Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №9**

Дисциплина: «Информатика»

Тема: Структуры данных

Вариант 20

Выполнила:

Студент группы ИВТ-20-1б

Морохин Александр Олегович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель:** работа с двоичными файлами, организация ввода-выводы структурированной информации и ее хранение на внешних носителях.

**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.

**Вариант**

Структура «Государство»:

* Название;
* Государственный язык;
* Денежная единица;
* Курс валюты относительно $.

Удалить элемент с указаным названием, добавить 2 элемента в конец файла

**Анализ задачи**

1. Для заполнения файла нужно использовать функцию, формирующую одну структуру, указанного в варианте типа. Значения элементов структуры вводятся с клавиатуры. Для ввода необходимо использовать операцию >> и функция gets ().
2. При вводе структур реализовать один из следующих механизмов:

* Ввод заранее выбранного количества структур (не менее 5);
* Ввод до появления структуры с заданным количеством признаков;
* Диалог с пользователем о необходимости продолжать ввод.

1. Для записи структуры в файл и чтения структуры из файла использовать функции блочного ввода/вывода fread и fwrite.
2. Для удаления/добавления элементов в файл использовать вспомогательный файл.

**Код программы**

#include <iostream>  
  
using namespace std;  
  
struct State {  
 string title; // название страны  
 string lang; // язык  
 string money; // Как называется валюта  
 float course; // Курс  
};  
  
struct List {  
 State data; // Элемент списка  
 List\* next; // Указатель на следующий элемент  
};  
  
// Чтение и сохранение  
List\* readFile();  
void saveFile(List\* first);  
  
// Создание  
List\* make();  
  
// Удаление и добавление  
void del(List\* &first);  
void add(List\* &first, int count);  
  
// Вывод  
void print(List\* first);  
  
int main() {  
 setlocale(LC\_ALL, "Russian");  
 // List\* list = readFile();  
 // return 0;  
 List\* list = make();  
 cout << "Сформирован файл data.dat" << endl;  
 saveFile(list);  
  
 cout << "Содержание файла" << endl;  
 List\* list\_file = readFile();  
 print(list\_file);  
  
 cout << "Актуальный список" << endl;  
 print(list);  
 cout << endl;  
  
 del(list);  
  
 cout << "Актуальный список" << endl;  
 print(list);  
 cout << endl;  
  
 add(list, 2);  
  
 cout << "Актуальный список" << endl;  
 print(list);  
 cout << endl;  
  
 cout << "Файл data.dat обновлен" << endl;  
 saveFile(list);  
  
 return 0;  
}  
  
List\* readFile() {  
 List\* first, \*p;  
 first = NULL; // Пока нет элементов, указываем на нул  
  
 FILE \*fp;  
 if((fp = fopen("data.dat", "rb")) == NULL) {  
 perror("\nОшибка при открытии файла data.dat");  
 exit(0);  
 }  
  
 p = new List;  
 if(!feof(fp)) {  
 fread(&p, sizeof(p), 1, fp);  
 }  
 first = p;  
 fclose(fp);  
 return first;  
}  
  
void saveFile(List\* first) {  
 if(first == NULL) {  
 cout << "Список пуст" << endl;  
 } else {  
 List\* p = first;  
  
 FILE \*fp;  
 if((fp = fopen("data.dat", "wb")) == NULL) {  
 perror("\nОшибка при открытии файла data.dat");  
 exit(0);  
 }  
 rewind(fp);  
  
 while (!feof(fp) && p != NULL) {  
 fwrite(&p, sizeof(\*p), 1, fp);  
 p = p->next;  
 }  
 fclose(fp);  
 }  
}  
  
List\* make() {  
 List\* first, \*p;  
 first = NULL; // Пока нет элементов, указываем на нул  
  
 p = new List; // Память под элемент  
  
 // Инициализация первого элемента  
 p->data.title = "Россия";  
 p->data.lang = "Русский";  
 p->data.money = "Рубль";  
 p->data.course = 72.4;  
 p->next = NULL; // Это последний элемент  
  
 first = p;  
  
 List\* h = new List; // Память под новый элемент  
 p->next = h;  
 p = p->next; // Перешли к управлению над след элементом, так сказать :)  
 p->data.title = "Украина";  
 p->data.lang = "Украинский";  
 p->data.money = "Гривна";  
 p->data.course = 18.4;  
 p->next = NULL; // Это последний элемент  
 return first;  
}  
  
void del(List\* &first) {  
 if(first == NULL) {  
 cout << "Список пуст" << endl;  
 } else {  
 List\* p = first;  
 string title;  
 cout << "Введите государство, которое будет удалено:" << endl;  
 cin >> title;  
 while(p->data.title == title) {  
 List\* q = p->next;  
 first = p->next;  
 delete q;  
 p = first;  
 cout << "Найден элемент" << endl;  
 }  
 while(p->next != NULL) {  
 if(p->next->data.title == title) {  
 List\* q = p->next;  
 p->next = p->next->next; // Делаем "прыжок" через одного  
 delete q;  
 } else  
 p = p->next;  
 }  
 }  
}  
  
void add(List\* &first, int count) {  
 if (first == NULL) {  
 cout << "Список пуст" << endl;  
 } else {  
 cout << "Добавление элементов" << endl;  
 List\* p = first;  
 int size = 0;  
 while (p->next != NULL) {  
 p = p->next;  
 size++;  
 }  
 p = first;  
 for(int i = 1; i < size; i++) {  
 p = p->next;  
 }  
  
 for (int i = 1; i <= count; i++) {  
 List\* h = new List;  
 List\* q = p;  
 p->next = h;  
 p = p->next;  
  
 cout << "Введите название страны: ";  
 cin >> p->data.title;  
  
 cout << "Введите язык: ";  
 cin >> p->data.lang;  
  
 cout << "Введите денежную еденицу: ";  
 cin >> p->data.money;  
  
 cout << "Введите курс USD: ";  
 cin >> p->data.course;  
  
 p->next = NULL;  
 }  
 }  
}  
  
void print(List\* first) {  
 if(first == NULL) {  
 cout << "Список пуст" << endl;  
 } else {  
 List\* p = first;  
 while (p != NULL) {  
 cout << "Страна: " << p->data.title << " | ";  
 cout << "Язык: " << p->data.lang << " | ";  
 cout << "Валюта: " << p->data.money << " | ";  
 cout << "Курс USD: " << p->data.course << endl;  
 p = p->next;  
 }  
 }  
}

**Результаты работы программы**

