**Лабораторная работа №6. Динамические структуры данных. Списки.**

**Вариант 10**

Условие:

FindMin – функция поиска минимального  элемента списка по одному из выбранных полей.

Код:

#include <iostream>

#include <fstream>

using namespace std;

fstream file;

struct Adr

{ char name[30];

char street[40];

char city[20];

struct Adr \*next;

struct Adr \*prev;

};

struct Adr \*head; struct Adr \*last;

int Menu(void)

{ char s[80]; int c; cout<<endl;

cout<<"1. Ввод имени"<<endl;

cout<<"2. Удаление имени"<<endl;

cout<<"3. Вывод на экран"<<endl;

cout<<"4. Поиск"<<endl;

cout<<"5. Сохранить в файл"<<endl;

cout<<"6. Загрузить из файла"<<endl;

cout<<"7. Выход"<<endl; cout<<endl;

do

{ cout<<"Ваш выбор: "; gets(s); cout<<endl;

c = atoi(s);

} while(c < 0 || c > 7);

return c;

}

void Sozdat(Adr \*i, Adr \*\*head, Adr \*\*last)

{ struct Adr \*old, \*p;

if(\*last == NULL)

{ i->next = NULL; i->prev = NULL;

\*last = i; \*head = i; return;

}

p = \*head; old = NULL;

while(p)

{ if(strcmp(p->name, i->name) < 0)

{ old = p;

p = p->next;

}

else

{ if(p->prev)

{ p->prev->next = i; i->next = p;

i->prev = p->prev; p->prev = i;

return;

}

i->next = p; i->prev = NULL;

p->prev = i; \*head = i;

return;

}

}

old->next = i; i->next = NULL;

i->prev = old; \*last = i;

}

void Vvod(char \*prompt, char \*s, int count)

{ char p[255];

do

{ cout<<(prompt); fgets(p, 254, stdin);

if(strlen(p) > count)

cout<<("Слишком длинная строка"); //длина строки

} while(strlen(p) > count);

p[strlen(p)-1] = 0; strcpy(s, p);

}

void VvodSp(void) // Ввод строки

{ struct Adr \*t; int i;

t = new (struct Adr);

if(!t) { cout<<("Нет свободной памяти"); return; }

Vvod("Введите имя: ", t->name, 30);

Vvod("Введите улицу: ", t->street, 40);

Vvod("Введите город: ", t->city, 20);

Sozdat(t, &head, &last);

}

void VyvodSp(void) //Вывод списка на экран

{ struct Adr \*t; t = head;

while(t)

{ cout<< t->name<< ' ' << t->street << ' ' << t->city<<endl;

t = t->next;

}

cout<<""<<endl;

}

void FindMin(void) // Поиск короткое имя в списке

{ char name[40], street[40],city[40]; struct Adr \*t; t = head;

int a[4],i=0,z,n;

while(t)

{

a[i]=strlen(t->name);

t=t->next;

i++;

}

n=a[1];

for(i=1;i<4;i++)

{

if(n>a[i])

{

n=a[i];

z=i+1;

}

}

t=head;

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

{

t=t->next;

}

cout << t->name << ' ' << t->street << ' ' << t->city<<endl;

}

void Udalit( Adr \*\*head, Adr \*\*last) // Удаление имени из списка

{ struct Adr \*t; char name[40]; t = \*head;

cout<<"Введите имя: "; gets(name);

while(t)

{ if(!strcmp(name, t->name)) break; t = t->next; }

if(t)

{ if(\*head == t)

{ \*head = t->next;

if(\*head) (\*head)->prev = NULL;

else \*last = NULL;

}

else

{ t->prev->next = t->next;

if(t != \*last) t->next->prev = t->prev;

else \*last = t->prev;

} delete t;

}

}

void Zapisat(void) //Запись в файл

{ struct Adr \*t;

fstream fp;

fp.open("mlist.txt");

if(!fp) { cout<<"Файл не открывается"<<endl; exit(1); }

cout<<"Сохранение в файл"<<endl;

t = head;

while(t)

{

fp << t->name<<" ";

fp << t->street<<" ";

fp << t->city<<"\n";

t = t->next;

}

file.close();

}

void Schitat() //Считывание из файла

{ struct Adr \*t; fstream fp;

fp.open("mlist.txt");

if(!fp) { cout<<"Файл не открывается"<<endl; exit(1); }

while(head)

{ last = head->next; delete head; head = last; }

head = last = NULL;

cout<<"Загрузка из файла"<<endl;

while(!fp.eof())

{ t = new (struct Adr);

if(!t) { cout<<"Нет свободной памяти"<<endl; return; }

fp>>t->name;

fp>>t->street;

fp>>t->city;

Sozdat(t, &head, &last);

}

fp.close();

}

int main(void)

{ setlocale (LC\_CTYPE, "Rus");

head = last = NULL;

for(;;)

{ switch(Menu())

{ case 1: VvodSp(); break;

case 2: Udalit(&head, &last); break;

case 3: VyvodSp(); break;

case 4: FindMin(); break;

case 5: Zapisat(); break;

case 6: Schitat(); break;

case 7: exit(0);

}

}

return 0;

}

Скриншот

