МИНОБРНАУКИ РФ

ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе № 2

по объектно-ориентированному программированию

Вариант №1

Выполнил студент группы Б04-191-3: Р.А. Гумметов

Принял: К.С. Чернышев

Ижевск 2020

Постановка задачи

Используя принципы ООП реализовать программу для вычисления площади следующих фигур:

Эллипс,

Прямоугольник,

Треугольник.

В программе необходимо иметь хотя бы один абстрактный класс.

При запуске, программа должна выполнять следующие действия:

* Создать 2 случайных фигуры.
* Задать фигурам случайные размеры.
* Вывести информацию о фигурах.
* Рассчитать площади фигур.

Сравнить площади фигур, вывести названия большей фигуры.

Иерархия классов

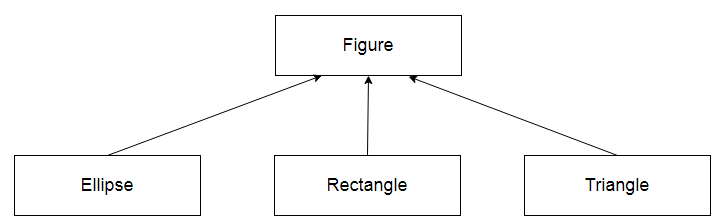


Рис. 1

Текст программы

#include <iostream>

#include <string>

#include <math.h>

#include <cstdlib>

#include <ctime>

using namespace std;

class Figure

{

public:

virtual double FindSquare() = 0;

virtual void OnScreen() = 0;

};

class Ellipse : public Figure

{

public:

double Width;

double Heigth;

Ellipse()

{

Width = rand() % 20 + 1;

Heigth = rand() % 20 + 1;

}

double FindSquare() override

{

return (Width / 2) \* (Heigth / 2) \* acos(-1.0);

}

void OnScreen() override

{

cout << "Эллипс: " << "ширина = " << Width << ", высота = " << Heigth << endl;

cout << "Площадь эллипса: " << FindSquare() << endl << endl;

}

};

class Rectangle : public Figure

{

public:

double Width;

double Heigth;

Rectangle()

{

Width = rand() % 20 + 1;

Heigth = rand() % 20 + 1;

}

double FindSquare() override

{

return Width \* Heigth;

}

void OnScreen() override

{

cout << "Прямоугольник: " << "ширина = " << Width << ", высота = " << Heigth << endl;

cout << "Площадь прямоугольника: " << FindSquare() << endl << endl;

}

};

class Triangle : public Figure

{

private:

double HalfPerimeter;

void SetHalfPerimeter()

{

HalfPerimeter = (A + B + C) / 2.0;

}

public:

double A;

double B;

double C;

double GetHalfPerimeter()

{

return HalfPerimeter;

}

Triangle()

{

A = rand() % 20 + 1;

B = rand() % 20 + 1;

C = rand() % (int)(A + B - abs(A - B)) + abs(A - B) - 1;

SetHalfPerimeter();

}

double FindSquare() override

{

return sqrt(HalfPerimeter \* abs(HalfPerimeter - A) \* abs(HalfPerimeter - B) \* abs(HalfPerimeter - C));

}

void OnScreen() override

{

cout << "Треугольник: " << "сторона A = " << A << ", сторона B = " << B << ", сторона C = " << C << endl;

cout << "Площадь треугольника: " << FindSquare() << endl << endl;

}

};

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

Figure\* firstFigure = nullptr;

Figure\* secondFigure = nullptr;

for (int i = 0; i < 2; i++)

{

int num = rand() % 3 + 1;

switch (num)

{

case 1:

{

if (i == 0)

firstFigure = new Ellipse();

else

secondFigure = new Ellipse();

break;

}

case 2:

{

if (i == 0)

firstFigure = new Rectangle();

else

secondFigure = new Rectangle();

break;

}

case 3:

{

if (i == 0)

firstFigure = new Triangle();

else

secondFigure = new Triangle();

break;

}

default:

break;

}

}

firstFigure->OnScreen();

secondFigure->OnScreen();

if (firstFigure->FindSquare() < secondFigure->FindSquare())

cout << "Фигура 1 меньше фигуры 2" << endl;

else if (firstFigure->FindSquare() > secondFigure->FindSquare())

cout << "Фигура 1 больше фигуры 2" << endl;

else

cout << "Фигуры равны" << endl;

}

Пример работы программы

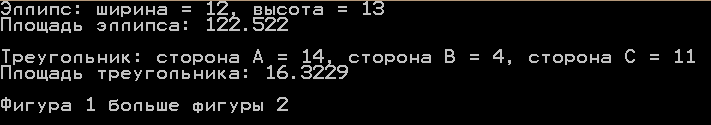


Рис. 2

Вопросы:

1) Что такое наследование?

Наследование позволяет описать новый класс на основе уже существующего. Свойства и функциональность базового класса заимствуются новым классом.

2)  Чем виртуальный метод отличается от чистого виртуального метода?

Виртуальный метод - это метод, который определяется в базовом классе, а любой порожденный класс может его переопределить.

Чистый виртуальный метод - это метод, определяемый в базовом классе, имеющий пустое тело. Определенное значение ему придается лишь в классах-наследниках.

3)  Что такое полиморфизм?

Полиморфизм — возможность объектов с одинаковой спецификацией иметь различную реализацию.

4) Что такое абстрактный класс?

Абстрактный класс - это класс, который не предполагает создание экземпляров. Такой класс предназначен только для создание от него производных классов.

5) Возможности указателей базовых и абстрактных классов?

Указатель на абстрактный класс может указывать на производные классы, таким образом в один массив собрать разные производные классы.