Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им. И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление: 710400 «Программная инженерия»

Дисциплина: «Объектно – ориентированное программирование»

**ОТЧЕТ**

# По лабораторной работе №7.

Тема: «Виртуальные методы.»

                                                                       Выполнила: студент группы

ПИ(б)-2-21 Марат к. Жылдыз

Проверил: Мусабаев Э.Б.

**Бишкек – 2024**

**Лабораторная работа №7.**

**Тема: Виртуальные методы.**

**Задания:**

1. Создайте базовый класс **Array1** с закрытыми атрибутами: два массива типа unsigned одинаковой размерности и поле для хранения количества элементов массива. Максимально-возможный размер массива задается статической константой.

Реализуйте в базовом классе виртуальный метод поэлементного сложения массивов.

Разработайте производный класс **Array2** с закрытыми атрибутами: два массива типа unsigned одинаковой размерности и поле для хранения количества элементов масс.

Переопределите виртуальный метод поэлементного сложения массивов в производном классе Array2.

**Код программы:**

#include<iostream>

#include<Windows.h>

using namespace std;

class Array {

protected:

unsigned arr[100], n;

double sum = 0;

public:

Array() {

cin >> n;

cout << "Введите данные: ";

for (unsigned i = 0; i < n; i++) {

cin >> arr[i];

}

}

virtual double plus() {

for (unsigned i = 0; i < n; i++) {

sum+= arr[i];

}

return sum;

}

};

class Array1 :public Array {

public:

Array1() :Array(){}

double plus() {

for (unsigned i = 0; i < n; i++) {

sum += arr[i];

}

sum = sum / 2;

cout << "Сложение элементов деленные на два: " << sum << endl;

return sum;

}

};

class Array2 :public Array {

public:

Array2() :Array() {}

double plus() {

for (unsigned i = 0; i < n; i++) {

sum += arr[i] \* 2;

}

cout << "Сложение элементов умноженное на два: " << sum << endl;

return sum;

}

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Количество элементов первого производного класса: ";

Array1 A;

cout << "Количество элементов второго производного класса: ";

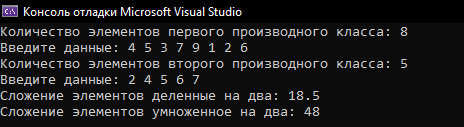
Array2 B;

A.plus();

B.plus();

}

**Результат программы:**



1. Создать абстрактный базовый класс Figure с виртуальными методами вычисления площади и периметра. Создать производные класса: Rectangle (прямоугольник), Circle (круг), Trapezium (трапеция) со своими функциями площади и периметра. Самостоятельно определить, какие поля необходимы, какие из них можно задать в базовом классе, а какие в производных. Площадь трапеции: *S = (a + b) x h/2.*

**Код программы:**

#include<iostream>

#include<Windows.h>

using namespace std;

class Figure {

protected:

double S, P;

public:

Figure() { S = 0; P = 0; }

virtual double area() = 0;

virtual double perimea() = 0;

};

class Circle :public Figure {

private:

int r;

const double pi = 3.1415;

public:

Circle() :Figure() {

cout << "Круг:" << endl;

cout << "Введите радиус: ";

cin >> r;

}

double area() {

return pi \* r \* r;

}

double perimea() {

return 2 \* pi \* r;

}

};

class Rectangl :public Figure {

private:

int a, b;

public:

Rectangl() :Figure() {

cout << "Прямоугольник:" << endl;

cout << "Введите сторону а: ";

cin >> a;

cout << "Введите сторону b: ";

cin >> b;

}

double area() {

S = a \* b;

return S;

}

double perimea() {

P = 2 \* (a + b);

return P;

}

};

class Trapezium :public Figure {

private:

double a, b, c, d, h;

public:

Trapezium() :Figure() {

cout << "Трапеция:" << endl;

cout << "Введите сторону а: ";

cin >> a;

cout << "Введите сторону b: ";

cin >> b;

cout << "Введите сторону c: ";

cin >> c;

cout << "Введите сторону d: ";

cin >> d;

cout << "Введите высоту h:";

cin >> h;

}

double area() {

S = (a + b)\*h/2;

return S;

}

double perimea() {

P = a + b+c+d;

return P;

}

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Circle A;

Rectangl B;

Trapezium C;

cout << "Прямоугольник:" << endl;

cout << "Площадь: " << B.area() << "\tПериметр: " << B.perimea() << endl;

cout << "Круг:" << endl;

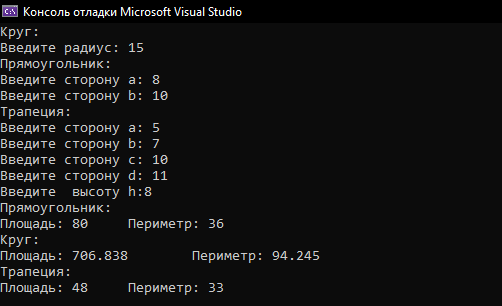
cout << "Площадь: " << A.area() << "\tПериметр: " << A.perimea() << endl;

cout << "Трапеция:" << endl;

cout << "Площадь: " << C.area() << "\tПериметр: " << C.perimea() << endl;

}

**Результат программы:**

****

1. Создать абстрактный базовый класс Currency (валюта) для работы с денежными суммами. Определить виртуальные функции перевода в рубли и вывода на экран. Реализовать производные классы Dollar (доллар) и Euro (евро) со своими функциями перевода и вывода на экран.

**Код программы:**

#include<iostream>

#include<Windows.h>

using namespace std;

class Currency {

public:

virtual double ruble() = 0;

};

class Dollar :public Currency {

private:

int d;

double b;

public:

Dollar() {

cout << "Введите счет доллара: ";

cin >> d;

cout << "курс валюты доллара: ";

cin >> b;

}

double ruble() {

return d \* b;

}

};

class Euro :public Currency {

private:

int e;

double a;

public:

Euro() {

cout << "Введите счет евро: ";

cin >> e;

cout << "курс валюты евро: ";

cin >> a;

}

double ruble() {

return e \* a;

}

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Dollar A;

Euro B;

int d, e;

cout << "В рублях: " << A.ruble()<<" рубль"<<endl;

cout << "В рублях: " << B.ruble() << " рубль"<<endl;

}

**Результат программы:**

