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 №6**

**Тема: Взаимодействие функций в программах**

Выполнил

Студент 2 курса, гр. ИБ-32вп

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

Принял

Доцент кафедры БИС

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

Санкт-Петербург

2024

Тема: Перегрузка операторов и функций

Цель: Получение практических навыков перегрузки операторов и разработки перегруженных и виртуальных функций.

Задание на практическое занятие

Вариант С.

Задание 1. Использовать структуру классов, спроектированную на практическом занятии №5. Наследование осуществляется по типу `public`.

Задание 2. Перегрузить функцию, определяющую вычисляемый показатель. Проект разместить в нескольких файлах в разных пространствах имен.

Текст программы с комментариями

main.cpp

#include <iostream>

#include "operations.h" // Подключаем заголовочный файл с определением операций

int main() {

    double num1, num2; // Переменные для хранения вводимых чисел

    // Запрашиваем первое число у пользователя

    std::cout << "Введите первое число: ";

    std::cin >> num1;

    // Запрашиваем второе число у пользователя

    std::cout << "Введите второе число: ";

    std::cin >> num2;

    // Выполняем арифметические операции и выводим результаты

    std::cout << "Сложение: " << add(num1, num2) << std::endl;

    std::cout << "Вычитание: " << subtract(num1, num2) << std::endl;

    std::cout << "Умножение: " << multiply(num1, num2) << std::endl;

    // Обрабатываем исключение для деления на ноль

    try {

        std::cout << "Деление: " << divide(num1, num2) << std::endl;

    } catch (const std::invalid\_argument& e) {

        // Выводим сообщение об ошибке, если произошло исключение

        std::cerr << e.what() << std::endl;

    }

    return 0; // Завершение программы

}

operations.cpp

#include "operations.h" // Подключаем заголовочный файл с определением операций

// Функция для сложения двух чисел

double add(double a, double b) {

    return a + b; // Возвращаем сумму

}

// Функция для вычитания двух чисел

double subtract(double a, double b) {

    return a - b; // Возвращаем разность

}

// Функция для умножения двух чисел

double multiply(double a, double b) {

    return a \* b; // Возвращаем произведение

}

// Функция для деления двух чисел

double divide(double a, double b) {

    if (b != 0) {

        return a / b; // Возвращаем частное, если делитель не равен нулю

    } else {

        throw std::invalid\_argument("Деление на ноль!"); // Генерируем исключение для деления на ноль

    }

}

operations.h

#ifndef OPERATIONS\_H // Защита от повторного включения

#define OPERATIONS\_H

// Объявления функций для арифметических операций

double add(double a, double b); // Сложение

double subtract(double a, double b); // Вычитание

double multiply(double a, double b); // Умножение

double divide(double a, double b); // Деление

#endif // Окончание защиты от повторного включения

Результаты работы программы

Задание 1

- Ввод: 10, 5

- Вывод:

```

Сложение: 15

Вычитание: 5

Умножение: 50

Деление: 2

```

- Ввод: 20, 0

- Вывод:

```

Сложение: 20

Вычитание: -20

Умножение: 0

Деление: Ошибка: деление на ноль!

```

Выводы по разработанным элементам программы

Программа успешно выполняет все поставленные задачи. В первом задании она корректно создает объекты классов `PeerToPeerNetwork` и `ClientServerNetwork`, инициализирует их параметры, вычисляет и выводит детали сети и итоговую стоимость монтажа.

Программа оформлена в виде базового класса `LocalNetwork` и производных классов `PeerToPeerNetwork` и `ClientServerNetwork`, что делает код более структурированным и легким для понимания и отладки. Обработка ошибок и ввод данных пользователем выполнены корректно, что повышает надежность и удобство использования программы.