Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им. И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление: 710400 «Программная инженерия»

Дисциплина: «Объектно-ориентированное программирование»

**ОТЧЕТ**

**По лабораторной работе №2**

Тема: Классы и объекты

Выполнила: студент группы

ПИ(б)-2-21 Мааданбекова Аяна

Проверил: Мусабаев Э. Б

Бишкек – 2024

Лабораторная работа №2

1. В здании аэропорта на экранах отображается информация о самолетах, а именно: информация о пункте отправления, пункте назначения, номере рейса, времени прибытия, времени отправления, номере секции для регистрации. Экраны – это средство, которое помогает своевременно зарегистрировать и отправить пассажиров. Важно, чтобы информация на экранах была понятной и верной.

Создайте необходимую информацию в виде таблицы для такого экрана, с помощью класса Aeroflot, содержащего в описании следующие поля: номер рейса; название пункта отправления; название пункта назначения рейса; время прибытия; время отправления; место регистрации.

Напишите код программы, выполняющей следующие действия: ввод с клавиатуры значений полей объектов; сортировку записей в таблице в алфавитном порядке по названию пунктов назначения; вывод на консоль значений полей объектов класса в виде таблицы рейсов; если таких рейсов нет, выдать соответствующее сообщение.

**Код программы:**

#include<iostream>

#include <iomanip>

using namespace std;

class Aeroflot // определение класса Aeroflot

{

int flightNumber;

string departurePointName;

string arrivalPointName;

string departureTime;

string arrivalTime;

string registrationPlace;

public:

void setAeoroflot()

{

cout << "Number of flight: ";

cin >> flightNumber;

cout << "Departure point name: ";

cin >> departurePointName;

cout << "Arrival point name: ";

cin >> arrivalPointName;

cout << "Departure time: ";

cin >> departureTime;

cout << "Arrival time: ";

cin >> arrivalTime;

cout << "Registration place: ";

cin >> registrationPlace;

}

void output()

{//Вывод

cout << " | " << setw(11) << flightNumber << " |"

<< setw(18) << departurePointName << " |"

<< setw(17) << arrivalPointName << " |"

<< setw(18) << departureTime << " |"

<< setw(15) << arrivalTime << " |"

<< setw(18) << registrationPlace << " |" << endl;

}

string getArrivalName() //Пункт назначения

{

return arrivalPointName;

}

void sort(Aeroflot arr[], size\_t size) {

Aeroflot t;

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

for (int j = 0; j < size - 1; j++) {

if (arr[j].getArrivalName() > arr[j + 1].getArrivalName()) {

t = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = t;

}

}

}

cout << "Sorting was successful! " << endl << endl;

}

void poisk(Aeroflot arr[], size\_t size) {

bool f = 0;

string a;

cout << "Enter the arrival point name: ";

cin >> a;

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

if (a == arr[i].arrivalPointName) {

cout << endl;

cout << " Table of flight" << endl << endl;

cout << setw(14) << "| Number of flight "

<< setw(20) << "| Departure point name "

<< setw(19) << "| Arrival point name "

<< setw(20) << "| Departure time "

<< setw(17) << "| Arrival time "

<< setw(21) << "| Registration place |" << endl;

cout << " +-------------------------------------------------------------------------------------------------------------+" << endl;

arr[i].output();

f = 1;

break; // выход из цикла, так как рейс найден

}

}

if (f == 0) {

cout << "Flight not found!" << endl << endl;

}

}

};

int main()

{

system("chcp 1251"); // изменение кодировки консоли для валидного ввода символов русского алфавита

Aeroflot info[10];

int choose, m;

do {

cout << " Menu " << endl;

cout << "1 - Add" << endl;

cout << "2 - Show list" << endl;

cout << "3 - Sort" << endl;

cout << "4 - Search" << endl;

cout << "0 - Exit" << endl;

cin >> choose;

switch (choose)

{

case 0:

{

return 0;

}

break;

case 1:

cout << "Flight count: ";

cin >> m;

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

{

info[i].setAeoroflot();//ввод

}

break;

case 2:

system("cls"); //очистка консоли

cout << endl;

cout << " Table of flight" << endl << endl;

cout << setw(14) << "| Number of flight "

<< setw(20) << "| Departure point name "

<< setw(19) << "| Arrival point name "

<< setw(20) << "| Departure time "

<< setw(17) << "| Arrival time "

<< setw(21) << "| Registration place |" << endl;

cout << " +-------------------------------------------------------------------------------------------------------------+" << endl;

{

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

{

info[i].output();

}

}

break;

case 3:

{

system("cls"); //очистка консоли

info[m].sort(info, m);

}

break;

case 4:

{

info[m].poisk(info, m);//поиск по пункту назначения

}

break;

}

} while (choose != 0);

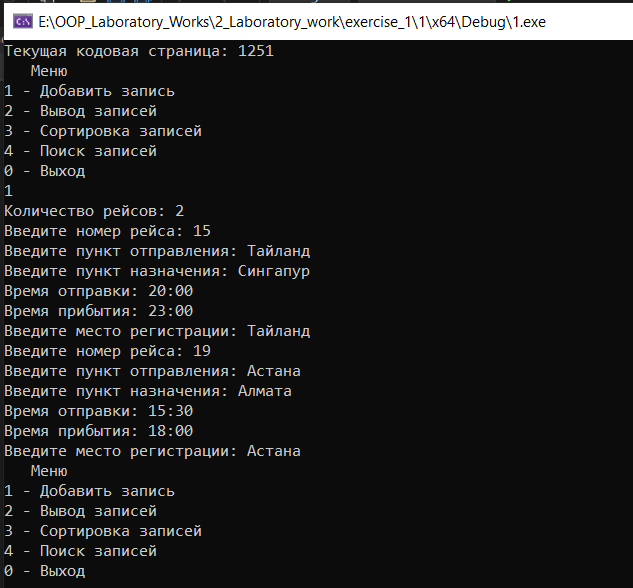
system("pause");

return 0;

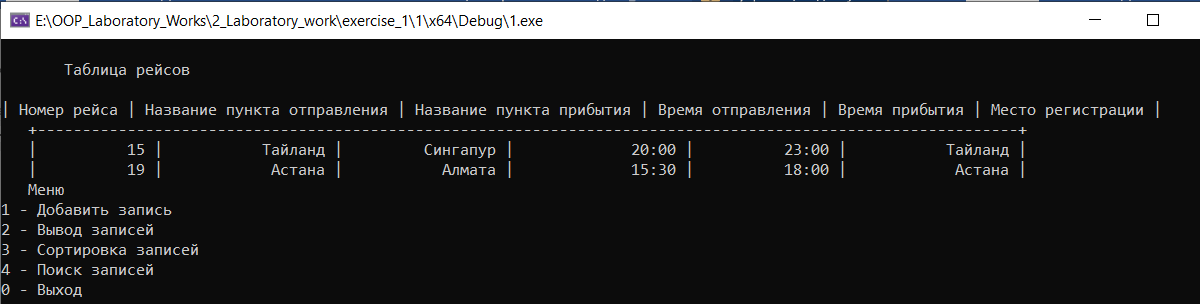
}

**Результат:**

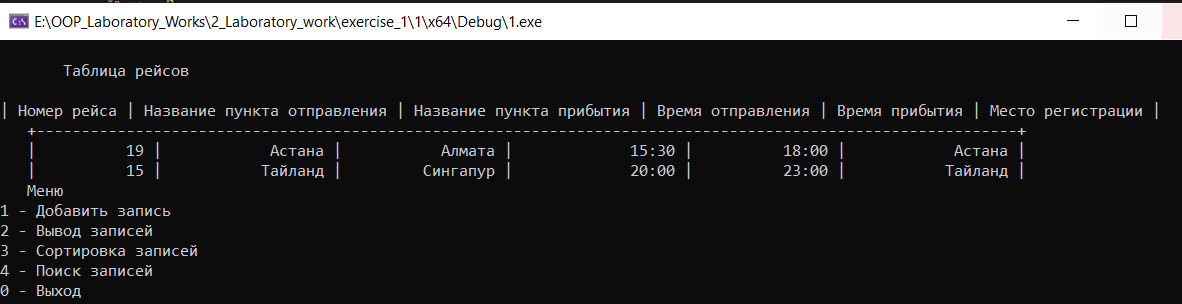
Добавление записей в таблицу:



Вывод данных:



Сортировка:



1. **Опишите класс «студенческая группа».**

Предусмотрите возможность: работы с переменным числом студентов; поиска студента по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона); добавления или удаления записей; сортировки по разным полям.

Разработайте программу, демонстрирующую работу с этим классом.

Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

**Код программы:**

#include <iostream>

#include<string>

#include <conio.h>

#include <process.h>

#include <iomanip>

#include<Windows.h>

using namespace std;

//Класс Студент

class Student

{

string surname;//Строка Фамилия

int tel = 0;//Типы данные номера телефона

string date;//Строка Даты рождения

int k = 0;//Количество

public:

//Метод ввода

void vvod()

{

cout << "Введите фамилию студента: ";

cin >> surname;

cout << "Введите номер телефона: ";

cin >> tel;

cout << "Дата рождения: ";

cin >> date;

k++;

}

//Метод вывода

void prosmotr()

{

cout << "|" << surname << "\t|" << tel << "\t|" << date << " |" << endl;

cout << "|---------------|---------------|-------------|" << endl;

}

//Метод поиска по Фамилии

void search\_surname(Student s[], size\_t size)

{

if (k != 0)

{

string sur;

cout << "Введите фамилию студента: ";

cin >> sur;

for (int i = 0; i <= k; i++)//Цикл для поиска студента по фамилии

{

if (sur == s[i].surname)

{

cout << "Студент найден!" << endl;

cout << "|Фамилия" << "\t|Номер телефона" << "\t|Дата рождения|" << endl;

cout << "|===============|===============|=============|" << endl;

s[i].prosmotr();

}

else

{

cout << "Такого студента нет!" << endl;

}

}

}

else cout << "База пуста!!";

}

//Метод поиска по номеру телефона

void search\_tel(Student s[], size\_t size)

{

if (k != 0)

{

int nomer;

cout << "Введите номер телефона: ";

cin >> nomer;

for (int i = 0; i <= k; i++)//Цикл для поиска студента по номеру телефона

{

if (nomer == s[i].tel)

{

cout << "Номер найден!" << endl;

cout << "|Фамилия" << "\t|Номер телефона" << "\t|Дата рождения|" << endl;

cout << "|===============|===============|=============|" << endl;

s[i].prosmotr();

}

else

{

cout << "Такого номера нет!" << endl;

}

}

}

else cout << "База пуста!!";

}

//Метод поиска по Дате рождения

void search\_date(Student s[], size\_t size)

{

if (k != 0)

{

string dat;

cout << "Введите дату рождения: ";

cin >> dat;

for (int i = 0; i <= k; i++)//Цикл для поиска студента по дате рождения

{

if (dat == s[i].date)

{

cout << "Номер найден!" << endl;

cout << "|Фамилия" << "\t|Номер телефона" << "\t|Дата рождения|" << endl;

cout << "|===============|===============|=============|" << endl;

s[i].prosmotr();

}

else

{

cout << "Такой даты рождения нет!" << endl;

}

}

}

else cout << "База пуста!!";

}

//Получение данных

string getSurname()

{

return surname;

}

//Передача данных

void setSurname(string temp)

{

surname = temp;

}

};

//Меню

int main()

{

//Кириллица

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string item;//Для удаления по фамилии

int n = 5;//Количесвто

Student\* s = new Student[n];

int num, num1;//Выбор операции

int k = 0;

do

{

system("cls");

cout << "\t\tГлавное меню" << endl;

cout << "\t1. Добавить запись" << endl;

cout << "\t2. Показать запись" << endl;

cout << "\t3. Поиск" << endl;

cout << "\t4. Удаление" << endl;

cout << "\t5. Выход" << endl;

cout << endl;

cout << "Введите номер операции: ";

cin >> num;

switch (num)

{

case 1:

system("cls");

if (k == n)

{

k = 0;

}

s[k].vvod();

k++;

system("pause");

break;

case 2:

system("cls");

if (k == 0)

{

cout << "База пуста!" << endl;

}

else

{

cout << "|Фамилия" << "\t|Номер телефона" << "\t|Дата рождения|" << endl;

cout << "|===============|===============|=============|" << endl;

for (int i = 0; i < k; i++)//Цикл для вывода введенных данных

{

s[i].prosmotr();

}

}

system("pause");

break;

case 3:

system("cls");

if (k == 0)

{

cout << "База пуста!" << endl;

}

else

{

cout << "Введите признаки по которым будете находить студента." << endl;

cout << "\t1. По фамилии" << endl;

cout << "\t2. По номеру телефона" << endl;

cout << "\t3. По дате рождения" << endl;

cout << endl;

cin >> num1;

if (num1 == 1)

{

system("cls");

s[0].search\_surname(s, n);

}

else if (num1 == 2)

{

system("cls");

s[0].search\_tel(s, n);

}

else if (num1 == 3)

{

system("cls");

s[0].search\_date(s, n);

}

}

system("pause");

break;

case 4:

system("cls");

cout << "Введите фамилию студента: ";

cin >> item;

for (int i = 0; i < n; i++)//Цикл

{

if (s[i].getSurname() == item)

{

s[i].setSurname(" ");

}

}

for (int j = 1; j < n; j++)

{

for (int i = 0; i < n - j; i++)

{

if (s[i].getSurname() == " ")

{

swap(s[i], s[i + 1]);

}

}

}

n--;

k--;

system("pause");

break;

case 5:

system("cls");

cout << "До свидания!" << endl;

exit(0);

break;

}

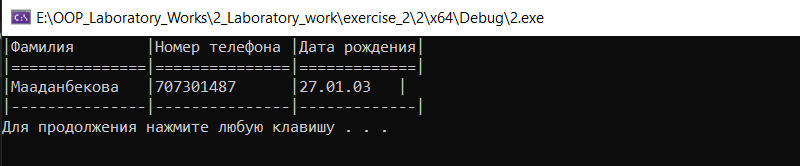
} while (num != '0');

delete[]s;

return 0;

}

**Результат:**



**3. Опишите класс, реализующий стек (Stack).**

Разработать программу, использующую этот класс для моделирования Т-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла или с клавиатуры. Возможно использование стандартных функций при работе со стеком в виде контейнера из библиотеки STL:

**push()** - поместить элемент в вершину стека;

**pop()** - удалить элемент из вершины стека;

**Код программы:**

#include<iostream>

using namespace std;

class stac

{

public:

struct stak

{

int item;

stak\* next;

};

stak\* top = NULL;

stak\* top1 = NULL;

stak\* top0 = NULL;

void add() {

stak\* pv = new stak;

pv->next = NULL;

cout << "Enter item 0 or 1:" << endl;

cin >> pv->item;

if (top == NULL)

{

top = pv;

}

else

{

pv->next = top;

top = pv;

}

pv = NULL;

}

void print(stak\*\* top) {

stak\* pv;

if (\*top == NULL)

{

cout << "Stak is empty!" << endl;

}

else

{

pv = \*top;

while (pv != NULL)

{

cout << pv->item << "";

pv = pv->next;

}

}

cout << endl;

}

void Sort() {

stak\* pv;

if (top == NULL)

{

cout << "Stak is empty!" << endl;

}

else

{

while (top != NULL)

{

pv = top;

top = top->next;

pv->next = NULL;

if (pv->item == 1)

{

if (top1 == NULL) {

top1 = pv;

}

else

{

pv->next = top1;

top1 = pv;

}

}

else

{

if (top0 == NULL)

{

top0 = pv;

}

else

{

pv->next = top0;

top0 = pv;

}

}

pv = NULL;

}

}

cout << "Main stak with 0 and 1 was successful sorting" << endl;

}

};

int main() {

setlocale(0, "");

int key;

stac train;

re:

cout << "1.Add\n2.Show\n3.Sort" << endl;

cin >> key;

switch (key) {

case 1:

cout << "How many items you want add?" << endl;

cin >> key;

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

train.add();

}

break;

case 2:

cout << "Which stak you want watch?" << endl;

cout << "1.Stak with 0 и 1" << endl;

cout << "2.Stak with 1?" << endl;

cout << "3.Stak with 0?" << endl;

cin >> key;

switch (key)

{

case 1:

train.print(&train.top);

break;

case 2:

train.print(&train.top1);

break;

case 3:

train.print(&train.top0);

break;

}

break;

case 3:

train.Sort();

break;

default:

cout << "Adding error!" << endl;

break;

}

goto re;

system("pause");

}