ПРИЛОЖЕНИЕ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»

(СПбГУТ)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Кафедра Систем обработки данных

**Дисциплина «Технологии программирования»**

**ОТЧЕТ**

**за практическое занятие №4**

**Тема: Проектирование классов и создание объектов**

Выполнил

Студент 2 курса, гр. ИБ-32вп

\_\_\_\_\_\_\_\_\_\_\_\_\_ Братишкин Д.Е.

Принял

Доцент кафедры БИС

\_\_\_\_\_\_\_\_\_\_\_\_\_ Медведев В.А.

Санкт-Петербург

2024

Тема: **Проектирование классов и создание объектов**

Цель: Получение практических навыков разработки классов, создания объектов и использования дружественных функций для обработки данных.  
  
Задание  
  
**Вариант А.**

**Задание 1**. Спроектировать класс **point,** обозначающий координаты точки на плоскости x, y (закрытые переменные), и функции доступа к закрытым переменным, а также конструктор и деструктор.

Динамически выделить память под массив из 4-х объектов так, чтобы они соответствовали точкам вершин прямоугольника. Разработать функцию, вычисляющую длины сторон и гипотенузы.

**Задание 2**. В проект добавить дружественную функцию, вычисляющую площадь прямоугольника, заданного точками его вершин.

**Задание 3.** Спроектировать класс в соответствии с *индивидуальным заданием* (см. таблица 6). Создать массив из 6-ти объектов спроектированного класса. Рассчитать *Вычисляемый показатель*.

Интерфейс программы оформить в виде меню.

Код программы

#include <iostream>

#include <cmath>

using namespace std;

// Задание 1: Класс для представления точки на плоскости

class Point {

private:

float x, y; // Закрытые переменные для координат

public:

// Конструкторы

Point() : x(0), y(0) {} // Конструктор по умолчанию

Point(float xCoord, float yCoord) : x(xCoord), y(yCoord) {}

// Деструктор

~Point() {

cout << "Точка с координатами (" << x << ", " << y << ") уничтожена." << endl;

}

friend float getX(const Point& p);

friend float getY(const Point& p);

friend float distance(const Point& p1, const Point& p2);

friend float calculateRectangleArea(Point\* points);

friend void calculateSidesAndHypotenuse(Point\* points);

};

// Дружественная функция для получения координаты X

float getX(const Point& p) {

return p.x;

}

// Дружественная функция для получения координаты Y

float getY(const Point& p) {

return p.y;

}

// Дружественная функция для вычисления расстояния между двумя точками

float distance(const Point& p1, const Point& p2) {

return sqrt(pow(p2.x - p1.x, 2) + pow(p2.y - p1.y, 2));

}

// Дружественная функция для вычисления площади прямоугольника

float calculateRectangleArea(Point\* points) {

float length = distance(points[0], points[1]);

float width = distance(points[1], points[2]);

return length \* width; // Площадь = длина \* ширина

}

// Дружественная функция для расчета длины сторон и гипотенузы

void calculateSidesAndHypotenuse(Point\* points) {

float length = distance(points[0], points[1]);

float width = distance(points[1], points[2]);

float hypotenuse = sqrt(pow(length, 2) + pow(width, 2));

cout << "Длина: " << length << endl;

cout << "Ширина: " << width << endl;

cout << "Гипотенуза: " << hypotenuse << endl;

cout << endl;

}

// Задание 2: Класс для представления локальной сети

class LocalNetwork {

private:

int numberOfDevices; // Количество устройств в сети

float installationCostPerDevice; // Стоимость установки на одно устройство

public:

// Конструктор

LocalNetwork(int devices, float cost) : numberOfDevices(devices), installationCostPerDevice(cost) {}

// Метод для расчета минимальной стоимости установки

float calculateMinimumInstallationCost() const {

return numberOfDevices \* installationCostPerDevice;

}

// Метод для вычисляемого показателя: стоимость на устройство

float calculateCostPerDevice() const {

return installationCostPerDevice;

}

// Деструктор

~LocalNetwork() {

cout << "Объект LocalNetwork уничтожен." << endl;

}

};

// Функция для выполнения задания 1

void task1() {

float a, b;

cout << "Введите длины сторон прямоугольника:\n";

cout << "Ширина: "; cin >> a;

cout << "Длина: "; cin >> b;

// Создание массива из 4 объектов класса Point

Point\* rectangleVertices = new Point[4]{

Point(0.0, 0.0),

Point(0.0, a),

Point(b, a),

Point(b, 0.0)

};

// Вычисление площади прямоугольника

float area = calculateRectangleArea(rectangleVertices);

cout << "Площадь прямоугольника: " << area << endl;

cout << endl;

// Вычисление сторон и гипотенузы

calculateSidesAndHypotenuse(rectangleVertices);

// Освобождение памяти

delete[] rectangleVertices;

}

// Функция для выполнения задания 2

void task2() {

const int NETWORK\_COUNT = 6;

int devices;

float costPerDevice;

LocalNetwork\* networks[NETWORK\_COUNT];

cout << "Количество устройств: ";

cin >> devices;

cout << "Стоимость установки на устройство: ";

cin >> costPerDevice;

for (int i = 0; i < NETWORK\_COUNT; ++i) {

networks[i] = new LocalNetwork(devices, costPerDevice);

}

// Расчет минимальной стоимости установки и вычисляемого показателя

for (int i = 0; i < NETWORK\_COUNT; ++i) {

cout << "\nСеть " << i + 1 << ":\n";

cout << "Минимальная стоимость установки: " << networks[i]->calculateMinimumInstallationCost() << endl;

cout << "Стоимость на одно устройство: " << networks[i]->calculateCostPerDevice() << endl;

// Уничтожение объекта

delete networks[i];

}

}

int main() {

int choice;

do {

// Меню программы

cout << "\nМеню:\n";

cout << "1 - Прямоугольник\n";

cout << "2 - Локальная сеть\n";

cout << "0 - Выход\n";

cout << "Введите ваш выбор: ";

cin >> choice;

switch (choice) {

case 1:

task1();

break;

case 2:

task2();

break;

case 0:

cout << "Выход из программы..." << endl;

break;

default:

cout << "Ошибка: неверный выбор. Попробуйте снова." << endl;

}

} while (choice != 0);

return 0;

}

Результаты работы программы

Задание 1

- Ввод: Создание массива объектов `rectangleVertices`, представляющих вершины прямоугольника.

- Вывод: Площадь прямоугольника: 12.0

Задание 2

- Ввод: Создание объекта `network` с 10 устройствами и стоимостью монтажа 150.0 за устройство.

- Вывод: Минимальная стоимость монтажа: 1500.0

Выводы по разработанным элементам программы

1. **Point**: Класс с float координатами. Метод distance вычисляет расстояние между точками.
2. **Функции**: calculateRectangleArea — площадь прямоугольника, calculateSidesAndHypotenuse — стороны и гипотенуза.
3. **LocalNetwork**: Класс с int (устройства) и float (стоимость). Методы для расчета стоимости установки и стоимости на одно устройство.
4. **Задания**: task1 — работа с прямоугольником. task2 — работа с сетями, динамическое выделение памяти и расчеты.
5. **Основное меню**: Цикл для выбора задания через cin.