

ГУАП

КАФЕДРА № 14

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доцент

должность, уч. степень, звание

подпись, дата

К.А. Курицын

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

ЛАБОРАТОРНАЯ РАБОТА № 3

по курсу: ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

1041

подпись, дата

Ю.В. Ахромова

инициалы, фамилия

Санкт-Петербург 2022

1. Постановка задачи

Создать абстрактный базовый класс с виртуальной функцией «Площадь». Создать производные классы «Прямоугольник», «Круг», «Прямоугольный треугольник», «Трапеция» со своими функциями площади и переменными. Для проверки определить массив ссылок на абстрактный класс, которым присваиваются адреса различных объектов.

2. Формализация задачи

Создать класс “Calculate”. Объявление класса Calculate находится в файле Calculate.h.

Класс имеет две публичные функции:

getSquare - площадь.

~Calculate - деструктор.

Параметры следующих классов:

Класс Circle. Объявление класса находится в файле Circle.h.

Поля с модификатором private:

double r; - радиус

double pi; - число π

Поля с модификатором public:

Circle(); - обычный конструктор

Circle(double r); - конструктор с параметром

double getSquare() override; - площадь фигуры

Класс Rectangle. Объявление класса находится в файле Rectangle.h.

Поля с модификатором private:

double length; - длина

double width; - ширина

Поля с модификатором public:

Rectangle(); - обычный конструктор

Rectangle(double a, double b); - конструктор с параметром

double getSquare() override; - площадь фигуры

Класс Trapezoid. Объявление класса находится в файле Trapezoid.h.

Поля с модификатором private:

double side1; - 1 основание

double side2; - 2 основание

double height; - высота

Поля с модификатором public:

Trapezoid(); - обычный конструктор

Trapezoid(double a, double b, double h); - конструктор с параметром

double getSquare() override; - площадь фигуры

Класс Triangle. Объявление класса находится в файле Triangle.h.

Поля с модификатором private:

double side1; - 1 катет

double side2; - 2 катет

Поля с модификатором public:

Triangle(); - обычный конструктор

Triangle(double a, double b); - конструктор с параметром

`double getSquare()override`;- площадь фигуры

Класс `Save_file` сохраняет данные в файл. Объявление класса находится в файле `Save_file.h`.

Поля с модификатором `public`:

`Save_file()`;- обычный конструктор

`Save_file(double a, double b, double f, const char name[12])`;- конструктор с 4 параметрами

`Save_file(double a, double b, double c, double f, const char name[12])`;- конструктор с 5 параметрами.

Основная функция `main()` определена в файле `ConsoleApplication1.cpp`. Программа считывает введённые значения с клавиатуры и с помощью цикла оператора `switch (case)` выбирает какую функцию выполнять:

1) Площадь прямоугольника - Выполняется класс `Rectangle` и предлагается сохранить файл. Сохранение файла описана в классе `Save_file`.

2) Площадь круга- Выполняется класс `Circle` и предлагается сохранить файл. Сохранение файла описана в классе `Save_file`.

3) Площадь прямоугольного треугольника - Выполняется класс `Triangle` и предлагается сохранить файл. Сохранение файла описана в классе `Save_file`.

4) Площадь трапеции - Выполняется класс `Trapezoid` и предлагается сохранить файл. Сохранение файла описана в классе `Save_file`.

5) Чтение файла. Выбирается какой файл открыть для чтения и выводится содержимое на экран.

0) выход - Завершение программы.

3. Листинг программы

3.1. Листинг файла `Calculate.cpp`

```
#include "Calculate.h"
```

```
double Calculate::getSquare()
{
    return 0.0;
}
```

```
Calculate::~Calculate()
{
    cout << "destructor" << std::endl;
}
```

3.2. Листинг файла `Circle.cpp`

```
#include "Circle.h"
```

```
Circle::Circle()
{
    r = 0;
    pi = 3.14;
}
```

```

Circle::Circle(double r1)
{
    r = r1;
    pi = 3.14;
}

double Circle::getSquare()
{
    return pi*r;
}

```

3.3. Листинг файла ConsoleApplication1.cpp

```

#include "Circle.h"
#include "Rectangle.h"
#include "Trapezoid.h"
#include "Triangle.h"
#include "Save_file.h"

void read_file(int type_file, const char name[12]) //чтение файла. в type_file указать
количество переменных 3 или 4
{
    double a = 0, b = 0, c = 0, d = 0;
    ifstream in(name); // открываем файл для чтения
    if (in.is_open())
    {
        if (type_file == 3) //3 значения в файле
        {
            in >> a;
            in >> b;
            in >> c;
            cout << "первое значение = " << a << "\n";
            cout << "второе значение = " << b << "\n";
            cout << "площадь = " << c << "\n";
        }
        else
        {
            in >> a;
            in >> b;
            in >> c;
            in >> d;
            cout << "первое значение = " << a << "\n";
            cout << "второе значение = " << b << "\n";
            cout << "третье значение = " << c << "\n";
            cout << "площадь = " << d << "\n";
        }
    }
    else

```

```

        cout << "Файла нет";

in.close();

}

bool is_digit(const char* S) {           //проверка на число
    const char* temp = "0123456789+";
    unsigned point_count = 0;

    for (int i = 0; i < strlen(S); i++) {
        if ((i > 0) && (S[i] == '+' || S[i] == '-')) return false;
        if (S[i] == '.') {
            point_count++;
            if (point_count > 1) return false;
        }
        if (!strchr(temp, S[i])) return false;
    }
    return true;
}

int return_int()
{
    const int N = 256;
    char S[N];

    cin >> S;
    while (!is_digit(S)) {           //сделать проверку на ввод числа
        cout << "введите число = ";
        cin >> S;
    }
    cout << "ok\n";
    return atof(S);
}

```

```

int main()
{
    setlocale(LC_ALL, "Russian");
    system("cls");
    int n = 1, n2 = 0;
    double res = 0;
    int a = 0, b = 0, c = 0;
    Trapezoid* trapezoid;
    Triangle* triangle;
    Circle* circle;
    Rectangle* rectangle;
    Save_file* savef;

```

```

        cout << "Посчитать площадь для:\n1)Прямоугольник 2)Круг 3)Прямоугольный
треугольник 4)Трапеция 5) открыть файл 0)выход";

```

```

while (n != 0) {
    cout << "\nномер операции = ";
    n = return_int();
    switch (n) {
    case 1:
        cout << "введите длину ";
        a = return_int();
        cout << "введите ширину ";
        b = return_int();

        rectangle = new Rectangle(a, b);
        res = rectangle->getSquare();

        cout << "площадь = " << res << endl;
        cout << "сохранить результат? 1-да, 2-нет";
        while (1) {
            cout << "\nсохранение: номер операции = ";
            n2 = return_int();
            if (n2 == 1)
            {
                savef = new Save_file(a, b, res, "res_rec.txt");
                n2 = 0;
                break;
            }

            if (n2 == 2)
            {
                n2 = 0;
                break;
            }
        }
        break;

```

case 2:

```

    cout << "введите радиус ";
    a = return_int();

    circle = new Circle(a);
    res = circle->getSquare();

    cout << "площадь = " << res << endl;
    cout << "сохранить результат? 1-да, 2-нет";
    while (1) {
        cout << "\nсохранение: номер операции = ";
        n2 = return_int();
        if (n2 == 1)

```

```

    {
        savef = new Save_file(a, 3.14, res, "res_cir.txt");
        n2 = 0;
        break;
    }

    if (n2 == 2)
    {
        n2 = 0;
        break;
    }
}
break;

```

case 3:

```

    cout << "введите 1 катет ";
    a = return_int();
    cout << "введите 2 катет ";
    b = return_int();

    triangle = new Triangle(a,b);
    res = triangle->getSquare();

    cout << "площадь = " << res << endl;
    cout << "сохранить результат? 1-да, 2-нет";
    while (1) {
        cout << "\nсохранение: номер операции = ";
        n2 = return_int();
        if (n2 == 1)
        {
            savef = new Save_file(a, b, res, "res_tri.txt");
            n2 = 0;
            break;
        }

        if (n2 == 2)
        {
            n2 = 0;
            break;
        }
    }
    break;

```

case 4:

```

    cout << "введите 1 основание ";
    a = return_int();
    cout << "введите 2 основание ";
    b = return_int();

```

```

cout << "ВВЕДИТЕ ВЫСОТУ ";
c = return_int();

trapezoid = new Trapezoid(a, b,c);
res = trapezoid->getSquare();

cout << "площадь = " << res << endl;
cout << "сохранить результат? 1-да, 2-нет";
while (1) {
    cout << "\nсохранение: номер операции = ";
    n2 = return_int();
    if (n2 == 1)
    {
        savef = new Save_file(a, b,c,res, "res_tra.txt");
        n2 = 0;
        break;
    }

    if (n2 == 2)
    {
        n2 = 0;
        break;
    }
}
break;

```

case 5:

```

while (1) {
    cout << "какой файл открыть: \n1)Прямоугольник 2)Круг
3)Прямоугольный треугольник 4)Трапедия ";
    n2 = return_int();
    if (n2 == 1)
    {
        read_file(3,"res_rec.txt");
        break;
    }
    if (n2 == 2)
    {
        read_file(3, "res_cir.txt");
        break;
    }
    if (n2 == 3)
    {
        read_file(3, "res_tri.txt");
        break;
    }
    if (n2 == 4)
    {
        read_file(4, "res_tra.txt");
        break;
    }
}

```



```

    }
}

break;

default:break;
}
}

```

```

return 0;
}

```

3.4. Листинг файла Rectangle.cpp

```
#include "Rectangle.h"
```

```

Rectangle::Rectangle()
{
length = 0;
width = 0;
}

```

```

Rectangle::Rectangle(double a, double b):length(a), width(b)
{
}

```

```

double Rectangle::getSquare()
{

return length*width;
}

```

3.5. Листинг файла Save_file.cpp

```
#include "Save_file.h"
```

```
#include "Calculate.h"
```

```
Save_file::Save_file()
```

```

{
}

```

```
Save_file::Save_file(double a, double b, double f, const char name[12])
```

```
{
```

```
    ofstream fout(name); //создаем поток
```

```
    fout.close();
```

```
    fout.open(name, ios::app); // создаём объект класса ofstream для дополнения
```

записи

```
    fout << a << "\n"; // запись строки в файл
```

```
    fout << b << "\n"; // запись строки в файл
```

```
    fout << f << "\n"; // запись строки в файл

```

```

        fout.close(); // закрываем файл
        cout << "данные сохранены";
    }

Save_file::Save_file(double a, double b, double f, double c, const char name[12])
{
    ofstream fout(name); //создаем поток
    fout.close();
    fout.open(name, ios::app); // создаём объект класса ofstream для дополнения
записи
    fout << a << "\n"; // запись строки в файл
    fout << b << "\n"; // запись строки в файл
    fout << f << "\n"; // запись строки в файл
    fout << c << "\n"; // запись строки в файл

    fout.close(); // закрываем файл
    cout << "данные сохранены";
}

```

3.6. Листинг файла Trapezoid.cpp

```
#include "Trapezoid.h"
```

```

Trapezoid::Trapezoid()
{
    side1 = 0;
    side2 = 0;
    height = 0;
}

```

```

Trapezoid::Trapezoid(double a, double b, double h):side1(a),side2(b),height(h)
{
}

```

```

double Trapezoid::getSquare()
{
    return (side1+side2)/2*height;
}

```

3.7. Листинг файла Triangle.cpp

```
#include "Triangle.h"
```

```

Triangle::Triangle()
{
    side1 = 0;
    side2 = 0;
}

```

```

Triangle::Triangle(double a, double b):side1(a), side2(b)
{
}

double Triangle::getSquare()
{
return side1*side2/2;
}

```

3.8. Листинг файла Calculate.h

```

#include <fstream>
#include <iostream>
using namespace std;
#pragma once
class Calculate
{

public:
virtual double getSquare() = 0;
virtual ~Calculate(); //деструктор
};

```

3.9. Листинг файла Circle.h

```

#include "Calculate.h"
#pragma once
class Circle:public Calculate
{
private:
double r;
double pi;

public:
Circle();
Circle(double r);
double getSquare()override;
};

```

3.10. Листинг файла Rectangle.h

```

#include "Calculate.h"
#pragma once
class Rectangle: public Calculate
{
private:
double length;
double width;

public:

```

```
Rectangle();  
Rectangle(double a, double b);  
double getSquare()override;  
};
```

3.11. Листинг файла Save_file.h

```
#include "Calculate.h"  
#pragma once  
class Save_file  
{  
public:  
    Save_file();  
  
    Save_file(double a, double b, double f, const char name[12]);  
    Save_file(double a, double b, double c, double f, const char name[12]);  
};
```

3.12. Листинг файла Trapezoid.h

```
#include "Calculate.h"  
#pragma once  
class Trapezoid:public Calculate  
{  
private:  
    double side1;  
    double side2;  
    double height;  
  
public:  
    Trapezoid();  
    Trapezoid(double a, double b, double h);  
    double getSquare()override;  
};
```

3.13. Листинг файла Triangle.h

```
#include "Calculate.h"  
#pragma once  
class Triangle:public Calculate  
{  
private:  
    double side1;  
    double side2;  
  
public:  
    Triangle();  
    Triangle(double a, double b);  
    double getSquare()override;  
};
```

4. Результаты работы программы

```
Посчитать площадь для:
1)Прямоугольник 2)Круг 3)Прямоугольный треугольник 4)Трапеция 5) открыть файл 0)выход
номер операции = 1
ok
введите длину 23
ok
введите ширину 34
ok
площадь = 782
сохранить результат? 1-да, 2-нет
сохранение: номер операции = 1
ok
данные сохранены
номер операции = 2
ok
введите радиус 33
ok
площадь = 103.62
сохранить результат? 1-да, 2-нет
сохранение: номер операции = 2
ok

номер операции = 3
ok
введите 1 катет 5
ok
введите 2 катет 6
ok
площадь = 15
сохранить результат? 1-да, 2-нет
сохранение: номер операции = 1
ok
данные сохранены
номер операции = 4
ok
введите 1 основание 4
ok
введите 2 основание 4
ok
введите высоту 4
ok
площадь = 16
сохранить результат? 1-да, 2-нет
сохранение: номер операции = 1
ok
данные сохранены
номер операции =
```

Рисунок 1 – Ввод данных.

Сначала выбираем фигуру, для которой будем рассчитывать площадь. Дальше вводим числа, каждый раз нажимая Enter. После ввода всех параметров, для расчета площади, будет выведена площадь фигуры, и программа предложит сохранить информацию. Если нажать 1, то программа сохранит .txt файл в директории проекта. Если нажать 2, программа продолжит работу, файл не будет сохранен. В примере выше, параметры круга не были сохранены.

```
Посчитать площадь для:
1)Прямоугольник 2)Круг 3)Прямоугольный треугольник 4)Трапеция 5) открыть файл 0)выход
номер операции = 5
ok
какой файл открыть:
1)Прямоугольник 2)Круг 3)Прямоугольный треугольник 4)Трапеция 1
ok
первое значение = 23
второе значение = 34
площадь = 782

номер операции = 5
ok
какой файл открыть:
1)Прямоугольник 2)Круг 3)Прямоугольный треугольник 4)Трапеция 2
ok
Файла нет
номер операции = 5
ok
какой файл открыть:
1)Прямоугольник 2)Круг 3)Прямоугольный треугольник 4)Трапеция 3
ok
первое значение = 5
второе значение = 6
площадь = 15

номер операции = 5
ok
какой файл открыть:
1)Прямоугольник 2)Круг 3)Прямоугольный треугольник 4)Трапеция 4
ok
первое значение = 4
второе значение = 4
третье значение = 4
площадь = 16

номер операции =
```

Рисунок 2 – Чтение файла.

Для чтения данных из файла нужно выбрать 5. Далее нужно выбрать какой файл (фигуру) нужно будет прочитать. В результате будет выведена информация из файла. Если файл будет не найден, выведется соответствующее сообщение. В примере выше файл круг не был сохранен, поэтому выведено сообщение что файла нет.

```

Посчитать площадь для:
1)Прямоугольник 2)Круг 3)Прямоугольный треугольник 4)Трапеция 5) открыть файл 0)выход
номер операции = 0
введите число = 2
введите число = 1
ok
введите длину аа
введите число = 23
введите число = 23
ok
введите ширину ак
введите число = 4
ok
площадь = 92
сохранить результат? 1-да, 2-нет
сохранение: номер операции = 232
ok

сохранение: номер операции = 0
введите число = 22
введите число = 2
введите число = 1
ok
данные сохранены
номер операции =

```

Рисунок 3 – Исключения.

Для корректной работы программы нужно вводить положительные числа. Если ввести буквы или отрицательные числа, программа это обработает и будет предлагать ввести корректное значение для продолжения работы.

```

номер операции = 0
ok
C:\Users\user\Desktop

```

Рисунок 4 – Завершение программы.

5. Вывод

В результате выполнения лабораторной работы я изучила работу с абстрактными классами в языке программирования C++. Была создана программа для расчета площади для 4 фигур, а также сохранение и чтение данных из файла.

Программа соответствует техническому заданию и работает исправно.