ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

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

(СПбГУТ)

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

Факультет Информационных технологий и программной инженерии

Кафедра Систем обработки данных

**Дисциплина «Технологии программирования»**

**ОТЧЕТ**

**за практическое занятие №4**

**Тема: Проектирование классов и создание объектов**

Выполнил

Студент 2 курса, гр. ИСТ-422

\_\_\_\_\_\_\_\_\_\_\_\_\_ Ващук А.А.

Принял

Доцент кафедры БИС

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

Санкт-Петербург

2025

**Тема и цель практического занятия:**

Получение практических навыков разработки классов, создания объектов и использования дружественных функций для обработки данных.

**Задание на практическое занятие:**

*Выберите один из вариантов задания.*

**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.

|  |  |  |
| --- | --- | --- |
|  | Документ | Количество документов, выданных в прошлом году |

**Текст программы:**

#include <iostream>

#include <cmath>

class point

{

private:

    int x, y;

public:

    point(int a = 0, int b = 0) : x(a), y(b) {}

    ~point() {}

    void set\_x(int a) { x = a; }

    void set\_y(int b) { y = b; }

    int get\_x() const { return x; }

    int get\_y() const { return y; }

    friend int Square(point\* points);

};

class document

{

private:

    int num, year;

public:

    document(int a = 0, int b = 0) : num(a), year(b) {}

    ~document() {}

    void set\_num(int a) { num = a; }

    void set\_year(int b) { year = b; }

    int get\_num() const { return num; }

    int get\_year() const { return year; }

};

int Square(point\* points)

{

    int side1 = std::abs(points[0].y - points[1].y);

    int side2 = std::abs(points[0].x - points[2].x);

    return side1 \* side2;

}

void SideLength(point\* points)

{

    int side1 = std::abs(points[0].get\_y() - points[1].get\_y());

    int side2 = std::abs(points[0].get\_x() - points[2].get\_x());

    int side3 = std::abs(points[2].get\_y() - points[3].get\_y());

    int side4 = std::abs(points[1].get\_x() - points[3].get\_x());

    int diagonal = std::sqrt((side1 \* side1) + (side2 \* side2));

    std::cout << "Side 1: " << side1 << std::endl;

    std::cout << "Side 2: " << side2 << std::endl;

    std::cout << "Side 3: " << side3 << std::endl;

    std::cout << "Side 4: " << side4 << std::endl;

    std::cout << "Diagonal: " << diagonal << "\n\n";

}

int check(document\* docMs)

{

    int count = 0;

    for (int i = 0; i < 6; i ++)

    {

        if (docMs[i].get\_year() == 2024) count += docMs[i].get\_num();

    }

    return count;

}

int main()

{

    point\* m = new point[4];

    m[0].set\_x(0); m[0].set\_y(0);

    m[1].set\_x(0); m[1].set\_y(3);

    m[2].set\_x(4); m[2].set\_y(0);

    m[3].set\_x(4); m[3].set\_y(3);

    document\* docM = new document[6];

    char button;

    int S, docA, docB;

    std::cout << "1. Length and Hypotenuse\n2. Square\n3. Document\nC. Exit\nInput: ";

    std::cin >> button;

    while (button != 'C')

    {

        system("cls");

        switch (button)

        {

        case '1':

            SideLength(m);

            break;

        case '2':

            S = Square(m);

            std::cout << "Square: " << S << "\n\n";

            break;

        case '3':

            system("cls");

            for (int i = 0; i < 6; i++)

            {

                std::cout << "year and num: ";

                std::cin >> docA >> docB;

                docM[i].set\_year(docA);

                docM[i].set\_num(docB);

            }

            std::cout << "Number of documents issued last year: " << check(docM) << "\n";

            break;

        default:

            if (button != 'C')

                std::cout << "Invalid input. Try again.\n";

            break;

        }

        std::cout << "1. Length and Hypotenuse\n2. Square\n3. Document\nC. Exit\nInput: ";

        std::cin >> button;

    }

    delete[] m;

    delete[] docM;

    return 0;

}

**Выводы по созданному проекту и использованию средств языка программирования:**

В программе были использованы такие технологии, как: классы и объекты, динамическое выделение памяти, дружественные функции.

Классы и объекты использовались для работы с координатами точек и данными о документах.

Динамическая память была использована для создания массивов объектов классов point и document.

Для взаимодействия с пользователем и выбора различных функций программы, в программе реализовано меню.