Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Информационных технологий и программирования

Программирование на C++

Лабораторная работа №4, Виртуальные функции.

Вариант 5

Выполнил: Гаджиев Саид M3115

Санкт-Петербург

2023 г.

Лабораторная работа №4. “Виртуальные функции”.
Реализовать все указанные интерфейсы (абстрактные базовые классы) для классов (согласно варианту):
A. Круг
B. Отрезок
C. Равносторонний треугольник
D. Прямоугольник
E. Шестиугольник
F. Параллелограмм
G. Равнобедренная трапеция
H. Эллипс
Функционал системы:
-Хранение множества фигур
-Динамическое добавление фигур пользователем. (через консоль)
-Отобразить все фигуры.
-Суммарная площадь всех фигур.
-Суммарный периметр всех фигур.
-Центр масс всей системы.
-Память, занимаемая всеми экземплярами классов.
-Сортировка фигур между собой по массе.
Вопросы для обдумывания:
-Есть ли необходимость делать методы сравнения по массе виртуальными?
-Получится ли также перегрузить операторы сравнения для интерфейса BaseCObject чтобы сравнивать объекты по объему занимаемой памяти?
-Предположите, что в дальнейшем придется изменить код таким образом, чтобы фигуры (оставаясь сами по себе плоскими) задавались уже не в двумерном, а в трехмерном пространстве. Укажите как бы вы действовали? Что пришлось бы изменить?

Вариант 5

Решение:

main.cpp

```
#include <iostream>
#include "figures.h"

int main() {
    Figure *fig1 = new Hexagon(5, 3.5, 0, 0);
    Figure *fig2 = new IsoscelesTrapezoid(8, 12, 5, 5, 2, 2);

    std::cout << "Figure 1:\n";
    fig1->print();
    std::cout << "Perimeter: " << fig1->getPerimeter() << "\n";
    std::cout << "Area: " << fig1->getArea() << "\n";
    std::cout << "Mass: " << fig1->getMass() << "\n";
    std::cout << "Coordinates: (" << fig1->getX() << ", " << fig1->getY() <<
")" << "\n";

    std::cout << "\nFigure 2:\n";
    fig2->print();
    std::cout << "Perimeter: " << fig2->getPerimeter() << "\n";
    std::cout << "Area: " << fig2->getArea() << "\n";
    std::cout << "Mass: " << fig2->getMass() << "\n";
    std::cout << "Coordinates: (" << fig2->getX() << ", " << fig2->getY() <<
")" << "\n";
    return 0;
}
```

Здесь написана функция main(), которая использует классы Figure, Hexagon и IsoscelesTrapezoid, описанные в заголовочном файле figures.h. Внутри main() создаются два указателя на объекты классов Hexagon и IsoscelesTrapezoid, соответственно, которые инициализируются с помощью конструкторов этих классов. Далее для каждого объекта выводятся на экран свойства фигуры - периметр, площадь, масса и координаты центра. Для этого вызываются соответствующие методы класса Figure - getPerimeter(), getArea(), getMass(), getX() и getY(), а также метод print() для вывода на экран информации о каждой фигуре. В конце функция main() возвращает 0.

figures.h

```
//
// Created by Redmibook on 16.03.2023.
//
#ifndef LAB_4_FIGURES_H
#define LAB_4_FIGURES_H

// Базовый абстрактный класс для всех фигур
class Figure {
    public:
        virtual double getPerimeter() = 0;
        virtual double getArea() = 0;
        virtual double getMass() = 0;
        virtual double getX() = 0;
        virtual double getY() = 0;
        virtual void print() = 0;
};

// Класс для Шестиугольника
class Hexagon : public Figure {
    private:
        double a;
        double mass;
        double x;
        double y;

    public:
        Hexagon(double side, double m, double x_coord, double y_coord);

        double getPerimeter() override;
        double getArea() override;
        double getMass() override;
        double getX() override;
        double getY() override;
        void print() override;
};

// Класс для Равнобедренной трапеции
class IsoscelesTrapezoid : public Figure {
    private:
        double base1;
        double base2;
        double height;
        double mass;
        double x;
        double y;

    public:
        IsoscelesTrapezoid(double b1, double b2, double h, double m, double
x_coord, double y_coord);

        double getPerimeter() override;
        double getArea() override;
```

```
        double getMass() override;
        double getX() override;
        double getY() override;
        void print() override;
};

#endif //LAB_4_FIGURES_H
```

Здесь определены два класса: Figure, абстрактный базовый класс для всех фигур, и его два наследника - Hexagon и IsoscelesTrapezoid, классы, представляющие соответствующие геометрические фигуры.

Класс Figure содержит только чисто виртуальные функции (функции без определения), которые должны быть реализованы в производных классах. Эти функции возвращают периметр, площадь, массу, координаты и печатают информацию о фигуре.

Класс Hexagon описывает шестиугольник и содержит данные о стороне, массе и координатах центра фигуры. Он реализует функции, унаследованные от Figure, для вычисления периметра, площади, массы, координат и печати информации о фигуре.

Класс IsoscelesTrapezoid описывает равнобедренную трапецию и содержит данные о двух основаниях, высоте, массе и координатах центра фигуры. Он также реализует функции, унаследованные от Figure, для вычисления периметра, площади, массы, координат и печати информации о фигуре.

figures.cpp

```
//
// Created by Redmibook on 16.03.2023.
//
#include <iostream>
#include <cmath>
#include "figures.h"

// Методы класса Hexagon
Hexagon::Hexagon(double side, double m, double x_coord, double y_coord) {
    a = side;
    mass = m;
    x = x_coord;
    y = y_coord;
}

double Hexagon::getPerimeter() {
    return 6 * a;
}

double Hexagon::getArea() {
    return 3 * sqrt(3) * a * a / 2;
}

double Hexagon::getMass() {
    return mass;
}

double Hexagon::getX() {
    return x;
}
```

```
double Hexagon::getY() {
    return y;
}

void Hexagon::print() {
    std::cout << "Hexagon with side " << a << ", mass " << mass << ", and
center (" << x << ", " << y << ")" << "\n";
}



// Методы класса IsoscelesTrapezoid
IsoscelesTrapezoid::IsoscelesTrapezoid(double b1, double b2, double h, double
m, double x_coord, double y_coord) {
    base1 = b1;
    base2 = b2;
    height = h;
    mass = m;
    x = x_coord;
    y = y_coord;
}

double IsoscelesTrapezoid::getPerimeter() {
    return base1 + base2 + 2 * sqrt(pow((base2 - base1) / 2, 2) + pow(height,
2));
}

double IsoscelesTrapezoid::getArea() {
    return (base1 + base2) * height / 2;
}

double IsoscelesTrapezoid::getMass() {
    return mass;
}

double IsoscelesTrapezoid::getX() {
    return x;
}

double IsoscelesTrapezoid::getY() {
    return y;
}

void IsoscelesTrapezoid::print() {
    std::cout << "Isosceles Trapezoid with bases " << base1 << " and " <<
base2 << ", height " << height << ", mass " << mass << ", and center (" << x
<< ", " << y << ")" << "\n";
}
```

Здесь определяются методы классов Hexagon и IsoscelesTrapezoid, наследуемых от базового абстрактного класса Figure. Класс Hexagon имеет конструктор, принимающий длину стороны шестиугольника, массу, а также координаты центра. Затем определяются методы для вычисления периметра, площади, массы, координаты x и y центра, а также метод вывода информации о фигуре. Аналогичным образом определяются методы класса IsoscelesTrapezoid, который имеет конструктор, принимающий длины верхнего и нижнего оснований, высоту, массу и координаты центра.