Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа**

**"** **Блоковый ввод-вывод "**

Выполнил:

студент группы ИВТ-23-1б

Машарипов.А

Проверила:

доцент кафедры ИТАС

Яруллин.Д.В

2024

1. **Постановка задачи.**

Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление и добавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляемых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

1. **Вариант задания**
2. Структура "Абитуриент":

* фамилия, имя, отчество;
* год рождения;
* оценки вступительных экзаменов (3);
* средний балл аттестата.

Удалить элемент с указанным номером, добавить элемент после элемента с указанной фамилией.

1. **Текст программы.**

#include <iostream>

#include <vector>

#include <string>

#include <iomanip>

struct Abiturient {

    std::string lastName;

    std::string firstName;

    std::string middleName;

    int birthYear;

    int examScores[3]; // Оценки вступительных экзаменов

    double averageScore; // Средний балл аттестата

    // Конструктор для удобства создания абитуриента

    Abiturient(const std::string& lastName, const std::string& firstName,

               const std::string& middleName, int birthYear,

               const int examScores[3], double averageScore)

        : lastName(lastName), firstName(firstName),

          middleName(middleName), birthYear(birthYear),

          averageScore(averageScore) {

        for (int i = 0; i < 3; ++i) {

            this->examScores[i] = examScores[i];

        }

    }

};

void addAbiturient(std::vector<Abiturient>& abiturients, const Abiturient& newAbiturient, const std::string& afterLastName) {

    for (size\_t i = 0; i < abiturients.size(); ++i) {

        if (abiturients[i].lastName == afterLastName) {

            abiturients.insert(abiturients.begin() + i + 1, newAbiturient);

            return;

        }

    }

    std::cout << "Абитуриент с фамилией " << afterLastName << " не найден.\n";

}

void removeAbiturient(std::vector<Abiturient>& abiturients, size\_t index) {

    if (index < abiturients.size()) {

        abiturients.erase(abiturients.begin() + index);

    } else {

        std::cout << "Индекс вне диапазона.\n";

    }

}

void printAbiturients(const std::vector<Abiturient>& abiturients) {

    for (const auto& abiturient : abiturients) {

        std::cout << "ФИО: " << abiturient.lastName << " " << abiturient.firstName << " " << abiturient.middleName

                  << ", Год рождения: " << abiturient.birthYear

                  << ", Оценки: [" << abiturient.examScores[0] << ", " << abiturient.examScores[1] << ", " << abiturient.examScores[2] << "]"

                  << ", Средний балл: " << std::fixed << std::setprecision(2) << abiturient.averageScore << "\n";

    }

}

int main() {

    std::vector<Abiturient> abiturients;

    // Добавляем несколько абитуриентов

    abiturients.emplace\_back("Иванов", "Иван", "Иванович", 2000, {85, 90, 78}, 4.5);

    abiturients.emplace\_back("Петров", "Петр", "Петрович", 1999, {80, 82, 88}, 4.3);

    abiturients.emplace\_back("Сидоров", "Сидор", "Сидорович", 2001, {90, 91, 92}, 4.8);

    std::cout << "Список абитуриентов:\n";

    printAbiturients(abiturients);

    // Удаляем абитуриента с индексом 1

    removeAbiturient(abiturients, 1);

    std::cout << "\nПосле удаления абитуриента с индексом 1:\n";

    printAbiturients(abiturients);

    // Добавляем нового абитуриента после Петрова

    Abiturient newAbit("Семенов", "Семен", "Семенович", 2002, {75, 80, 85}, 4.0);

    addAbiturient(abiturients, newAbit, "Петров");

    std::cout << "\nПосле добавления Семенова после Петрова:\n";

    printAbiturients(abiturients);

    return 0;

}

1. **Результат решения конкретного варианта.**