**Задание (Вариант №7)**

Спроектировать шаблон класса **spisok** для реализации односвязного линейного списка. При использовании шаблона класса в качестве данных для элемента односвязного списка использовать структурную переменную, содержащую следующую персональную информацию:

**+**) личный номер;

**+**) фамилия;

**+**) имя;

**+**) возраст;

**+**) страна проживания;

**+**) профессия;

**+**) хобби;

**+**) номер телефона.

Создать список из 10-15 элементов.

Разработать функции работы со списком:

добавления элемента в список;

вывод на экран данных элемента списка;

поиск информации;

сортировка списка.

**Поля** для поиска информации и сортировки списка выбрать из таблицы по последней цифре зачетной книжки.

|  |  |  |
| --- | --- | --- |
|  | **Поиск информации по полю** | **Сортировка списка по полю** |
| 1. | номер | страна проживания |
| 2. | фамилия | возраст |
| 3. | имя | номер |
| 4. | возраст | профессия |
| 5. | страна проживания | возраст |
| 6. | профессия | номер |
| 7. | хобби | фамилия |
| 8. | фамилия | номер |
| 9. | возраст | хобби |
| 0. | номер | имя |

Интерфейс пользователя разработать в виде меню.

*Текст программы:*

#include <iostream>

#include <iomanip>

#include <cstring>

using namespace std;

 struct stud

 {

 int Lich\_nom;

 char LastName[15];

 char Name[10];

 int Old;

 char Country[10];

 char Job[15];

 char Hobby[15];

 char Numb\_Phone[13];

 };

 stud Stud[10]={{123,"Ivanov ","Ivan ",22,"Russia ","IT-ingener ","Avia","347-10-88"},

 {124,"Ivanov ","Urii ",25,"Russia ","Administrator ","Cooking","347-11-88"},

 {125,"Prtrov ","Sergei ",25,"Canada ","Manager ","Music","347-12-88"},

 {126,"Sidorov ","Pavel ",21,"Russia ","Operator-PC ","Football","347-13-88"},

 {127,"Smirnov ","Grigorii ",23,"USA ","Manager ","Swiming","347-14-88"},

 {128,"Losin ","Evgenii ",24,"Russia ","IT-ingener ","Computer","347-15-88"},

 {129,"Elec ","Sergei ",24,"Russia ","Administrator ","Computer","347-16-88"},

 {130,"Kondakova ","Marina ",26,"Russia ","Buhgalter ","Dekypaj","347-17-88"},

 {131,"Yakovleva ","Yana ",23,"USA ","Programmist ","Treveling","347-18-88"},

 {132,"Malihin ","Vasilii ",26,"Russia ","Operator-PC ","Dansing","347-19-88"}

 };

 template <class data\_t>class spisok

 {

 data\_t data;

 spisok \*next;

 public:

 spisok(data\_t d);

 void ADD(spisok \*node) {node->next=this;next=0;}

 spisok \*getnext(){return next;}

 data\_t getdata(){return data;}

 };

 template <class data\_t> spisok<data\_t>::spisok(data\_t d)

 {

 data=d;

 next=0;

 };

void menu()//функция вывода меню

{

 cout << "1. ADD ELEMENI IN THE SPILOK \n";

 cout << "2. PRINT DATA \n";

 cout << "3. FINDE INFORMATION \n";

 cout << "4. SORT SPISOK \n";

 cout << "5. EXIT \n\n";

};

void select()//функция выбора пункта меню

{

 char N;

 do

 {

 cout << "Select MENU - Click '0' ";

 cin >> N;

 if (N == '0') { system("cls"); menu(); }

 }

 while (N != '0');

};

 void Head()//функция вывода шапки таблицы

 {

 cout<<"NUMBER ";

 cout<<"LAST NAME ";

 cout<<"NAME ";

 cout<<" OLD ";

 cout<<" COUNTRY ";

 cout<<" PROFESSION ";

 cout<<" HOBBY ";

 cout<<" PHONE \n";

 };

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

{

 int Age, i;

 char N;

 stud tmp, temp, St,St2;

 spisok<struct stud> start(Stud[0]);

 spisok<struct stud> \*p,\*last;

 //создание списка

 last = &start;

 for(i=0;i<10;i++)

 {

 p = new spisok<struct stud> (Stud[i]);

 p->ADD(last);

 last=p;

 }

 do

 {

 system("cls");

 menu();

 cout<<"Click Number!! ";

 cin >> N;

 switch (N)

 {

 case '1':

 system("cls");

 struct stud Stu;

 cout<<"Entre Lichnii Number "; cin>>Stu.Lich\_nom;

 cout<<"Entre Last Name "; cin>>Stu.LastName;

 cout<<"Entre Name "; cin>>Stu.Name;

 cout<<"Entre Age "; cin>>Stu.Old;

 cout<<"Entre Country "; cin>>Stu.Country;

 cout<<"Entre Profession "; cin>>Stu.Job;

 cout<<"Entre Hobbby "; cin>>Stu.Hobby;

 cout<<"Entre Number Phone "; cin>>Stu.Numb\_Phone;

 cout << "\n\n 1. GOOD!! \n\n";

 p = new spisok<struct stud> (Stu);

 p->ADD(last);

 last=p;

 select();

 break;

 case '2':

 system("cls");

 Head();

 p=&start;

 while(p)

 {

 temp=p->getdata();

 cout.setf(ios::fixed);

 cout<<setw(8)<<left<<temp.Lich\_nom;

 cout<<setw(10)<<left<<temp.LastName;

 cout<<setw(10)<<left<<temp.Name;

 cout<<setw(8)<<left<<temp.Old;

 cout<<setw(10)<<left<<temp.Country;

 cout<<setw(10)<<left<<temp.Job;

 cout<<setw(10)<<left<<temp.Hobby;

 cout<<temp.Numb\_Phone;

 p=p->getnext();

 cout<<"\n";

 }

 cout << "2. GOOD!! \n\n";

 select();

 break;

 case '3':

 system("cls");

 p=&start;

 cout<<"ENTER THE DESIRES AGE ";

 cin>>Age;

 system("cls");

 Head();

 while(p)

 {

 St=p->getdata();

 if(Age==St.Old)

 {

 if(p!=NULL)

 {

 cout.setf(ios::fixed);

 cout<<setw(6)<<left<<St.Lich\_nom;

 cout<<setw(10)<<left<<St.LastName;

 cout<<setw(10)<<left<<St.Name;

 cout<<setw(6)<<left<<St.Old;

 cout<<setw(10)<<left<<St.Country;

 cout<<setw(10)<<left<<St.Job;

 cout<<setw(10)<<left<<St.Hobby;

 cout<<St.Numb\_Phone;

 cout<<"\n";

 }

 }

 p=p->getnext();

 };

 cout << "3. GOOD!! \n\n";

 select();

 break;

 case '4':

 system("cls");

 p=&start;

 while (p)

 {

 St=p->getdata();

 last=p->getnext();

 while (last)

 {

 St2=last->getdata();

 if(strcmp(St.Hobby,St2.Hobby)>0)

 {

 tmp=St;

 St=St2;

 St2=tmp;

 }

 last=last->getnext();

 }

 p=p->getnext();

 }

 Head();

 p=&start;

 while(p)

 {

 St=p->getdata();

 cout.setf(ios::fixed);

 cout<<setw(8)<<left<<St.Lich\_nom;

 cout<<setw(10)<<left<<St.LastName;

 cout<<setw(10)<<left<<St.Name;

 cout<<setw(8)<<left<<St.Old;

 cout<<setw(10)<<left<<St.Country;

 cout<<setw(10)<<left<<St.Job;

 cout<<setw(10)<<left<<St.Hobby;

 cout<<St.Numb\_Phone;

 p=p->getnext();

 cout<<"\n";

 }

 cout << "4. GOOD!! \n\n";

 select();

 break;

 case '5': break;

 default: break;

 }

 } while (N != '5');

 system("pause");

 return 0;

}