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 №6**

**Тема: Перегрузка операторов и функций**

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 перегрузки операторов и разработки перегруженных и виртуальных функций.

**Задание на практическое занятие:**

*Выберите один из вариантов задания.*

**Вариант А.**

**Задание 1.** Использовать структуру классов, спроектированную на практическом занятии №5. Наследование одного из производных классов осуществить по типу **private**. *Перегрузить оператор* для работы с объектами классов и использовать его для определения *Вычисляемого показателя*.

**Задание 2.** Перегрузить функцию, определяющую *Вычисляемый показатель*. В проект включить *виртуальную функцию,* выводящую на экран характеристики объекта каждого класса (кроме наследника с типом **private**). Добавить в проект статический член в каждый класс, обозначающий общее свойство для всех объектов класса.

Проект разместить в нескольких файлах *в разных пространствах имён.*

**Задание 3.** Создать массивы объектов спроектированных классов и разработать меню для демонстрации содержимого проекта.

**Текст программы:**

#include <iostream>

#include <vector>

#include <limits>

#include "../include/Doxbase.h"

#include "../include/Passport.h"

#include "../include/StudBilet.h"

using namespace DoxNamespace;

using namespace PassportNamespace;

using namespace StudBiletNamespace;

void show\_menu() {

    std::vector<Passport> passports;

    std::vector<StudBilet> stud\_bilets;

    int choice;

    while(choice != 6) {

        std::cout << "\n=== Document Manager ===\n"

                  << "1. Add New Passport\n"

                  << "2. Add Student Ticket\n"

                  << "3. List All Documents\n"

                  << "4. Search by Year\n"

                  << "6. Exit\n"

                  << "Select option: ";

        std::cin >> choice;

        system("cls");

        switch(choice) {

            case 1: {

                int y; char s;

                std::cout << "Enter issue year: ";

                std::cin >> y;

                std::cout << "Enter passport series: ";

                std::cin >> s;

                passports.emplace\_back(y, s);

                break;

            }

            case 2: {

                int y, num;

                std::cout << "Enter issue year: ";

                std::cin >> y;

                std::cout << "Enter ticket number: ";

                std::cin >> num;

                stud\_bilets.emplace\_back(y, num);

                break;

            }

            case 3: {

                std::cout << "\n--- Passports ---\n";

                for(const auto& p : passports) p.print\_info();

                std::cout << "\n--- Student Tickets ---\n";

                for(const auto& s : stud\_bilets) s.print\_info();

                break;

            }

            case 4: {

                std::cout << "Enter target year: ";

                int year;

                std::cin >> year;

                int count = 0;

                for(const auto& p : passports) {

                    if(p == year) count++;

                }

                for(const auto& s : stud\_bilets) {

                    if(s["year"] == year) count++;

                }

                std::cout << "Found " << count << " documents for year " << year << "\n";

                break;

            }

            case 6:

                return;

            default:

                std::cout << "Invalid option!\n";

        }

    }

}

int main() {

    show\_menu();

    return 0;

}

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

В программе было реализовано private-наследование: класс studbilet от doxbase. (class StudBilet : private DoxNamespace::Doxbase {...})

Были использованы такие технологии, как: перегрузка операторов, виртуальные функции, статические члены, пространство имен, массивы объектов, полиморфизм.

Статические члены: Doxbase: static int total\_count (общий счетчик документов), Passport: static int passport\_count, StudBilet: static int studbilet\_count.

Пространство имен: DoxNamespace для базового класса, PassportNamespace для производного класса, StudBiletNamespace для private-наследника.