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-ориентрованное проектирование / Объектно-ориентрованное программирование»

**ОТЧЕТ**

По лабораторной работе №2

Тема: «Классы и объекты»

Выполнила: студентка группы

ПИ(б)-3-21 Пак Ксения

Проверил: Мусабаев Э. Б.

Бишкек – 2023

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.

#include <string>

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

class Aeroflot {

private:

int flightNumber;

string departureCity;

string destinationCity;

string departureTime;

string arrivalTime;

int registrationSection;

public:

Aeroflot(int number, const string& departure, const string& destination,

const string& arrival, const string& departureTime, int section)

: flightNumber(number), departureCity(departure), destinationCity(destination),

arrivalTime(arrival), departureTime(departureTime), registrationSection(section) {}

void inputFlightData() {

cout << "Enter Flight Number: ";

cin >> flightNumber;

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << "Enter Departure City: ";

getline(cin, departureCity);

cout << "Enter Destination City: ";

getline(cin, destinationCity);

cout << "Enter Arrival Time: ";

getline(cin, arrivalTime);

cout << "Enter Departure Time: ";

getline(cin, departureTime);

cout << "Enter Registration Section: ";

cin >> registrationSection;

}

void displayFlightData() const {

cout << flightNumber << "\t\t" << departureCity << "\t\t" << destinationCity << "\t"

<< arrivalTime << "\t\t" << departureTime << "\t\t" << registrationSection << endl;

}

const string& getDestinationCity() const {

return destinationCity;

}

};

bool compareFlights(const Aeroflot& flight1, const Aeroflot& flight2) {

return flight1.getDestinationCity() < flight2.getDestinationCity();

}

int main() {

vector<Aeroflot> flights;

int numFlights;

cout << "Enter the number of flights: ";

cin >> numFlights;

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

cout << "\nEnter details for Flight " << i + 1 << ":\n";

Aeroflot flight(0, "", "", "", "", 0);

flight.inputFlightData();

flights.push\_back(flight);

}

sort(flights.begin(), flights.end(), compareFlights);

cout << "\nFlight Number\tDeparture City\tDestination City\tArrival Time\tDeparture Time\tRegistration Section\n";

for (const auto& flight : flights) {

flight.displayFlightData();

}

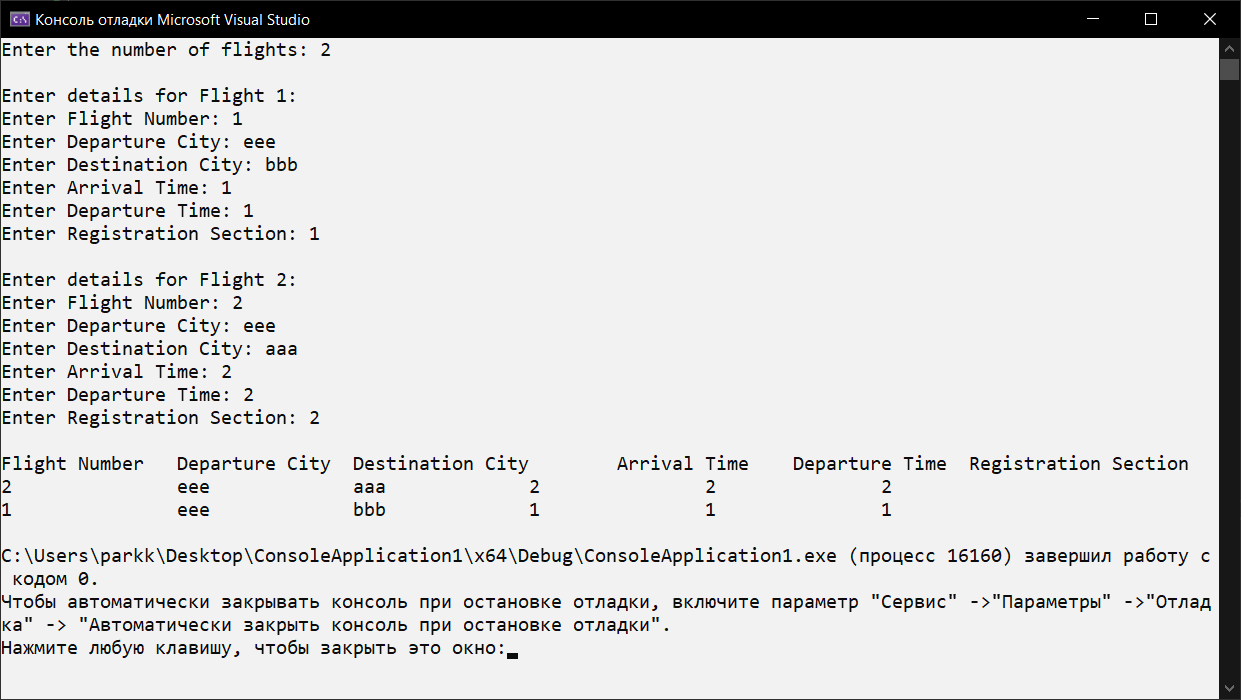
if (flights.empty()) {

cout << "\nNo flights available.\n";

}

return 0;

}



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.

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.

Дополнительное задание

1. Разработайте спецификации требований к ПО, которые решают эти задачи.
2. Спроектируйте ПО: входной и выходной интерфейсы для консоли, блок-схему программ.
3. Определитесь со структурой программы (прототипы, описание, вызов функций), разработайте код на языке объектно-ориентированного программирования С++, протестируйте и отладьте его.

Ответы на вопросы

1. Как вы понимаете, что такое класс и объект? Приведите примеры.

**Класс (class)** - это абстрактный тип данных, заданный программистом. Класс описывает структуру своих объектов (поля) и их поведение (методы). Класс представляет модель реального объекта в виде данных и функций для работы с ними. Данные класса называются полями, а функции класса методами.

1. Каков синтаксис и смысл определения класса? Как называются составные части класса?

Общий вид синтаксиса класса**:**

*// имя должно быть уникальным среди имен других классов, определенных в вашей программе.*

**сlass <**имя\_ класса**>**

**{**

**private:**

<сокрытые поля и методы класса>

**public:**

<общедоступные поля и методыкласса**>**

**} [**список объектов]**;**

**Смысл определения класса**: фактически определение класса задает внутреннюю структуру объектов класса, т.е. из каких полей они состоят и какими методами пользуется объект класса при работе с этими полями. При определении класса не происходит выделения физической памяти. Память отводится при объявлении (создании) объекта.

Поля и методы класса называют **членами** или **элементами класса**.

1. Что такое идентификаторы доступа к элементам класса? Дайте определения.

Модификаторы доступа — это ключевые слова, с помощью которых осуществляется управление сокрытием данных в классе. Простыми словами, модификаторы доступа задают уровень доступности членам класса, ограничивая область их видимости.

**Разграничение доступ к полям и методам** класса можно регулировать с помощью идентификаторов доступа **private** и **public**. Элементы, описанные после служебного слова **private,** доступнытолько внутри класса. Этот вид доступа принят в классе по умолчанию (т.е. все элементы класса – данные и методы будут скрыты по умолчанию). К закрытым элементам класса можно получить доступ только с помощью методов самого класса. Элементы, описанные после служебного слова **public,** доступныи за пределами класса. Действие любого спецификатора распространяется до следующего спецификатора или до конца класса.

1. Что такое инкапсуляция?

**Инкапсуляция** – это механизм, который объединяет данные и код, манипулирующий с этими данными, а также защищает и то и другое от внешнего вмешательства или неправильного использования. Позволяет скрыть конкретную реализацию класса, облегчая отладку и модификацию программ.

1. Где в программе с классами можно создавать объекты? Внутри функций, в куче (с использованием оператора “new”), в стеке (как часть массива или коллекции), в качестве члена другого класса.
2. Каков смысл объявления объектов?

При определении класса мы не создали объекты. Определение класса лишь задает структуру будущего объекта. Все операции программа производит с объектами. Объявление объекта похоже на определение переменной, оно означает выделение памяти, необходимой для хранения объекта.

1. В какой момент метод готов для вызова объектом? Когда объект этого класса создан и находится в корректном состоянии.
2. Как называются методы, встроенные в структуру класса? Членами или элементами класса.
3. Как определяется метод, если внутри класса записан только его заголовок, сам метод определен вне класса? Это называется "разделением объявления и определения метода". В языке C++ это делается для того, чтобы вынести реализацию метода за пределы заголовка класса. Такой подход может быть полезен для улучшения структурированности кода, уменьшения сложности заголовков и обеспечения более гибкой организации кода.

class MyClass {

public:

// Заголовок метода

void myMethod();

};

// Реализация метода

void MyClass::myMethod() {

std::cout << "Implementation of myMethod" << std::endl;

}

1. Если в классе два поля данных и два объекта, сколько полей принадлежит каждому объекту? Совпадет ли имена и значения этих полей для объектов? Если в классе определены два поля данных, то каждому объекту этого класса будут принадлежать свои собственные два экземпляра этих полей. Каждый объект содержит свой собственный набор данных, который может иметь разные значения. Имена полей могут совпадать, но их значения будут независимыми для каждого объекта.
2. Тиражируются ли методы класса? Методы класса не тиражируются, то есть всегда хранятся в единственном экземпляре.
3. Как в программе написать доступ к открытым и закрытым полям класса?

Доступ к открытым полям объекта класса **напрямую** через операцию (точка):

part1.ModelNumber=2;

part1. ComponentNumber =5;

Чтобы обратиться к полю, надо указать имя объекта класса, которому принадлежит поле, а затем через точку – имя поля.

Доступ к закрытым полям объекта класса **только через методы** объекта:

part1.ShowCost();

Чтобы вызвать метод, надо указать имя объекта класса, для которого будет вызван метод, а затем через точку – имя метода.