Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

ОТЧЕТ

по лабораторной работе №2

ПРОСТОЕ НАСЛЕДОВАНИЕ. ВИРТУАЛЬНЫЕ ФУНКЦИИ.

Выполнил: ст. гр. ИТб-21

Куркчи А.Э.

Проверила:

Тимофеев И.С.

Севастополь

2015

# 1. Цель

Приобретение практических навыков при написании объектно-ориентированных программ с использованием механизма наследования и механизма виртуальных функций. Освоение особенностей отладки объектно-ориентированных программ.

2. Постановка задач

Для заданной по варианту иерархии описать классы, конструкторы и деструктор, функции ввода и вывода информации на экран. Базовый класс определить, как абстрактный, а заданную функцию – как чисто виртуальную в базовом классе и переопределить её в остальных классах иерархии. Проиллюстрировать корректную работу виртуальных функций и механизма наследования.

Вариант №1

Создать абстрактный базовый класс Фигура с виртуальной функцией Площадь. Создать производные классы: прямоугольник, круг, прямоугольный треугольник со своими функциями площади.

3. Текст программы

#include <iostream>

#include <stdio.h>

#include <string.h>

#include <math.h>

**using** **namespace** std**;**

class figure **{**

public**:**

virtual void print**()** **=** 0**;**

virtual void scan**()** **=** 0**;**

virtual double area**()** **=** 0**;**

**};**

class rectangle**:** public figure **{**

private**:**

double width**;**

double height**;**

public**:**

rectangle**()** **{}**

rectangle**(**double w**,** double h**)** **{**

width **=** w**;**

height **=** h**;**

**}**

**~**rectangle**()** **{**

width **=** 0**;**

height **=** 0**;**

**}**

void print**()** **{**

cout **<<** "Rectangle width = " **<<** width **<<** "; height = " **<<** height **<<** "; area = " **<<** area**()** **<<** endl**;**

**}**

void scan**()** **{**

cout **<<** "Input rectangle width:" **<<** endl**;**

cin **>>** width**;**

cout **<<** "Input rectangle height:" **<<** endl**;**

cin **>>** height**;**

**}**

double area**()** **{**

**return** width**\***height**;**

**}**

**};**

class circle**:** public figure **{**

private**:**

double radius**;**

public**:**

circle**()** **{}**

circle**(**double r**)** **{**

radius **=** r**;**

**}**

**~**circle**()** **{**

radius **=** 0**;**

**}**

void print**()** **{**

cout **<<** "Circle radius = " **<<** radius **<<** "; area = " **<<** area**()** **<<** endl**;**

**}**

void scan**()** **{**

cout **<<** "Input circle radius:" **<<** endl**;**

cin **>>** radius**;**

**}**

double area**()** **{**

**return** M\_PI**\***pow**(**radius**,**2**);**

**}**

**};**

class rtriangle**:** public figure **{**

private**:**

double base**;**

double side**;**

public**:**

rtriangle**()** **{}**

rtriangle**(**double b**,** double s**)** **{**

base **=** b**;**

side **=** s**;**

**}**

**~**rtriangle**()** **{**

base **=** 0**;**

side **=** 0**;**

**}**

void print**()** **{**

cout **<<** "Right triangle base = " **<<** base **<<** "; side = " **<<** side **<<** "; area = " **<<** area**()** **<<** endl**;**

**}**

void scan**()** **{**

cout **<<** "Input right triangle base:" **<<** endl**;**

cin **>>** base**;**

cout **<<** "Input right triangle side:" **<<** endl**;**

cin **>>** side**;**

**}**

double area**()** **{**

**return** base**/**2.0 **\*** sqrt**(**pow**(**side**,**2**)** **-** base**/**2.0**);**

**}**

**};**

int main**()** **{**

cout **<<** "SevGU semestr.03 OOP lab.02" **<<** endl**;**

cout **<<** "Demonstration start" **<<** endl**;**

figure**\*** f**;**

// cout << "Creating rectangle and assigning it to f" << endl;

cout **<<** "> f = new rectangle();" **<<** endl**;**

f **=** **new** rectangle**();**

// cout << "Scaning rectangle" << endl;

cout **<<** "> f->scan();" **<<** endl**;**

f**->**scan**();**

// cout << "Printing rectangle" << endl;

cout **<<** "> f->print();" **<<** endl**;**

f**->**print**();**

// cout << "Creating circle and assigning it to f" << endl;

cout **<<** "> f = new circle();" **<<** endl**;**

f **=** **new** circle**();**

// cout << "Scaning circle" << endl;

cout **<<** "> f->scan();" **<<** endl**;**

f**->**scan**();**

// cout << "Printing circle" << endl;

cout **<<** "> f->print();" **<<** endl**;**

f**->**print**();**

// cout << "Creating rtriangle and assigning it to f" << endl;

cout **<<** "> f = new rtriangle();" **<<** endl**;**

f **=** **new** rtriangle**();**

// cout << "Scaning rtriangle" << endl;

cout **<<** "> f->scan();" **<<** endl**;**

f**->**scan**();**

// cout << "Printing circle" << endl;

cout **<<** "> f->print();" **<<** endl**;**

f**->**print**();**

cout **<<** "Demonstration end" **<<** endl**;**

**return** 0**;**

**}**

4. Тестовые примеры

На рисунке 1 представлен вывод программы

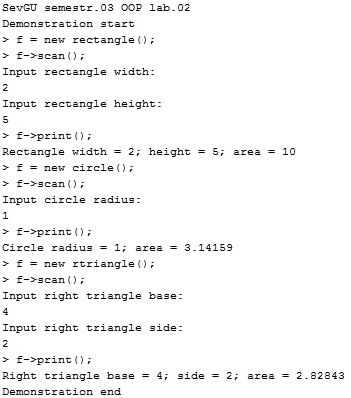


Рисунок 1 – Вывод программы

После ввода 3х событий начинается демонстрация корректности программы. В строках, начинающихся с «>», печатается выполняемая в программном коде строка, в остальных строках – вывод программы. Метод scan отвечает за чтение параметров фигуры из консоли, а print за параметров фигуры и её площади.

Вывод

В ходе лабораторной работы были приобретены практические навыки написания объектно-ориентированных программ с использованием механизма наследования и механизма виртуальных функций. Освоены особенности отладки объектно-ориентированных программ.