Тема: Структуры

Цель работы: Научиться работать со структурами данных, укрепить навыки работы с функциями.

Программное обеспечение: ОС Windows, СП С++ Builder

Ход работы

1. Объявить структуру данных и две переменных такого типа. Записать значения в переменные, вывести их на экран.
2. Объявить объединение (union), разобраться, чем отличается структура от объединения.
3. Объявить объединение из двух структур и структуру из двух объединений. Заполнить значениями, вывести на экран.
4. Объявить массив из структур. Записать значения в массив с помощью цикла, вывести на экран.
5. Написать функцию вывода структуры на экран. Переделать программу, полученную в п.4, с использованием этой функции.
6. Написать функцию, выводящую весь массив структур, используя функцию из п.5 в цикле.
7. Написать функцию заполнения структуры случайными данными.
8. Написать функцию заполнения всего массива структур, использующую функцию из п.7 в цикле.
9. Написать функцию перестановки данных в двух структурах, передаваемых по ссылке. Проверить ее работоспособность.
10. Отсортировать массив из структур по одному из полей структуры любым способом.

/\*

Лабораторная работа №5

СТРУКТУРЫ

Выполнил студент группы РПЗ-11 1/9

Кобусь И. Л.

30.01.14

\*/

#include <vcl.h>

#include <iostream.h>

#include <stdio.h>

#include <conio.h>

#pragma hdrstop

//---------------------------------------------------------------------------

#pragma argsused

// объявление структур

struct test\_st1{

int x; char c;

};

struct test\_st2{

float x; char s;

};

// объявление объединений

union test\_un1{

double b; int x;

};

union test\_un2{

long double a; int x; char c;

};

// объявление структур и объединений, состоящих из других структур/объединений

struct st\_big{

test\_un1 u1;

test\_un2 u2;

};

union un\_big{

test\_st1 t1;

test\_st2 t2;

};

// прототипы используемых в программе функций

void input\_struct(int& k, test\_st2\* t); // ручное заполнение массива структур

void output\_struct(test\_st2\* t); // вывод структуры

void output\_all\_structs(int& k, test\_st2\* t); // вывод всего массива структур

void input\_random(int& k, test\_st2\* t); // заполнение массива структур случайными числами

void sort(int& k, test\_st2\* t); // сортировка по возростанию величины поля float x

void recombination(test\_st2& t1, test\_st2& t2); // перестановка двух структур местами

int main(int argc, char\* argv[])

{

test\_st1 a[2]; int k;

for(int i = 1; i >= 0; i--){

printf("Vvedite zeloe chyslo i 1 simvol (ostalos' %d raz): ", i + 1);

cin >> a[i].x >> a[i].c;

}

cout << "\nVvedennie Vami dannie: \n";

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

cout << a[i].x << " " << a[i].c << "\n";

cout << "Vvedite kolichestvo struktur v massive (ne men'she 2): ";

cin >> k;

test\_st2\* t = new test\_st2[k]; // выделение памяти для массива структур

input\_random(k, t);

cout << "\nMassiv byl zapolnen sluchaynim obrazom... \n";

// input\_struct(k, t);

output\_struct(t);

output\_all\_structs(k, t);

sort(k, t);

cout << "\nPosle sortirovky: \n";

output\_all\_structs(k, t);

int x1, x2;

cout << "Vvedite nоmera structur dlya obmena mestami: ";

cin >> x1 >> x2;

recombination(t[x1 - 1], t[x2 - 1]);

cout << "\nPosle perestanovki: \n";

output\_all\_structs(k, t);

system("pause");

return 0;

};

void input\_struct(int& k, test\_st2\* t){

for(int i = k - 1; i >= 0; i++) {

printf("Vvedite veshestvennoe 1 chyslo i 1 simvol (ostalos' %d raz):\n", k);

scanf("%f %s", &t[i].x, &t[i].s);

}

};

void output\_struct(test\_st2\* t){

int k;

cout << "\nVVedite nomer structury dl'ya vivoda: ";

cin >> k;

cout << t[k - 1].x << " " << &t[k - 1].s << "\n";

};

void output\_all\_structs(int& k, test\_st2\* t){

cout << "\nZnacheniya vo vseh vvedinnyh structurah: \n";

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

cout << i + 1 << ". " << t[i].x << " " << &t[i].s << "\n";

};

void input\_random(int& k, test\_st2\* t){

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

t[i].x = random(1000)/10;

t[i].s = (char) random(100) + 1;

}

};

void sort(int& k, test\_st2\* t){

test\_st2 temp;

for(int i = 0; i < k - 1; i++) // пузырьковая сортировка

for(int z = i + 1; z < k; z++)

if(t[i].x > t[z].x){

temp.x = t[i].x;

t[i].x = t[z].x;

t[z].x = temp.x;

}

};

void recombination(test\_st2& t1, test\_st2& t2){

test\_st2 temp;

temp = t1;

t1 = t2;

t2 = temp;

};

Вывод: на данной лаборатрной работе я научился использовать средства ввода-вывода С++ при программировании

Тема: Файлы

Цель работы: Изучение основных приемов и алгоритмов, используемых при обмене информацией с внешней памятью, а также особенностей их программной реализации на зыке Си.

Программное обеспечение: ОС Windows, СП С++ Builder

/\*

Лабораторная работа №6

ФАЙЛЫ

Выполнил студент группы РПЗ-11 1/9

Кобусь И. Л.

30.01.14

\*/

#include <vcl.h>

#include <iostream.h>

#include <stdio.h>

#include <conio.h>

#pragma hdrstop

//---------------------------------------------------------------------------

#pragma argsused

// объявление структур

struct test\_st1{

int x; char c;

};

struct test\_st2{

float x; char s;

};

// объявление объединений

union test\_un1{

double b; int x;

};

union test\_un2{

long double a; int x; char c;

};

// объявление структур и объединений, состоящих из других структур/объединений

struct st\_big{

test\_un1 u1;

test\_un2 u2;

};

union un\_big{

test\_st1 t1;

test\_st2 t2;

};

// прототипы используемых в программе функций

void input\_struct(int& k, test\_st2\* t); // ручное заполнение массива структур

void output\_struct(test\_st2\* t); // вывод структуры

void output\_all\_structs(int& k, test\_st2\* t); // вывод всего массива структур

void input\_random(int& k, test\_st2\* t); // заполнение массива структур случайными числами

void sort(int& k, test\_st2\* t); // сортировка по возростанию величины поля float x

void recombination(test\_st2& t1, test\_st2& t2); // перестановка двух структур местами

int main(int argc, char\* argv[])

{

test\_st1 a[2]; int k;

for(int i = 1; i >= 0; i--){

printf("Vvedite zeloe chyslo i 1 simvol (ostalos' %d raz): ", i + 1);

cin >> a[i].x >> a[i].c;

}

cout << "\nVvedennie Vami dannie: \n";

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

cout << a[i].x << " " << a[i].c << "\n";

cout << "Vvedite kolichestvo struktur v massive (ne men'she 2): ";

cin >> k;

test\_st2\* t = new test\_st2[k]; // выделение памяти для массива структур

input\_random(k, t);

cout << "\nMassiv byl zapolnen sluchaynim obrazom... \n";

// input\_struct(k, t);

output\_struct(t);

output\_all\_structs(k, t);

sort(k, t);

cout << "\nPosle sortirovky: \n";

output\_all\_structs(k, t);

int x1, x2;

cout << "Vvedite nоmera structur dlya obmena mestami: ";

cin >> x1 >> x2;

recombination(t[x1 - 1], t[x2 - 1]);

cout << "\nPosle perestanovki: \n";

output\_all\_structs(k, t);

cout << "\nDannie sohraneni v fail... \n";

FILE\* f;

int size;

test\_st2 array;

f = fopen("lab1.dat", "r+");

if (f) {

fseek(f,0L,SEEK\_END);

size = ftell(f) / sizeof(test\_st2);

fseek(f,0L,SEEK\_SET);

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

fwrite(&t[i],sizeof(test\_st2),1,f); // запись данных в файл

rewind(f);

cout << "\nChetinei zapisannyh v fail dannyh... \n";

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

fread(&array,sizeof(test\_st2),1,f); // чтение данных из файла

cout << array.x << " " << array.s << "\n";

}

}

system("pause");

return 0;

};

void input\_struct(int& k, test\_st2\* t){

for(int i = k - 1; i >= 0; i++) {

printf("Vvedite veshestvennoe 1 chyslo i 1 simvol (ostalos' %d raz):\n", k);

scanf("%f %s", &t[i].x, &t[i].s);

}

};

void output\_struct(test\_st2\* t){

int k;

cout << "\nVVedite nomer structury dl'ya vivoda: ";

cin >> k;

cout << t[k - 1].x << " " << &t[k - 1].s << "\n";

};

void output\_all\_structs(int& k, test\_st2\* t){

cout << "\nZnacheniya vo vseh vvedinnyh structurah: \n";

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

cout << i + 1 << ". " << t[i].x << " " << &t[i].s << "\n";

};

void input\_random(int& k, test\_st2\* t){

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

t[i].x = random(1000)/10;

t[i].s = (char) random(100) + 1;

}

};

void sort(int& k, test\_st2\* t){

test\_st2 temp;

for(int i = 0; i < k - 1; i++) // пузырьковая сортировка

for(int z = i + 1; z < k; z++)

if(t[i].x > t[z].x){

temp.x = t[i].x;

t[i].x = t[z].x;

t[z].x = temp.x;

}

};

void recombination(test\_st2& t1, test\_st2& t2){

test\_st2 temp;

temp = t1;

t1 = t2;

t2 = temp;

};