Задача 2. Интерфейс Shape (дополнительное задание)

У вас есть набор классов различных геометрических фигур: круг(Circle), прямоугольник(Rectangle) и треугольник(Triangle), которые наследуются от общего класса геометрическая фигура (Shape). Каждый из этих классов имеет методы подсчёта площади, описывающего прямоугольника и взятия типа. Также есть общая функция printParams которая по переданному ей указателю на объект типа геометрическая фигура выводит на экран тип фигуры.

Что нужно сделать

Создайте интерфейс Shape (фигуру), который предоставляет следующие методы:

— virtual double ***square***() = 0; — возвращает площадь конкретной фигуры;

— virtual BoundingBoxDimensions ***dimensions***() = 0; — размеры описывающего прямоугольника;

BoundingBoxDimensions — простая структура, которая содержит ширину и высоту;

—virtual std::string ***type***() = 0; — название конкретного типа, например, вернуть строку Triangle для класса Triangle;

void **printParams**(Shape \***shape**) — сюда можно будет передавать любые фигуры:

{   
   std::cout <<"Type: " << shape->type() << std::endl;

комбинация virtual … = 0; — создаёт чисто виртуальный метод. Это означает, что такой метод обязательно должен быть переопредёлен в классе-наследнике. Если программист пронаследует свой класс от такого интерфейса, то компилятор будет ругаться на отсутствие методов, описанных как virtual … = 0. То есть программист будет вынужден добавить такие методы в свой класс, чтобы создать объект такого типа.

   ….

Создайте классы-наследники Shape: Circle, Rectangle и Triangle.

У каждого из классов будет разный набор полей: у треугольника будут три стороны в виде переменных double, у круга — радиус, а у прямоугольника — длина и ширина. Так как все они наследуются от Shape, в каждом из классов следует реализовать методы нахождения размерности, площади и метод, возвращающий название типа.

Площадь треугольника при известных длинах сторон находится при помощи формулы Герона:

sqrt(p \* (p - a) \* (p - b) \* (p - c)), где p — полупериметр.

Размеры описывающего прямоугольника для треугольника находятся по похожей формуле — сначала находим радиус описанной окружности:

a \* b \* c / (4 \* sqrt(p\*(p - a)\*(p - b)\*(p - c))), где р — снова полупериметр.

Затем умножаем его на два и получаем сторону квадрата, в который помещён треугольник.

В функции main вызовите printParams со всеми типами, которые вы создали.

**Пример**

Triangle t(3,4,5);   
    printParams(&t);

**Вывод**

Type: Triangle   
Square: 6   
Width: 5   
Height: 5

Что оценивается

* Приложение выводит в консоль сообщения о типе фигуры, которые соответствуют действительности.
* Функция принимает корректные типы данных. Тип возвращаемого значения — void.
* Решение выполнено через создание наследников от интерфейса Shape.
* Сигнатура функции void — printParams (Shape \*shape). Любая другая некорректна.