)

-- Тестирование поворота фигуры

print("---- setFigureRotate 30 ----")

print(setFigureRotate testCircle 30)

print(setFigureRotate testTriangle 30)

print(setFigureRotate testRectangle 30)

-- Тестирование получения площади фигур

print("---- getFigureArea ----")

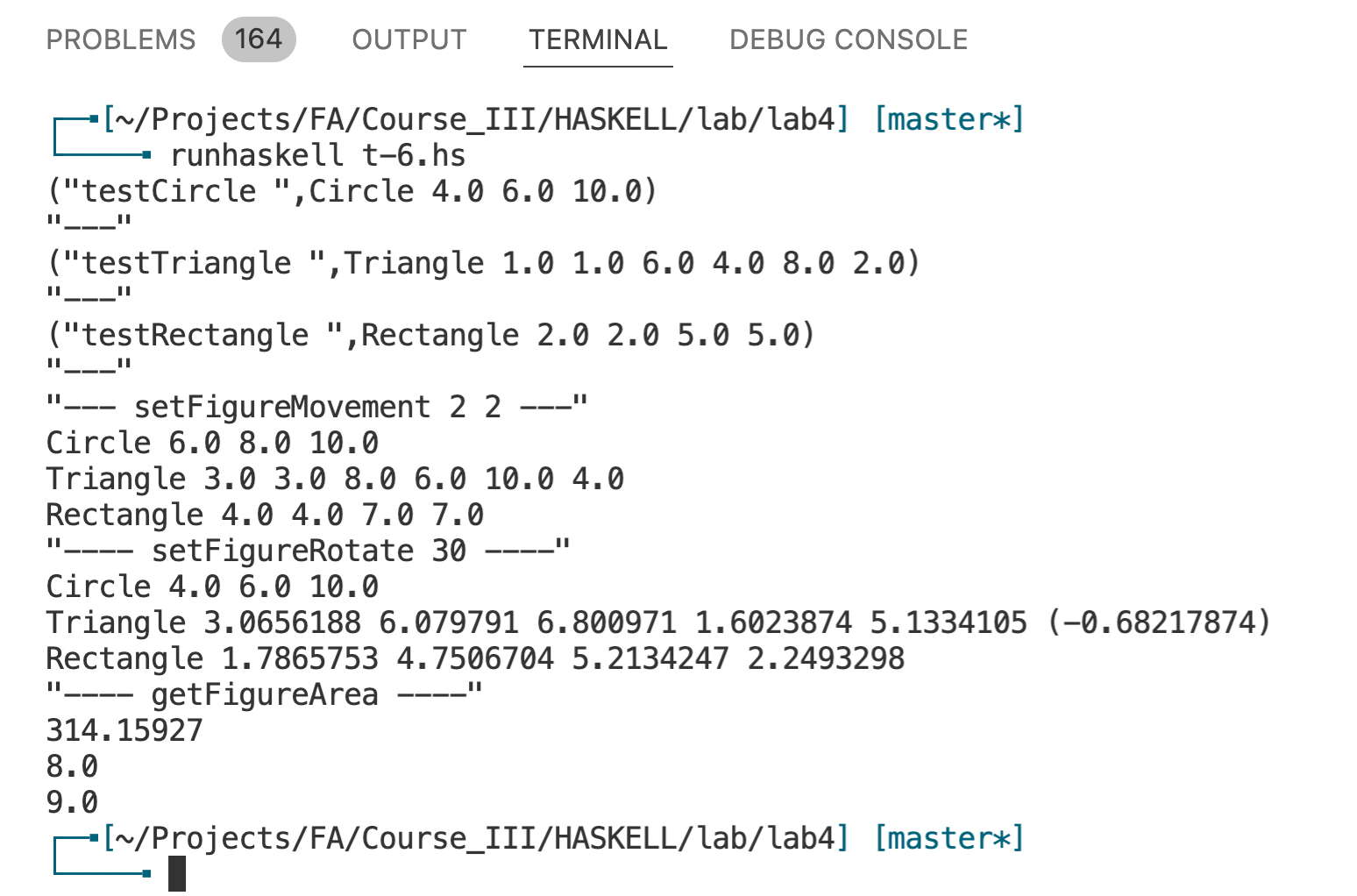
print(getFigureArea testCircle)

print(getFigureArea testTriangle)

print(getFigureArea testRectangle)

## Скриншоты

Пример отработки точки входа программы со всеми вызываемыми функциями



Сначала выводятся тестовые фигуры, которые мы используем для тестирования написанных функций, далее вызываются сами функции:

* setFigureMovement с аргументами 2 2
* setFigureRotate с аргументом 30
* getFigureArea без аргументов