

МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение высшего образования*

***«МИРЭА – Российский технологический университет»***

**РТУ МИРЭА**

Отчет по выполнению практического задания № 4

**Тема**: **Стандартные типы данных для представления многоэлементных не однородных структур данных**

Дисциплина: Структуры и алгоритмы обработки данных

|  |  |
| --- | --- |
| Выполнил студент | Дмитриев П. В. |
|  | Фамилия И.О. |
| Группа | ИКБО-33-21 |
|  | Номер группы |

Москва 2022

Вариант № 7

ЗАДАНИЕ 1

Разработать набор операций для управления таблицей, созданной на основе статического массива.

Задача варианта

Учет кадров организации. Сотрудник: табельный номер (трехзначное число), фамилия и инициалы, код занимаемой должности, дата поступления на работу. Создать справочник по должностям – массив из 5 элементов, структура элемента: код (двухзначное число), название должности. Справочник заполните при инициализации.

Операции

1. Заполнить с клавиатуры запись по одному сотруднику.
2. Вставить в таблицу сведения по сотруднику так, чтобы запись разместилась после последней записи с такой же должностью.
3. Заменить у всех сотрудников код заданной должности на новый код (перевели на другую должность)
4. Удалить сотрудников, занимающих заданную должность.
5. **Разработка программы задачи 1**
   1. **Постановка задачи**

Дано. Дана структура, которую нужно заполнить данными сотрудников

Результат. Вывод таблицы отредактированной таблицы сведений по сотрудникам

**1.2 Описание модели решения**

Дан массив объектов структуры Employee, которые хранят сведения по сотрудникам, заполненные с клавиатуры.

Employee:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| int id | string fam | string name | string o | int pos | Date date |
| 100 | Иванов | И | И | 10 | 28 4 2022 |

Сотрудник:

* Табельный номер – id;
* Фамилия – fam;
* Инициалы – name, o;
* Код занимаемой должности – pos;
* Дата поступления на работу – date.

Сведения о дате поступления на работу записываются в объекте структуры Date.

Date:

|  |  |  |
| --- | --- | --- |
| int d | int m | int y |
| 28 | 4 | 2022 |

Сначала программа запрашивает количество сотрудников, после чего происходит ввод данных каждого из них с помощью функции input\_pers. После, функцией add\_pers запрашиваются данные еще одного сотрудника, запись которого должна быть расположена после последней записи сотрудника с такой же должностью, за это действие отвечает функция sort, в которой сортируются объекты структуры по возрастанию числа кода занимаемой должности с помощью сортировки выбором. Функция output\_pers выводит текущую отредактированную таблицу. Далее функция replace\_pos запрашивает данные о старом и новом кодах должностей, после это опять выводится промежуточный результат. В функцию del\_pers передается с клавиатуры код занимаемой должности, сотрудников которой нужно удалить. Эта функция выполняет задачу с помощью создания нового массива, в который передаются объекты, чей код не равен заданному коду.

**1.3 Декомпозиция:**

Определение прототипов функций:

1. Заполнение с клавиатуры данных о нескольких сотрудниках

Предусловие. \*a – указатель на массив emp объектов структуры Employee, x – количество элементов в массиве emp, причем x ≤ 100, где 100 – максимальное число элементов.

Постусловие. Заполненный массив emp.

void input\_pers(Employee \*a, int x)

1. Вывод данных о сотрудниках на экран

Предусловие. \*emp – указатель на массив emp объектов структуры Employee, x – количество элементов в массиве emp, причем x ≤ 100, где 100 – максимальное число элементов.

Постусловие. Вывод элементов массива emp.

void output\_pers(Employee \*emp, int x)

1. Сортировка элементов массива emp по возрастанию кода должности сотрудника

Предусловие. \*emp – указатель на массив emp объектов структуры Employee, x – количество элементов в массиве emp, причем x ≤ 100, где 100 – максимальное число элементов.

Постусловие. Отсортированный массив.

void sort(Employee \*emp, int x)

1. Добавление нового объекта в массив emp

Предусловие. \*a – указатель на массив emp объектов структуры Employee, x – количество элементов в массиве emp, причем x ≤ 100, где 100 – максимальное число элементов.

Постусловие. Массив с добавленным элементом.

void add\_pers(Employee \*a, int x)

1. Замена кода заданной должности у всех сотрудников на новый код

Предусловие. \*emp – указатель на массив emp объектов структуры Employee, x – количество элементов в массиве emp, причем x ≤ 100, где 100 – максимальное число элементов, old\_pos – код заданной должности, new\_pos – код новой должности.

Постусловие. Изменение свойств определенных объектов.

void replace\_pos(Employee \*emp, int x, int old\_pos, int new\_pos)

1. Удаление сотрудников, занимающих заданную должность

Предусловие. \*emp – указатель на массив emp объектов структуры Employee, \*emp2 – указатель на массив emp2 отобранных объектов, x – количество элементов в массиве emp, причем x ≤ 100, где 100 – максимальное число элементов, del\_pos – код заданной должности, sum – размер массива emp2.

Постусловие. Создание нового массива.

void del\_pers(Employee \*emp, Employee\* emp2, int x, int del\_pos, int sum)

**1.4 Код программы:**

Код из файла 4\_static.cpp:

#include "4.h"

const int N = 100;

struct Position {

int code;

string name;

};

int main() {

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Position pos[5];

pos[0].code = 10;

pos[0].name = "Директор";

pos[1].code = 11;

pos[1].name = "Стажер";

pos[2].code = 12;

pos[2].name = "Менеджер";

pos[3].code = 13;

pos[3].name = "Аналитик";

pos[4].code = 14;

pos[4].name = "Юрист";

Employee emp[N];

Employee emp2[N];

int x;

cout << "Введите количество сотрудников: ";

cin >> x;

cout << "\n";

input\_pers(emp, x);

add\_pers(emp, x);

sort(emp, x);

output\_pers(emp, x);

int old\_pos, new\_pos;

cout << "Введите код старой и новой должности: "; cin >> old\_pos >> new\_pos;

replace\_pos(emp, x, old\_pos, new\_pos);

output\_pers(emp, x);

int del\_pos, size\_emp2 = 0;

cout << "Введите код должности, сотрудников которой нужно удалить (от 10 до 14): "; cin >> del\_pos;

del\_pers(emp, emp2, x, del\_pos, size\_emp2);

output\_pers(emp2, size\_emp2 + 1);

}

Код из файла 4.h:

#ifndef \_4\_H

#define \_4\_H

#include <iostream>

#include <string>

#include <clocale>

#include <Windows.h>

using namespace std;

struct Date {

int d, m, y;

};

struct Employee {

int id;

string fam;

string name;

string o;

int pos;

Date date;

};

void input\_pers(Employee \*a, int x) {

for (int i = 0; i < x; i++) {

cout << "Введите данные " << i + 1 << "-го сотрудника:" << "\n";

cout << "Табельный номер: "; cin >> a[i].id;

cout << "Фамилия и инициалы: "; cin >> a[i].fam >> a[i].name >> a[i].o;

cout << "Код занимаемой должности (от 10 до 14): "; cin >> a[i].pos;

cout << "Дата поступления на работу (в формате: день мессяц год): "; cin >> a[i].date.d >> a[i].date.m >> a[i].date.y;

cout << "\n";

}

}

void output\_pers(Employee \*emp, int x) {

for (int i = 0; i < x + 1; i++) {

cout << "Фамилия и инициалы: " << emp[i].fam << " " << emp[i].name << " " << emp[i].o << "\n";

cout << "Табельный номер: " << emp[i].id << "\n";

cout << "Код занимаемой должности: " << emp[i].pos << "\n";

cout << "Дата поступления на работу: " << emp[i].date.d << "." << emp[i].date.m << "." << emp[i].date.y;

cout << "\n";

}

}

void sort(Employee \*emp, int x) {

cout << "\n";

for (int i = 0; i < x; i++) {

int min = i;

for (int j = i + 1; j < x + 1; j++) {

if (emp[j].pos < emp[min].pos) {

min = j;

}

}

if (min != i) {

swap(emp[i], emp[min]);

}

}

}

void add\_pers(Employee \*a, int x) {

cout << "Введите данные сотрудника:" << "\n";

cout << "Табельный номер: "; cin >> a[x].id;

cout << "Фамилия и инициалы: "; cin >> a[x].fam >> a[x].name >> a[x].o;

cout << "Код занимаемой должности (от 10 до 14): "; cin >> a[x].pos;

cout << "Дата поступления на работу (в формате: день мессяц год): "; cin >> a[x].date.d >> a[x].date.m >> a[x].date.y;

cout << "\n";

cout << "Размещение записи после последней с такой же должностью" << "\n";

}

void replace\_pos(Employee \*emp, int x, int old\_pos, int new\_pos) {

for (int i = 0; i < x + 1; i++) {

if (emp[i].pos == old\_pos) {

emp[i].pos = new\_pos;

}

}

}

void del\_pers(Employee \*emp, Employee\* emp2, int x, int del\_pos, int sum) {

for (int i = 0; i < x + 1; i++) {

if (emp[i].pos != del\_pos) {

emp2[sum] = emp[i];

sum++;

}

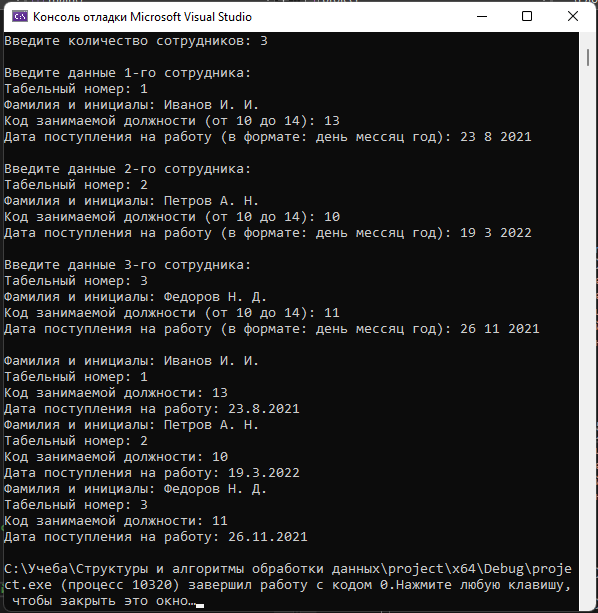
}

}

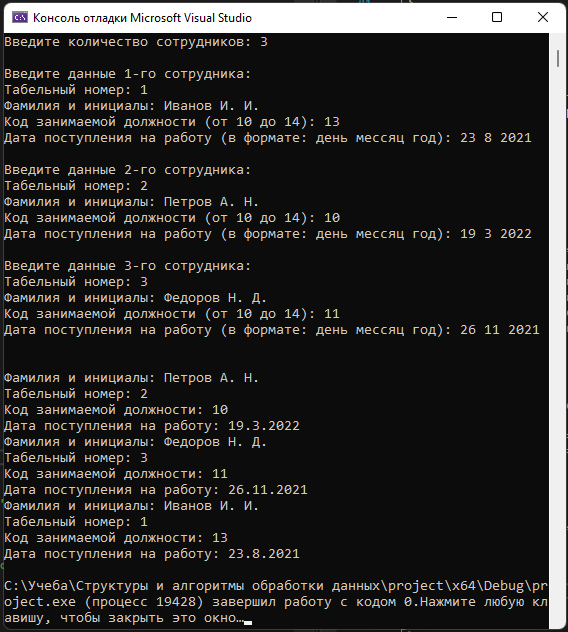
#endif

* 1. **Скриншоты:**

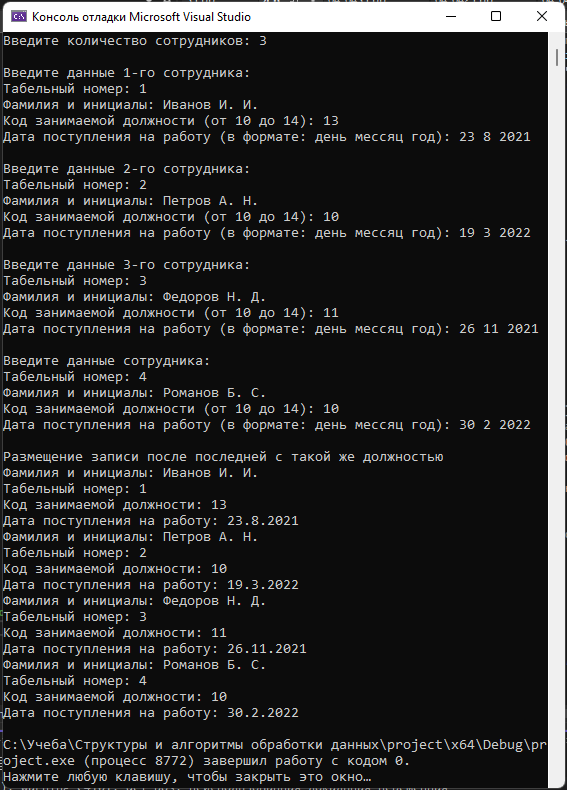
1. Тестирование функции декомпозиции: вывод данных о сотрудниках на экран



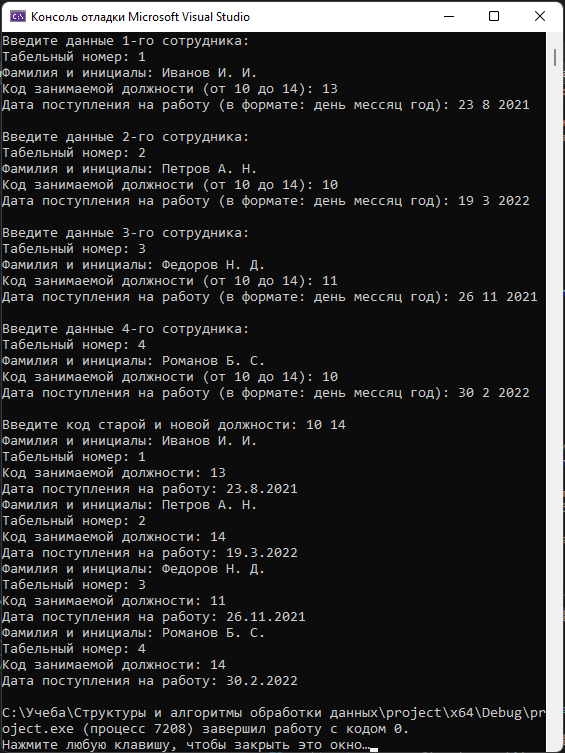
1. Тестирование функции декомпозиции: сортировка элементов массива emp по возрастанию кода должности сотрудника



1. Тестирование функции декомпозиции: добавление нового объекта в массив emp



1. Тестирование функции декомпозиции: замена кода заданной должности у всех сотрудников на новый код



1. Тестирование функции декомпозиции: удаление сотрудников, занимающих заданную должность

