**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСТИТЕТ**

**им. И.Раззакова**

**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Кафедра «**Программное обеспечение компьютерных систем**»

Направление: 710400 «**Программная инженерия**»

Дисциплина: ««**Объектно-ориентированное программирование**»»

ОТЧЕТ

Лабораторная работа №6

Тема: Наследование

Выполнил: Кудайбердиев Эрлан

Группа: ПИ-2-21

Проверил: Мусабаев Э. Б.

Бишкек – 2024

**Задание №1**

Создать базовый класс Array с полями: массив типа unsigned и поле для хранения количества элементов у текущего объекта массива. Максимально возможный размер массива задается статической константой. Реализовать конструктор инициализации, задающий количество элементов и начальное значение (по умолчанию 0). Реализовать в классе Array виртуальную функцию поэлементного сложения массивов. Реализовать два класса, переопределив виртуальную функцию сложения. Вызывающая программа должна продемонстрировать все варианты вызова виртуальных функций.

**Код программы:**

#include <iostream>

using namespace std;

const int MAX\_SIZE = 100;

class Array {

public:

unsigned int elements[MAX\_SIZE];

unsigned int size;

public:

Array(unsigned int initialSize, unsigned int initValue = 0) : size(initialSize) {

for (unsigned int i = 0; i < size; ++i) {

elements[i] = initValue;

}

}

virtual void add(const Array& other) {

for (unsigned int i = 0; i < size && i < other.size; ++i) {

elements[i] += other.elements[i];

}

}

void display(unsigned int initialSize) {

for (unsigned int i = 0; i < 5; ++i) {

cout << elements[i] << " ";

}

cout << endl;

}

};

class ArrayA : public Array {

public:

ArrayA(unsigned int initialSize, unsigned int initValue = 0) : Array(initialSize, initValue) {}

void add(const Array& other) override {

for (unsigned int i = 0; i < size && i < other.size; ++i) {

elements[i] -= other.elements[i];

}

}

};

class ArrayB : public Array {

public:

ArrayB(unsigned int initialSize, unsigned int initValue = 0) : Array(initialSize, initValue) {}

void add(const Array& other) override {

for (unsigned int i = 0; i < size && i < other.size; ++i) {

elements[i] \*= other.elements[i];

}

}

};

int main() {

setlocale(LC\_ALL, "rus");

cout << "Объекты а и b класса Array:\n";

Array a(5, 2);

Array b(5, 3);

cout << " Элементы массива а:";

a.display(5);

cout << " Элементы массива b:";

b.display(5);

cout << " Элементы массива a после поэлементного сложения с массивом b:";

a.add(b);

a.display(5);

cout << "--------------------------------------------------------------------------------\n";

cout << "Объекты c и d класса ArrayA:\n";

ArrayA c(5, 2);

ArrayB d(5, 2);

cout << " Элементы массива c:";

c.display(5);

cout << " Элементы массива d:";

d.display(5);

cout << " Элементы массива a после поэлементного вычитания с массивом b:";

c.add(d);

c.display(5);

cout << "--------------------------------------------------------------------------------\n";

cout << "Объекты e и f класса ArrayB:\n";

ArrayB e(5, 2);

ArrayB f(5, 2);

cout << " Элементы массива e:";

e.display(5);

cout << " Элементы массива f:";

f.display(5);

cout << " Элементы массива e после поэлементного умножения c массивом f:";

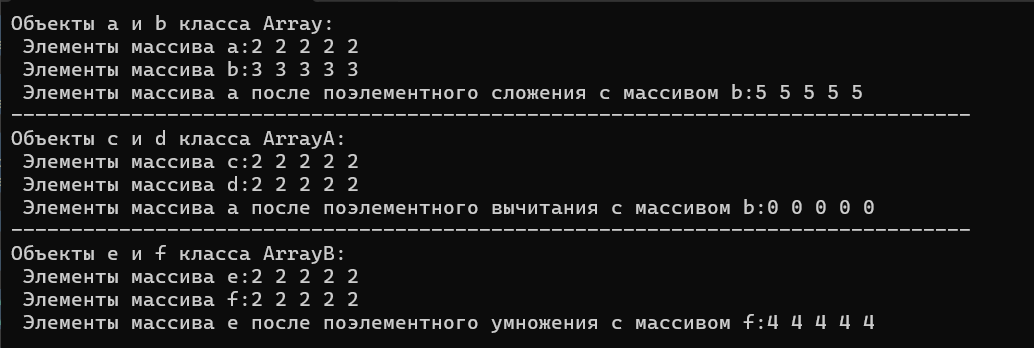
e.add(f);

e.display(5);

return 0;

}

**Результат:**



**Задание №2**

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>

using namespace std;

class Figure {

public:

virtual double area() const = 0;

virtual double perimeter() const = 0;

};

class Rectangle : public Figure {

private:

double width;

double height;

public:

Rectangle(double w, double h) : width(w), height(h) {}

double area() const override {

return width \* height;

}

double perimeter() const override {

return 2 \* (width + height);

}

};

class Circle : public Figure {

private:

double radius;

public:

Circle(double r) : radius(r) {}

double area() const override {

return 3.14159265 \* radius \* radius;

}

double perimeter() const override {

return 2 \* 3.14159265 \* radius;

}

};

class Trapezium : public Figure {

private:

double side\_a;

double side\_b;

double height;

public:

Trapezium(double a, double b, double h) : side\_a(a), side\_b(b), height(h) {}

double area() const override {

return (side\_a + side\_b) \* height / 2.0;

}

double perimeter() const override {

return side\_a + side\_b + 2 \* (height / 2);

}

};

int main() {

setlocale(LC\_ALL, "rus");

Rectangle rect(4, 5);

Circle circle(3);

Trapezium trap(2, 4, 3);

cout << "Площадь прямоугольника: " << rect.area() << ", Периметр: " << rect.perimeter() <<endl;

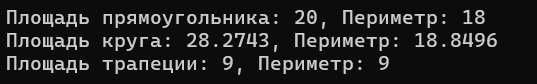
cout << "Площадь круга: " << circle.area() << ", Периметр: " << circle.perimeter() << endl;

cout << "Площадь трапеции: " << trap.area() << ", Периметр: " << trap.perimeter() << endl;

return 0;

}

**Результат:**



**Задание №3**

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>

using namespace std;

class Currency {

public:

virtual double toRubles(double amount) const = 0;

virtual void display(double amount) const = 0;

};

class Dollar : public Currency {

public:

double toRubles(double amount) const override {

return amount \* 75.0;

}

void display(double amount) const override {

cout << "Значение доллара в рублях после перевода : " << toRubles(amount) << endl;

cout << endl;

}

};

class Euro : public Currency {

public:

double toRubles(double amount) const override {

return amount \* 90.0;

}

void display(double amount) const override {

cout << "Значение евро в рублях после перевода : " << toRubles(amount) << endl;

cout << endl;

}

};

int main() {

setlocale(LC\_ALL,"rus");

Dollar dollar;

Euro euro;

double amount;

cout << "Введите сумму в долларах: ";

cin >> amount;

dollar.display(amount);

cout << "Введите сумму в евро: ";

cin >> amount;

euro.display(amount);

return 0;

}

**Результат:**

