

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

Институт непрерывного образования
Кафедра Безопасности информационных систем

Дисциплина «Технологии программирования»

ОТЧЕТ
За практическое занятие №5

Тема: Проектирование классов и создание объектов
Вариант 9 «Б»

Выполнил:

Студент 2 курса, гр. ИБ-12з:

Кондылев А.А.

Принял:

Доцент кафедры БИС

Медведев В.А.

Тема: Проектирование классов и создание объектов

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

Задание 1. Спроектировать класс **point**, обозначающий координаты точки на плоскости: x,y. Разработать:

а) конструктор с параметрами - для инициализации объектов класса **point**;

б) функции доступа к закрытым членам класса;

в) деструктор, обозначающий момент уничтожения объекта.

Создать два объекта и продемонстрировать работу методов класса.

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

9.	Представитель университета	Количество студентов, обучающихся у конкретного преподавателя
----	----------------------------	---

Исходный код:

Заголовочный файл point.h:

```
#ifndef HEADER_POINTCLASS
#define HEADER_POINTCLASS

class Point
{
private:
```

```

    int x;
    int y;
    float v_length;

    void calc_length();

public:
    Point(int x, int y) :x(x),y(y) {this->calc_length();}

    int get_x() const {return this->x;}

    int get_y() const {return this->y;}

    float vector_length() const {return this->v_length;}

    void set_coors(int x, int y);

    ~Point() {cout << "Point(" << this << ") was deleted." << endl;}

};

#endif

```

Содержимое файла point.cpp:

```

#include <iostream>
using namespace std;
#include "point.h"

void Point::set_coors(int x, int y)
{
    this->x = x;
    this->y = y;
    this->calc_length();
}

void Point::calc_length()
{
    this->v_length = sqrt(pow(x, 2) + pow(y, 2));
}

```

Заголовочный файл university.h:

```

#include <iostream>
#include <vector>

#ifndef HEADER_UNICLASS
#define HEADER_UNICLASS

// Класс института для легенды

```

```

class University
{
private:
    string uni_name;
public:
    University(string name) : uni_name(name) {};
    string get_name() const { return uni_name; }
};

// Базовый класс для студента и преподавателя
class Person
{
protected:
    string name;
    string last_name;
    unsigned short age;

public:
    Person(string name, string last_name, unsigned short age)
        : name(name), last_name(last_name), age(age)
    {}
};

// Класс студента
class Student : public Person
{
    friend class Teacher;
private:
    int group;
    const University& university;
    vector<Teacher*> teachers;

public:
    Student(string name, string last_name, unsigned short age, const Univer-
sity& university, int group)
        : Person(name, last_name, age), group(group), university(univer-
sity);

    ~Student();
};

// Класс преподавателя
class Teacher : public Person
{
    // Описание друж. функции
    friend size_t get_students(const Teacher& teacher);

private:
    string specialization;
    vector<Student*> students;
    const University& university;

public:
    Teacher(string name, string last_name, unsigned short age, const Univer-
sity& university, string specialization)
        : Person(name, last_name, age);

    string get_name() const { return last_name + " " + name; }
    // Добавление студента
    void add_student(Student* stu);

    // Удаление студента

```

```
void drop_student(const Student* std);};
```

```
#endif
```

Содержимое файла university.cpp:

```
#include <iostream>
using namespace std;
#include "university.h"
```

```
// Конструктор класса студента
Student::Student(string name, string last_name, unsigned short age, const Uni-
versity& university, int group)
    : Person(name, last_name, age), group(group), university(university)
{
    printf("Student: %s %s joined to group [%u] of the %s\n",
        last_name.c_str(), name.c_str(), group, universi-
ty.get_name().c_str());
}
```

```
// Конструктор класса преподавателя
Teacher::Teacher(string name, string last_name, unsigned short age, const
University& university, string specialization)
    : Person(name, last_name, age), university(university), specializa-
tion(specialization)
{
    printf("Teacher: %s %s joined to the %s as %s-teacher\n",
        last_name.c_str(), name.c_str(), universi-
ty.get_name().c_str(), specialization.c_str());
}
```

```
void Teacher::add_student(Student* stu)
{
    stu->teachers.push_back(this);
    students.push_back(stu);
};
```

```
// Деструктор студента. При удалении сообщает всем преподавателям
Student::~Student()
{
    printf("Student: %s %s left the %s\n",
        last_name.c_str(), name.c_str(), university.get_name().c_str());
    for (Teacher* t : teachers)
    {
        t->drop_student(this);
    }
}
```

```
// Удаление студента
```

```

void Teacher::drop_student(const Student* std)
{
    for (auto it = students.begin(); it != students.end(); it++)
    {
        if (*it == std) { students.erase(it); break; }
    }
}

// Определение дружественной функции по получению студентов у преподавателя
size_t get_students(const Teacher& teacher)
{
    return teacher.students.size();
}

```

Содержимое файла main.cpp:

```

#include <iostream>
#include "university.h"
#include "point.h"

using namespace std;

void ex1()
{
    printf("Exercise 1\n");
    Point a(10, 23);
    Point b(-5, -8);
    cout << a.vector_length() << endl;
    a.set_coors(-3, 0);
    b.set_coors(0, 0);
    cout << a.vector_length() << endl;
    cout << b.get_x() << endl;
}

void ex2()
{
    printf("Exercise 2\n");
    University spbgut("SPBGUT");
    Teacher t1("Andrey", "Kolobkov", 34, spbgut, "Mathematics");
    Teacher t2("Mikhail", "Lermontov", 48, spbgut, "History");

    Student* students = new Student[6]{
        {"Vitaliy", "Mamonov", 18, spbgut, 1},
        {"Dmitriy", "Lastochkin", 19, spbgut, 1},
        {"Evgeniy", "Chernov", 19, spbgut, 2},
        {"Anastasia", "Mironova", 19, spbgut, 2},
        {"Sergey", "Markov", 18, spbgut, 1},
        {"Andrey", "Kondylev", 20, spbgut, 2} };

    t1.add_student(&students[0]);
    t1.add_student(&students[1]);
    t1.add_student(&students[4]);

    t2.add_student(&students[2]);
    t2.add_student(&students[3]);
}

```

```

    t2.add_student (&students[5]);

    printf("Number of students at %s: %zu\n", t1.get_name().c_str(), get_students(t1));
    printf("Number of students at %s: %zu\n", t2.get_name().c_str(), get_students(t2));

    delete[] students;
}

int main()
{
    setlocale(LC_ALL, "ru");

    ex1();
    ex2();

    return 0;
}

```

Вывод программы:

```

Консоль отладки Microsoft Visual Studio
Exercise 1
25.0799
3
0
Point(0000006B0B7BF908) was deleted.
Point(0000006B0B7BF8D8) was deleted.

```

```

Консоль отладки Microsoft Visual Studio
Teacher: Kolobkov Andrey joined to the SPBGUT as Mathematics-teacher
Teacher: Lermontov Mikhail joined to the SPBGUT as History-teacher
Student: Mamonov Vitaliy joined to group [1] of the SPBGUT
Student: Lastochkin Dmitriy joined to group [1] of the SPBGUT
Student: Chernov Evgeniy joined to group [2] of the SPBGUT
Student: Mironova Anastasia joined to group [2] of the SPBGUT
Student: Markov Sergey joined to group [1] of the SPBGUT
Student: Kondylev Andrey joined to group [2] of the SPBGUT
Number of students at Kolobkov Andrey: 3
Number of students at Lermontov Mikhail: 3
Student: Kondylev Andrey left the SPBGUT
Student: Markov Sergey left the SPBGUT
Student: Mironova Anastasia left the SPBGUT
Student: Chernov Evgeniy left the SPBGUT
Student: Lastochkin Dmitriy left the SPBGUT
Student: Mamonov Vitaliy left the SPBGUT

```