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 Марат кызы Жылдыз

Проверил: Мусабаев Э. Б.

**Бишкек – 2024**

**Лабораторная работа №2.**

**Тема: Классы и объекты.**

**Задания:**

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<string>

#include<Windows.h>

using namespace std;

class Aeroflot {

public:

int rays, h, m, hh, mm;

string mesto, naz, punkt;

};

int main() {

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

Aeroflot af[100];

int i = 0;

string ans;

bool f = true, tr = true;

while (f == true) {

cout << "Введите номер рейса: ";

cin >> af[i].rays;

cout << "Пункт отправления: ";

cin >> af[i].naz;

cout << "Пункт назначения: ";

cin >> af[i].punkt;

cout << "Время отправления(час/мин): ";

cin >> af[i].h >> af[i].m;

cout << "Время прибытия(час/мин): ";

cin >> af[i].hh >> af[i].mm;

cout << "место регистрации: ";

cin >> af[i].mesto;

cout << "Будут ли еще рейсы?(y/n): ";

cin >> ans;

if (ans == "y")

i++;

else

f = false;

}

cout << endl;

while (true) {

cout << "1)Выбор рейса: " << endl;

cout << "2) Показать все рейсы: " << endl;

cout << "3)Добавить рейс" << endl;

int vyb;

cin >> vyb;

if (vyb == 1) {

string pnk;

cout << "Введите пункт назначения: ";

cin >> pnk;

cout << "Номер рейса" << "\t Пункт отправления: " << "\t Пункт назначения : " << "\t Время отправления" << "\t Время прибытия" << "\t место регистрации" << endl;

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

if (pnk == af[j].punkt) {

tr = false;

cout << af[j].rays << "\t \t" << af[j].naz << "\t \t \t" << af[j].punkt << "\t \t \t" << af[j].h << ":";

if (af[j].m == 0)

cout << "00";

else

cout << af[j].m;

cout << "\t \t" << af[j].hh << ":";

if (af[j].mm == 0)

cout << "00";

else

cout << af[j].mm;

cout << "\t \t \t" << af[j].mesto << endl;

}

}

if (tr == true) {

cout << "такого пункта нету" << endl;

}

}

if (vyb == 2) {

cout << "Номер рейса" << "\t Пункт отправления: " << "\t Пункт назначения : " << "\t Время отправления" << "\t Время прибытия" << "\t место регистрации" << endl;

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

{

cout << af[j].rays << "\t \t" << af[j].naz << "\t \t \t " << af[j].punkt << "\t \t \t" << af[j].h << ":";

if (af[j].m == 0)

cout << "00";

else

cout << af[j].m;

cout << "\t \t" << af[j].hh << ":";

if (af[j].mm == 0)

cout << "00";

else

cout << af[j].mm;

cout << "\t \t \t" << af[j].mesto << endl;

}

}

}

if (vyb == 3) {

i++;

cout << "Введите номер рейса: ";

cin >> af[i].rays;

cout << "Пункт отправления: ";

cin >> af[i].naz;

cout << "Пункт назначения : ";

cin >> af[i].punkt;

cout << "Время прибытия(час/мин): ";

cin >> af[i].h >> af[i].m;

cout << "Время отправления(час/мин): ";

cin >> af[i].hh >> af[i].mm;

cout << "место регистрации: ";

cin >> af[i].mesto;

}

}

}

**Результат программы:**

Рис.1. **Ввод рейса.**

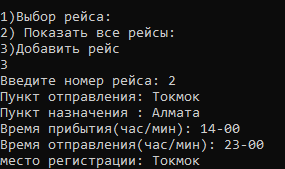
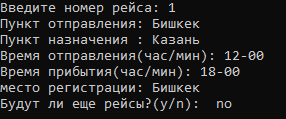


Рис.2. **Все рейсы.**

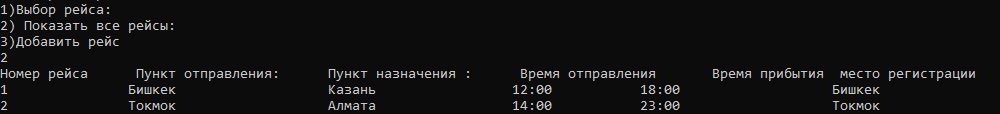
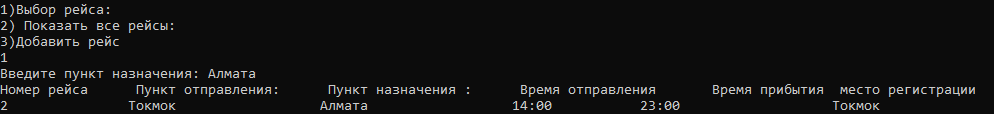


Рис.3. **Выбор определенного рейса.**



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<iomanip>

#include<string>

#include<Windows.h>

using namespace std;

class Student {

string name;

string sername;

int day;

int month;

int year;

string group;

public:

void add() {

cout << "Введите фамилию студента: ";

cin >> sername;

cout << "Введите имя студента: ";

cin >> name;

cout << "Группа студента: ";

cin >> group;

cout << "День рождение студента: " << endl;

cout << "Год: ";

cin >> year;

cout << "месяц(число): ";

cin >> month;

cout << "день: ";

cin >> day;

}

string poiskName(string n) {

if (name == n)

print();

return n;

}

string poiskSername(string n) {

if (sername == n)

print();

return n;

}

string poiskGroup(string n) {

if (group == n)

print();

return n;

}

string getName() {

return name;

}

string getSername() {

return sername;

}

string getGroup() {

return group;

}

string print() {

cout << setw(11) << sername << setw(11) << name << setw(10) << group << setw(10);

if (day < 10)

cout << "0" << day << ".";

else

cout << day << ".";

if (month < 10)

cout << "0" << month << "." << year << endl;

else

cout << month << "." << year << endl;

return name;

}

void poisk(string nm, int\* k) {

if (name == nm) {

\*k = \*k + 1;

cout << \*k;

print();

}

}

void del(string nm, int d, int\* l, bool\* tr) {

if (name == nm) {

\*l = \*l + 1;

}

if (\*l == d) {

\*tr = false;

}

}

};

int main() {

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

int val, i = 0, vall;

string n, nm;

Student student[100];

Student m;

while (true) {

metka:cout << "\tМеню" << endl;

int k = 0, d, l = 0, t = 0;

bool tr = true;

cout << "1) Добавить" << "\n2) Поиск" << "\n3) Сортировка" << "\n4) Удаление" << "\n5) Выход" << endl;

cout << "Выберите действие: ";

cin >> val;

if (val == 1) {

student[i].add();

i++;

system("pause");

system("CLS");

}

else if (val == 2) {

cout << "1) Поиск по имени" << "\n2) Поиск по фамилии " << "\n3) Поиск по группам" << endl;

cout << "Выберите действие: ";

cin >> vall;

if (vall == 1) {

cout << "Введите имя: ";

cin >> n;

cout << setw(11) << "Фамилия" << setw(11) << "Имя" << setw(10) << "Группа" << setw(10) << "ДР" << endl;

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

student[j].poiskName(n);

}

if (vall == 2) {

cout << "Введите фамилию: ";

cin >> n;

cout << setw(11) << "Фамилия" << setw(11) << "Имя" << setw(10) << "Группа" << setw(10) << "ДР" << endl;

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

student[j].poiskSername(n);

}

if (vall == 3) {

cout << "Введите группу: ";

cin >> n;

cout << setw(11) << "Фамилия" << setw(11) << "Имя" << setw(10) << "Группа" << setw(10) << "ДР" << endl;

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

student[j].poiskGroup(n);

}

system("pause");

system("CLS");

}

else if (val == 3)

{

cout << "1) Сортировка по имени" << "\n2) Сортировка по фамилии " << "\n3) Сортировка по группам" << endl;

cout << "Выберите действие: ";

cin >> vall;

if (vall == 1) {

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

for (int h = 0; h < j; h++) {

if (student[h].getName() > student[h + 1].getName()) {

m = student[h];

student[h] = student[h + 1];

student[h + 1] = m;

}

}

cout << setw(11) << "Фамилия" << setw(11) << "Имя" << setw(10) << "Группа" << setw(10) << "ДР" << endl;

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

student[j].print();

}

if (vall == 2) {

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

for (int h = 0; h < j; h++) {

if (student[h].getSername() > student[h + 1].getSername()) {

m = student[h];

student[h] = student[h + 1];

student[h + 1] = m;

}

}

cout << setw(11) << "Фамилия" << setw(11) << "Имя" << setw(10) << "Группа" << setw(10) << "ДР" << endl;

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

student[j].print();

}

if (vall == 3) {

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

for (int h = 0; h < j; h++) {

if (student[h].getGroup() > student[h + 1].getGroup()) {

m = student[h];

student[h] = student[h + 1];

student[h + 1] = m;

}

}

cout << setw(11) << "Фамилия" << setw(11) << "Имя" << setw(10) << "Группа" << setw(10) << "ДР" << endl;

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

student[j].print();

}

system("pause");

system("CLS");

}

else if (val == 4)

{

cout << "Введите имя: ";

cin >> nm;

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

student[j].poisk(nm, &k);

}

if (k > 0) {

cout << "выберите: ";

cin >> d;

while (t < i && tr == true) {

student[t].del(nm, d, &l, &tr);

if (tr == false) {

for (int j = t; j < i; j++)

student[j] = student[j + 1];

i--;

}

t++;

}

}

else

cout << "не найдено" << endl;

system("pause");

system("CLS");

}

else if (val == 5) {

exit(0);

system("pause");

system("CLS");

}

else {

cout << "Выберайте из меню!" << endl;

goto metka;

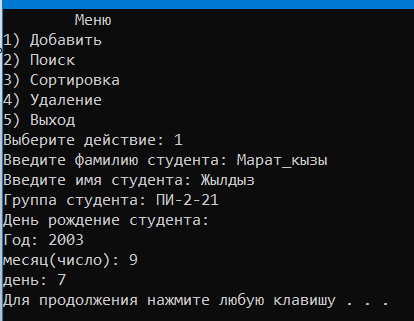
}

}

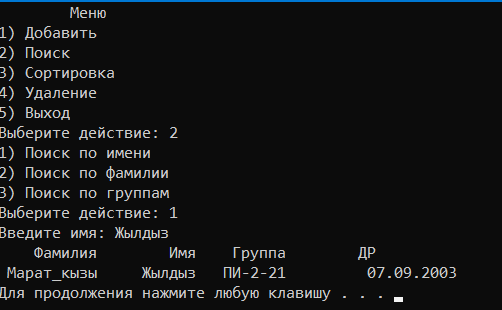
}

**Результат программы:**

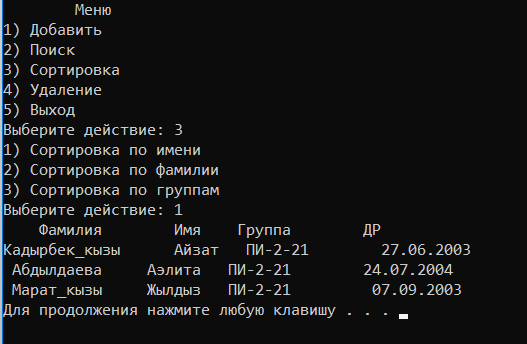
**Рис.1. Ввод данных студента.**

****

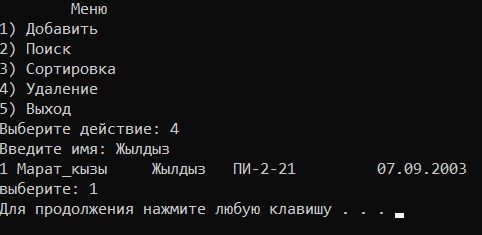
**Рис.2. Поиск студента по критериям.**

****

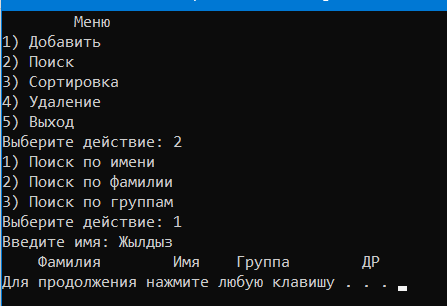
**Рис.3. Сортировка студентов по критериям.**

****

**Рис.4. Удаление студента.**

****

**Рис.4.1. Студент удален.**

****

1. **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>

#include <cstdlib>

#include <ctime>

#include <Windows.h>

#include <fstream>

#include<stack>

using namespace std;

class STACK {

private:

stack <int>myStack1;

stack <int>myStack2;

stack <int>myStack3;

stack <int>myStack4;

public:

void input(int a) {

myStack1.push(a);

myStack2 = myStack1;

}

void raz(int n) {

if (myStack1.empty()) {

cout << "Нет данных!" << endl;

}

else {

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

if (myStack1.top() % 2 == 0) {

myStack3.push(myStack1.top());

myStack1.pop();

}

else {

myStack4.push(myStack1.top());

myStack1.pop();

}

}

cout << "Деление завершено!" << endl;

}

}

void file\_Show() {

ifstream fin;

fin.open("file2.txt");

if (!fin.is\_open()) {

cout << "Файл не открылся! " << endl;

}

else {

int a;

while (fin >> a) {

myStack1.push(a);

}

myStack2 = myStack1;

}

fin.close();

cout << "Данные считаны из файла!!" << endl;

}

void Show() {

myStack2 = myStack1;

if (!myStack2.empty()) {

while (!myStack2.empty()) {

cout << myStack2.top() << endl;

myStack2.pop();

}

}

else {

cout << "Стек пуст!" << endl;

}

}

void chetn() {

myStack2 = myStack3;

if (!myStack2.empty()) {

while (!myStack2.empty()) {

cout << myStack2.top() << endl;

myStack2.pop();

}

}

else {

cout << "Нет таких вагонов!" << endl;

}

}

void ne\_chetn() {

myStack2 = myStack4;

if (!myStack2.empty()) {

while (!myStack2.empty()) {

cout << myStack2.top() << endl;

myStack2.pop();

}

}

else {

cout << "Нет таких вагонов!" << endl;

}

}

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "RUS");

srand(time(NULL));

int col = 0, zn = 0;

int val = 0;

cout << "Введите количество вагонов: ";

cin >> col;

ofstream fout;

STACK obj;

fout.open("file2.txt");

if (!fout.is\_open()) {

cout << "Файл не открылся!" << endl;

}

else {

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

fout << rand() % 20 << endl;

}

}

fout.close();

while (val != 6)

{

system("CLS");

cout << "\tМеню" << endl;

cout << "0) Ввод данных вагонов" << "\n1) Формирование состава из файла" << "\n2) Вывод " << "\n3) Разделить вагоны" << "\n4) Вывод четных вагонов" << "\n5) Вывод нечетных вагонов" <<"\n6) Выход" << endl;

cout << endl << "Выберите из меню: ";

cin >> val;

if (val == 0) {

system("CLS");

cout << "Введите номера вагонов (int): " << endl;

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

cin >> zn;

obj.input(zn);

}

cout << endl << "Данные записаны!" << endl << endl;

system("pause");

}

else if (val == 1) {

system("CLS");

obj.file\_Show();

system("pause");

}

else if (val == 2) {

system("CLS");

cout << "\tВсе вагоны: " << endl << endl;

obj.Show();

system("pause");

}

else if (val == 3) {

system("CLS");

obj.raz(col);

system("pause");

}

else if (val == 4) {

system("CLS");

cout << "\tВагоны с четными номерами: " << endl << endl;

obj.chetn();

system("pause");

}

else if (val == 5) {

system("CLS");

cout << "\tВагоны с нечетными номерами: " << endl << endl;

obj.ne\_chetn();

system("pause");

}

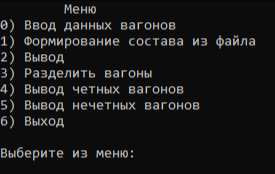
}

cout << "До свидания!" << endl;

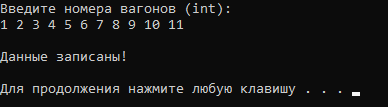
}

**Результат программы:**

**Рис.1. Меню.**

****

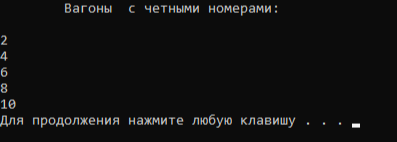
**Рис.2. Ввод данных вагонов (0)**

****

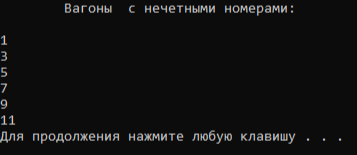
**Рис.3. Разделение вагонов. (3)**

****

**Рис.4. Четные вагоны. (4)**

****

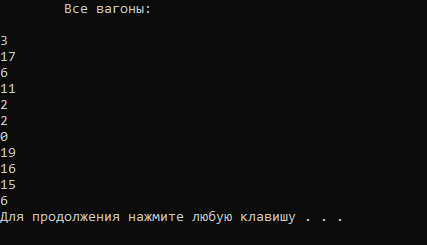
**Рис.5. Нечетные вагоны. (5)**

****

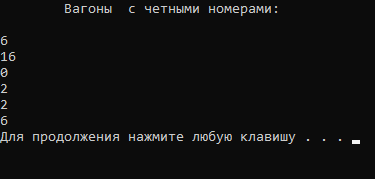
**Рис.6. Формирование состава из файла (1)**

****

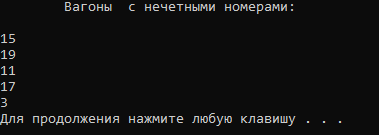
**Рис.7. Вывод из файла (2)**

****

**Рис.8. Четные вагоны.**

****

**Рис.9. Нечетные вагоны.**

****