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 №5**

**Тема: Проектирование классов с наследованием**

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 проектирования классов и разработки механизмов открытого и закрытого наследования в С++.

**Задание на практическое занятие:**

*Выберите один из вариантов задания.*

**Вариант А.**

**Задание 1.** Спроектировать структуру классов в соответствии с индивидуальным заданием (см. таблица 7). Наследование одного из производных классов спроектировать по типу **private**. *Вычисляемый показатель* реализовать для объектов производных классов в отдельной функции, не являющейся дружественной к классам.

**Задание 2.** Для каждого производного класса динамически выделить память для 5-элементного массива объектов. Свойства объектов задавать случайным образом. Определить *Вычисляемый показатель*.

**Задание 3.** Проект разместить в нескольких файлах.

**Текст программы:**

#include <iostream>

#include <cstdlib>

#include <ctime>

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

#include "../include/Studbilet.h"

#include "../include/func.h"

int main()

{

    srand(time(0));

    Alphabet();

    const int SIZE = 10;

    passport\* passports = new passport[SIZE];

    studBilet\* studBilets = new studBilet[SIZE];

    // Вычисляем показатель (документы, выпущенные в 2024 году)

    for (int i = 0; i < SIZE; ++i)

    {

        CountY(passports[i].getYear());

        std::cout << passports[i].getYear() << " " << passports[i].getName() << "\n";

        CountY(studBilets[i].getYear()); // Теперь есть доступ через public метод

        std::cout << studBilets[i].getYear() << " " << studBilets[i].getNumber() << "\n";

    }

    std::cout << "Number of documents issued in 2024: " << count << std::endl;

    delete[] passports;

    delete[] studBilets;

    delete[] alphabet;

    return 0;

}

//g++ main.cpp passport.cpp studbilet.cpp func.cpp -o main

}

#include "../include/func.h"

#include <cstdlib>

char\* alphabet = new char[26];

int count = 0;

void Alphabet()

{

    for (int i = 0; i < 26; ++i)

    {

        alphabet[i] = 'A' + i;

    }

}

void CountY(int x)

{

    if (x == 2024) count++; // Исправлено = на ==

}

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

#include <iostream>

#include <cstdlib>

extern char\* alphabet; // Объявление внешней переменной

passport::passport()

{

    setYear(2020 + rand()%6);

    name = alphabet[rand()%26];

    //std::cout << "Passport created\n";

}

passport::~passport()

{

    //std::cout << "Passport deleted\n";

}

#include "../include/Studbilet.h"

#include <iostream>

#include <cstdlib>

studBilet::studBilet()

{

    setYear(2000 + rand()%25);

    number = (rand() % 1024);

    //std::cout << "StudBilet created\n";

}

studBilet::~studBilet() {

    //std::cout << "StudBilet deleted\n";

}

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

В программе были использованы такие технологии, как: классы, наследуемые классы.

Были изучены взаимоотношения между членами классов и доступ к ним. На примере Doxbase.h и Passport.h – наследование public Doxbase (открытых полей). В больших проектах очень удобная технология.

Также были рассмотрены защищенные члены класса: private. У каждого класса есть свои данные, до которых можно добраться только через get, set.

Программа использует виртуальные функции – это такие функции, которые можно унаследовать, и под каждую задачу он будет реализован по-своему. Пример: virtual ~Doxbase() = default;