Учреждение образования

«БЕЛОРУССКИЙГОСУДАРСТВЕННЫЙТЕХНОЛОГИЧЕСКИЙУНИВЕРСИТЕТ»

Н. Н. Пустовалова, Н. В. Пацей

# ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

**в 2-х частях**

**Часть 2**

**Лабораторный практикум**

**Лабораторная работа № 5. Объединения, перечисления, битовые поля**

**Лабораторная работа № 5. Объединения, перечисления, битовые поля**

|  |  |
| --- | --- |
| **Задание** | **Программа** |
| 1. Изучить принципы работы с *объединениями*, выполнив программу, записанную в данном пункте. |  |
| 2. Изучить принципы работы с *перечислениями*, выполнив программу, записанную в данном пункте. |  |
| 3. Изучить принципы работы с *битовыми полями*, выполнив программу, записанную в данном пункте. |  |
| 4. В соответствии со своим вариантом разработать программу с использованием ***перечислений*** и ***битовых*** *полей* для работы с данными из таблицы, приведенной ниже. Реализовать функции ввода с клавиатуры, вывода на экран, удаления, поиска элементов. Интерфейс пользователя осуществить в виде меню.  **Вар.1:**  **Отдел кадров**. Ф.И.О. работника, образование, специальность, подразделение, должность, оклад, дата поступления на предприятие. Выбор по стражу работы. Даты реализовать с помощью битового поля, должность − с помощью перечисления. |  |

#include <iostream>

#include <windows.h>

using namespace std;

FILE\* fnewdata;

FILE\* fdata;

struct date

{

unsigned int day : 5;

unsigned int mouth : 4;

unsigned int year : 11;

};

union job

{

char title[100];

};

struct worker

{

int id;

char fio[100];

char education[100];

char speciality[100];

char subdivision[100];

job jobTitle;

int salary;

date dateReceipt;

};

int getid()

{

worker buf;

int newid = 0;

if (!fopen\_s(&fdata, "data.bin", "rb"))

{

while (fread(&buf, sizeof(buf), 1, fdata))

{

if (newid < buf.id)

{

newid = buf.id;

}

}

fclose(fdata);

}

else // файл не получилось открыть для чтения, значит его не сущестует, сотрудников нет newid=1;

{

newid = 0;

}

return newid + 1;

}

void printtab(int num)

{

for (int i = 0; i < num; i++)

{

cout << "\t";

}

}

void printline()

{

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

}

void output()

{

worker buf;

cout << "id |" << "\tФио сотрудника\t\t|" << "Образование\t|" << "\tСпециальность\t|";

cout << "\tПодразделение\t|" << "\tДолжность\t|" << "Оклад\t|" << " Дата поступления на работу\t|\n";

printline();

if (!fopen\_s(&fdata, "data.bin", "rb"))

{

while (fread(&buf, sizeof(buf), 1, fdata))

{

cout << buf.id << " |" << buf.fio << "\r";

printtab(4);

cout << "| " << buf.education << "\r";

printtab(6);

cout << "| " << buf.speciality << "\r";

printtab(9);

cout << "| " << buf.subdivision << "\r";

printtab(12);

cout << "| ";

cout << buf.jobTitle.title << "\r";

printtab(15);

cout << "| " << buf.salary << "\t|\t" << buf.dateReceipt.day <<"."<<buf.dateReceipt.mouth<<"."<<buf.dateReceipt.year << "\t\t|\n";

printline();

}

fclose(fdata);

}

else

{

cout << "Ошибка открытия файла\n";

}

}

void input()

{

int date;

int size;

worker buf;

cout << "Введите сколько новых сотрудников хотите записать: ";

cin >> size;

for (int i = 0; i < size; i++)

{

buf.id = getid();

fopen\_s(&fdata, "data.bin", "ab");

cout << "Введите фио сотрудника: ";

cin.ignore();

cin.getline(buf.fio, 100);

cout << "Введите образование сотрудника: ";

cin.getline(buf.education, 100);

cout << "Введите специальность сотрудника: ";

cin.getline(buf.speciality, 100);

cout << "Введите подразделение сотрудника: ";

cin.getline(buf.subdivision, 100);

cout << "Введите должность сотрудника: ";

cin.getline(buf.jobTitle.title, 100);

cout << "Введите оклад сотрудника: ";

cin >> buf.salary;

cout << "Введите год поступления на работу: ";

cin >> date;

buf.dateReceipt.year = date;

cout << "Введите месяц поступления на работу: ";

cin >> date;

buf.dateReceipt.mouth = date;

cout << "Введите день поступления на работу: ";

cin >> date;

buf.dateReceipt.day = date;

fwrite(&buf, sizeof(buf), 1, fdata);

fclose(fdata);

}

}

void deleteworker()

{

worker buf;

int index;

cout << "Введите id работника которого хотите уволить: ";

cin >> index;

if (!fopen\_s(&fdata, "data.bin", "rb"))

{

while (fread(&buf, sizeof(buf), 1, fdata))

{

if (index == buf.id)

{

continue;

}

fopen\_s(&fnewdata, "newdata.bin", "ab");

fwrite(&buf, sizeof(buf), 1, fnewdata);

fclose(fnewdata);

}

fclose(fdata);

remove("data.bin");

rename("newdata.bin", "data.bin");

}

else

{

cout << "Ошибка открытия файла\n";

}

}

void searchworker()

{

worker buf;

int year;

cout << "Введите стаж работы: ";

cin >> year;

cout << "Информация о сотрудниках, чей стаж работы превышает " << year << " лет:\n";

if (!fopen\_s(&fdata, "data.bin", "rb"))

{

while (fread(&buf, sizeof(buf), 1, fdata))

{

if (year < 2025 - buf.dateReceipt.year)

{

printline();

cout << "Фио сотрудн: " << buf.fio << "\nСтаж работы: " << 2025 - buf.dateReceipt.year << "\nОбразование: ";

cout << buf.education << "\nДолжность: " << buf.jobTitle.title << "\nОклад: " << buf.salary << "\nСпециальность: " << buf.speciality;

cout << "\nПодразделение: " << buf.subdivision << "\nid для взаимодействия: " << buf.id << "\n";

printline();

}

}

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int userInput;

setlocale(LC\_CTYPE, "Russian");

cout << "Меню:\n" << "1 - Ввод данных о новом сотруднике\n" << "2 - Вывод информации о всех сотрудниках\n";

cout << "3 - Увольнение сотрудника\n" << "4 - Поиск сотрудника по стажу работы\n" << "0 - Выход из программы\n";

do

{

cin >> userInput;

switch (userInput)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

deleteworker();

break;

case 4:

searchworker();

break;

case 0:

break;

default:

cout << "Ошибка выбора\n";

break;

}

cout << "Что дальше? \n";

} while (userInput != 0);

}

|  |  |
| --- | --- |
| 5. В соответствии со своим вариантом разработать программу с использованием структуры в виде ***объединения****,* для работы с данными из таблицы, приведенной ниже. Реализовать функции ввода с клавиатуры, записи в файл, вывода на экран, чтения из файла и поиска.  **Вар.1: Преподаватели**. Фамилия преподавателя, название экзамена, дата экзамена. Выбор по фамилии. |  |

#include <iostream>

#include <windows.h>

#include <string>

using namespace std;

FILE\* fdata;

union teacher

{

char surname[100];

char exam[100];

char date[11];

};

struct teacheroutput

{

char surname[100];

char exam[100];

char date[11];

};

void inputTeacher()

{

teacher buf;

int size;

cout << "Введите сколько новых преподавателей хотите добавить: ";

cin >> size;

cin.ignore();

if (!fopen\_s(&fdata, "data.bin", "ab"))

{

for (int i = 0; i < size; i++)

{

cout << "Введите фамилию преподавателя: ";

cin.getline(buf.surname, 100);

fwrite(&buf, sizeof(buf.surname), 1, fdata);

cout << "Введите название экзамена: ";

cin.getline(buf.exam, 100);

fwrite(&buf, sizeof(buf.exam), 1, fdata);

cout << "Введите дату проведения экзамена (в формате 09.08.2025): ";

cin.getline(buf.date, 11);

fwrite(&buf, sizeof(buf.date), 1, fdata);

}

fclose(fdata);

}

else

{

cout << "Ошибка открытия файла: ";

}

}

void printtab(int size)

{

for (int i = 0; i < size; i++)

{

cout << "\t";

}

}

void printline()

{

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

}

void showTeacher()

{

teacheroutput buf;

cout << "|\tПреподаватель\t|" << "\tЭкзамен\t\t|" << "Дата проведения экзамена|\n";

printline();

if (!fopen\_s(&fdata, "data.bin", "rb"))

{

while (fread(&buf, sizeof(buf), 1, fdata))

{

cout << "|" << buf.surname << "\r";

printtab(3);

cout << " | " << buf.exam << "\r";

printtab(6);

cout << "| " << buf.date << "\r";

printtab(9);

cout << " | \n";

printline();

}

fclose(fdata);

}

else

{

cout << "Ошибка открытия файла\n";

}

}

void searchTeacher()

{

teacheroutput buf;

string surname, searchSurname;

bool resultOfsearch = 0;

cout << "Введите фамилию преподавателя: ";

cin >> surname;

if (!fopen\_s(&fdata, "data.bin", "rb"))

{

while (fread(&buf, sizeof(buf), 1, fdata))

{

searchSurname = string(buf.surname);

if (surname == searchSurname)

{

resultOfsearch = true;

printline();

cout << "Фамилия преподавателя: " << buf.surname;

cout << "\nЭкзамен: " << buf.exam;

cout << "\nДата проведения экзамена: " << buf.date << "\n";

printline();

}

}

fclose(fdata);

}

else

{

cout << "Ошибка открытия файла\n";

}

if (!resultOfsearch)

{

cout << "Преподавателей с заданой фамилией не найдено.\n";

}

}

int main()

{

int userInput;

setlocale(LC\_CTYPE, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Введите действие:\n" << "1 - Добавить нового преподавателя/экзамен\n";

cout << "2 - Вывести список преподавателей\n" << "3 - Поиск преподавателя по фамилии\n" << "0 - Выход из программы\n";

do

{

cin >> userInput;

switch (userInput)

{

case 1:

inputTeacher();

break;

case 2:

showTeacher();

break;

case 3:

searchTeacher();

break;

case 0:

break;

default:

cout << "Ошибка выбора\n";

break;

}

cout << "Что дальше?\n";

} while (userInput != 0);

}