

Лабораторная работа № 14

Модульное программирование на примере языка программирования C++ Структуры. Вектор структур

Учебная задача 1. Описать структуру *SiteUser* (*Пользователь сайта*). Сформировать массив структур с клавиатуры, размерность массива вводится с клавиатуры. Вывести всех пользователей сайта на экран, в бинарный файл. Разбить задачу на блоки – отдельные модули, реализовать решение в виде модулей.

Описание структуры:

```
struct SiteUser
{
    string login, password, status;
    int visits;
};
```

Обращение к полю записи через точку:

```
SiteUser user1;
cout<< user1.login;
cout<< user1.password;
cout<< user1.status;
cout<< user1.visits;
```

Модульность в языке C++

Модуль логически состоит из двух файлов:

заголовочного файла (header file) – *Site.h*, включает прототипы функций, объявление переменных и констант, о которых должны знать остальные модули, подключение необходимых для реализации заголовочных файлов

```
#ifndef _SITE_H
#define _SITE_H
...
#endif // _SITE_H
```

файла с исходным кодом (source file) – *Site.cpp*, содержит определение функций.

Пример:

Файл *Site.h* содержит функции работы с меню в программе

Основы программирования

```
#ifndef _SITE_H
#define _SITE_H

#include <string>
using namespace std;

struct SiteUser
{
    string login, password, status;
    int visits;
};

int initconsol(SiteUser *, int); //формирование из консоли массива структур
void printconsol(SiteUser *, int); //вывод в консоль
void printbinfile(SiteUser *, int); //вывод в бинарный файл
ostream &operator<<(std::ostream &, SiteUser &); //перегрузка оператора вывода
структуры

#endif _SITE_H
```

Файл *Site.cpp* содержит функции работы с меню в программе

```
#include "stdafx.h"
#include <iostream>
#include <fstream>

int initconsol(SiteUser *a, int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << "Input " << i << " element\n ";
        cout << "Login ";
        cin >> a[i].login;
        cout << "Password ";
        cin >> a[i].password;
        cout << "Status ";
        cin >> a[i].status;
        cout << "Visits ";
        cin >> a[i].visits;
    }
    return n;
}

ostream &operator<<(std::ostream & out, SiteUser & user)
{
    out << user.login << " " << user.password << " " << user.status << " " << user.visits
    << ")\n";
    return out;
}

void printconsol(SiteUser * a, int n)
{
    cout<< "-----\n";
    for (int i = 0; i < n; i++)
        cout <<a[i];
    //cout << "-----";
}
```

```
void printbinfile(SiteUser * a, int n)
{
    char namefile[15];
    cout << "Input filename ";
    cin>>namefile;
    ofstream outfile(namefile, ios::binary | ios::out);
    if (outfile)
    {
        for (int i=0; i<n; ++i) //запись в бинарный файл структуры
            outfile.write(reinterpret_cast<char*> (a + i), sizeof(SiteUser));
    }
    else cerr << "File not found";
    outfile.close();}
```

Можно записать в бинарный файл сразу весь массив размерности n

```
//запись в бинарный файл всего массива структур
outfile.write(reinterpret_cast<char*> (a), sizeof(SiteUser)*n);
```

Учебная задача 2. Описать вектор из структур *SiteUser*. Сформировать вектор из файла (получившегося в задаче 1). Обеспечить добавление новой записи – с клавиатуры. Вывести всех пользователей сайта на экран. Разбить задачу на блоки – отдельные модули, реализовать решение в виде модулей.

В заголовочном файле

```
#ifndef _SITE_H
#define _SITE_H

#include <string>
#include <vector>
using namespace std;

struct SiteUser
{
    string login, password, status;
    int visits;
};

void initfromfile(vector<SiteUser> &); //формирование вектора структур из бинарного файла
void printconsol(vector<SiteUser> &); //вывод вектора структур в консоль
void add(vector<SiteUser> &); //добавление новой структуры в вектор
istream &operator >> (std::istream &, SiteUser &); //перегрузка оператора ввода структуры

#endif _SITE_H
```

В описательном файле

```
#include "stdafx.h"
#include <iostream>
#include <fstream>
```

```
using namespace std;

istream &operator >> (std::istream &in , SiteUser &user)
{
    cout << "Login ";
    in >> user.login;
    cout << "Password ";
    in >> user.password;
    cout << "Status ";
    in >> user.status;
    cout << "Visits ";
    in >> user.visits;
    return in;
}

void add(vector<SiteUser> &a)
{
    SiteUser x;
    cin >> x;
    a.push_back(x);
}

void printconsol(vector<SiteUser> &a)
{
    //проход по вектору с помощью итератора
    for (vector<SiteUser>::iterator it = a.begin(); it < a.end(); it++)
        cout << *(it);
}

void initfromfile(vector<SiteUser> &a)
{
    char namefile[15];
    cout << "Input filename ";
    cin >> namefile;
    ifstream infile(namefile, ios::binary | ios::in);
    if (infile.is_open())
    {
        SiteUser *x = new SiteUser;
        //считывание из бинарного файла записей
        while (infile.read((char*)(x), sizeof(SiteUser)))
        {
            //добавление записи в вектор
            a.push_back(*x);
        }
    }
    else cerr << "File not found";
    infile.close();
}
```

Задание на лабораторную работу

Задание выполняется по вариантам. Номер варианта получить у преподавателя.

1 вариант. Создать запись «Участник» для таблицы конференции. Поля записи: ФИО, страна, город, место работы, возраст и телефон.

Упорядочить список участников конференции по фамилии в алфавитном порядке. Сделать выборку иностранных участников конференции. Найти самого молодого русского участника конференции. Проверить, есть ли в списке участник из ВятГУ.

2 вариант. Создать запись «Сотрудник» для таблицы фирмы. Поля записи: ФИО, № паспорта, место прописки, возраст, пол, семейное положение и телефон. Упорядочить список сотрудников по номеру паспорта. Сделать выборку работников, которым до пенсии осталось не более 5 лет. Проверить, работают ли в фирме женщины, если работают, то найти самую молодую незамужнюю сотрудницу.

3 вариант. Создать запись «Авто» для таблицы автосалона. Поля записи: марка, модель, тип кузова, год выпуска, пробег, объем двигателя, коробка передач. Упорядочить список автомобилей по объему двигателя по убыванию. Сделать выборку автомобилей по типу кузова. Найти автомобиль с наибольшим пробегом с коробкой автомат. Проверить, есть ли в списке автомобили определенной марки.

4 вариант. Создать запись «Сотрудник» для таблицы фирмы. Поля записи: ФИО, фамилия, имя, отчество, дата поступления на работу (месяц, год), образование, телефон. Упорядочить список сотрудников по алфавиту. Вывести список тех, кто проработал 5 полных лет и имеет среднее образование. Найти сотрудника, у которого максимальный стаж работы и высшее образование.

5 вариант. Создать запись «Авто» для таблицы заводского гаража: марка, номер, цвет, год выпуска, пробег, фамилия шофера, номер машиноместа. Упорядочить список автомобилей по пробегу. Вывести список машин, подготовленных на списание (старше заданного года выпуска или с пробегом больше заданной величины) и их количество. Проверить, есть ли автомобиль заданной марки и цвета ранее 2015 года выпуска.

Порядок выполнения работы

1. Создать файл `struct.h` для описания в нем записи и перегрузки операторов вывода << и ввода >>.
2. Заполните массив (не менее 8 элементов в массиве) из структур с клавиатуры. Запишите массив в бинарный файл. Бинарный файл заполнить 1 раз (для этого нужна вспомогательная программа с пользовательским интерфейсом). Для работы с массивом структур создать файлы заголовочный и описательный.

3. Заполнить вектор из структур из бинарного файла. Для работы с вектором структур создать **заголовочный и описательный** файлы.
4. Начальные данные для формирования вектора брать только из **бинарного файла**.
5. Все задания по выборке и упорядочиванию элементов выполнить для вектора из структур. Вывод результатов выборки и упорядочивания выводить в текстовые файлы. Дублировать вывод на экран.
6. Организовать добавление элементов в вектор. Результирующий вектор перезаписывать в бинарный файл.
7. Программа должна иметь пользовательский интерфейс с возможностью выбора действий.