**Ministry of education and science of the Kyrgyz Republic**

**Kyrgyz State Technical University named after I.Razzakov**

**Faculty of Information Technologies**

**Department of Software of Computer Systems**

**Major: 710400 «Software Engineering»**

Report

Discipline: «**Object-Oriented Design**»

Software requirements

Done by**:** student of the SE(eng)- 1- 21

Muratbek kyzy Eliza

Checked by: Musabaev E.B.

Bishkek 2024

**Task1**

**Class Aeroflot**

class Aeroflot

{

public:

short int NumOfFlight;

string DepartP; //Пункт отправления

string DestinP; //Пункт назначения

string DestinT;//Время прибытия

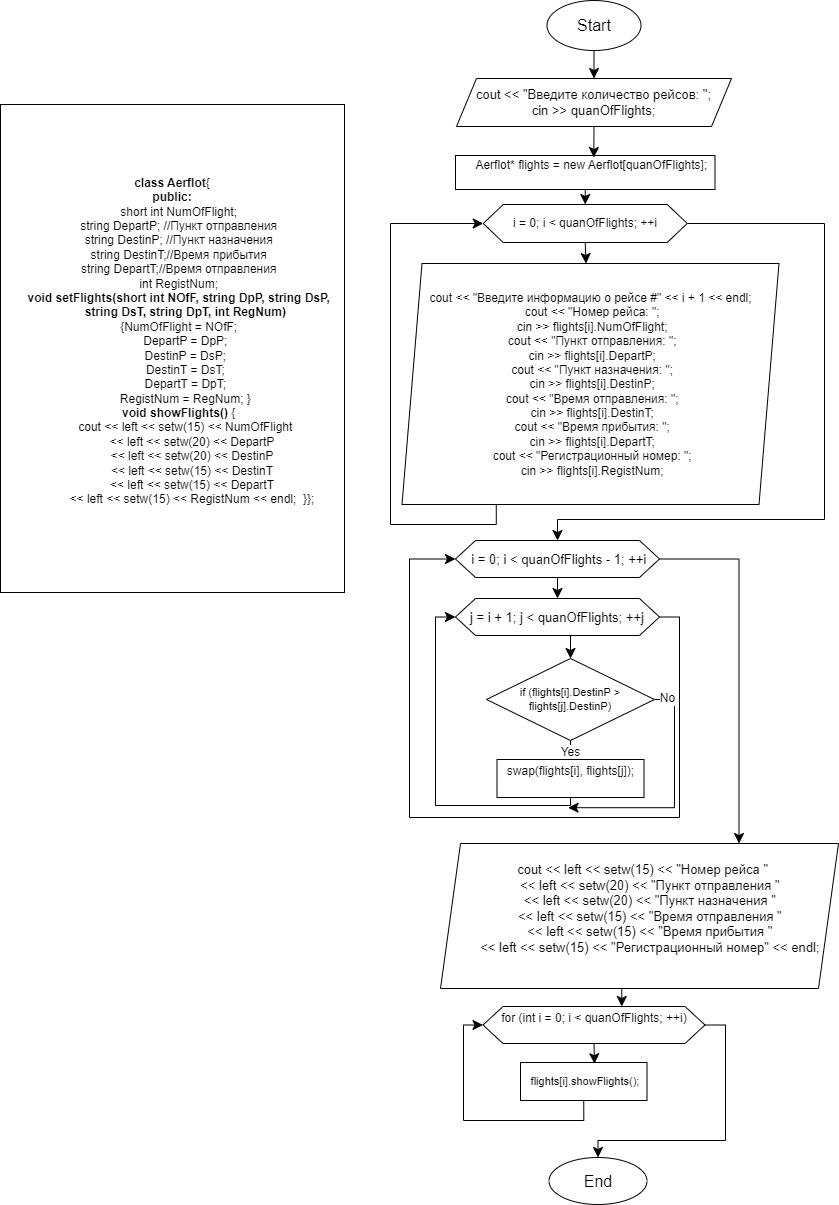
string DepartT;//Время отправления

int RegistNum;

**Functional requirements**

1. Ability to input values for the class Aeroflot objects from the keyboard(Enter details such as flight number, departure point, destination point, arrival time, departure time, and registration section.)
2. Sorting records in the table in alphabetical order by the destination point.
3. Displaying values of the class objects' fields on the console in the form of a flight table.

**Flowchart**

****

**Task 2**

**Class Students**

class Student

{

public:

string Surname;

string Name;

string DateOfBirth;

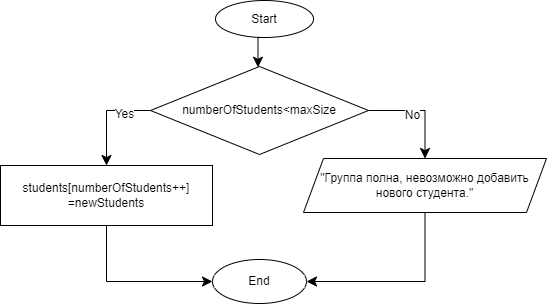
string PhoneNumber;}

**Functional requirements**

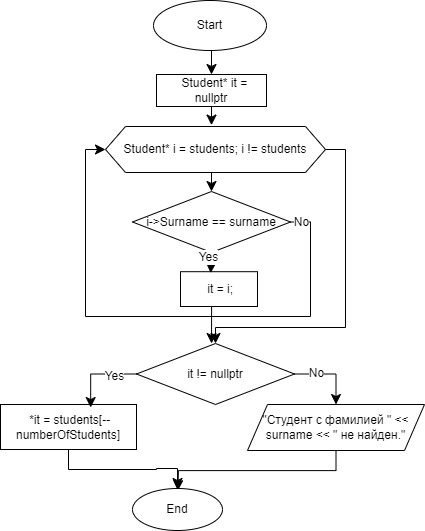
1. The system should handle a variable number of student records.
2. Users should be able to add or remove student records dynamically.
3. Users should be able to search for a student by various attributes, such as last name, date of birth, or phone number.
4. The system should allow users to sort the student records based on different fields, such as last name.

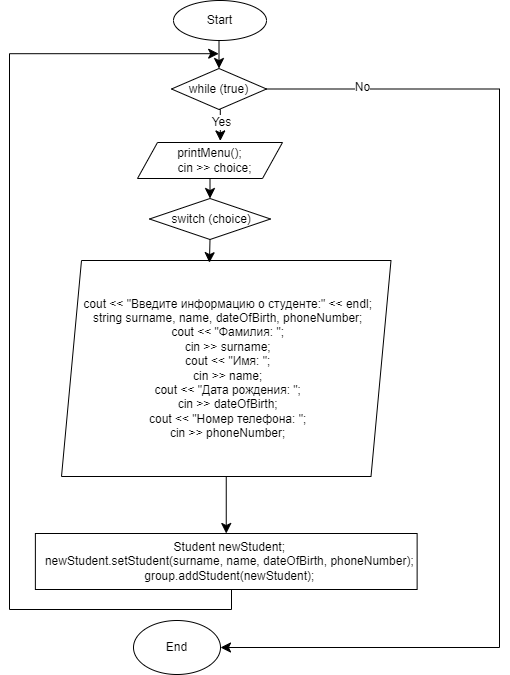
**Flowchart:**

**AddStudent**

****

**RemoveStudent**





**Task 3**

class Wagon {

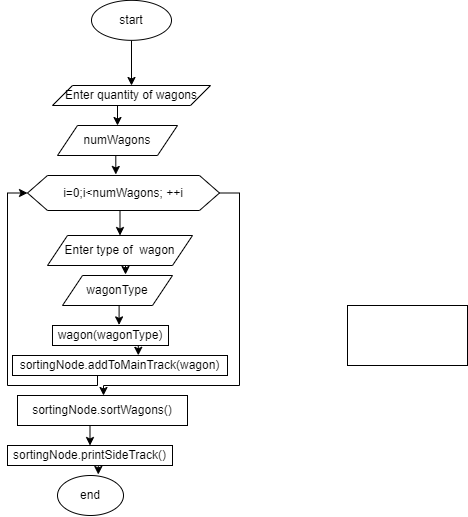
public:

int type; // Тип вагона: 0 или 1

Wagon(int t) : type(t) {}

};

1. Accept wagon types (0 or 1) from either file or keyboard.
2. Simulate a T-shaped sorting node with a main track and a side track.
3. Allow adding wagons to the main track, notifying the user.
4. Sort wagons: type 0 to side track, type 1 continues on the main track.
5. Output the final state of the side track.



**Nonfunctional requirements**

1. Response Time: The program should quickly calculate results,
2. Availability-is the system available only when connected to the internet or does it also work offline
3. Usability-The interface must be simple and understandable
4. OS Windows 7 and higher
5. Processor intel core i5 and higher

**Answers to questions:**

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

**Класс**-некоторое описание характеристик

**Объект**-Конкретный экземпляр у которого каждая характеристика имеет свое значение

**Пример:**

Класс: Человек{

Имя

Фамилия

Возраст

}

Объект{

Анна

Джонс

17}

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

**сlass <**имя\_ класса**>**

**{**

**private:**

<сокрытые поля и методы класса>

**public:**

<общедоступные поля и методыкласса**>**

**} [**список объектов]**;**

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

**public (открытый):** Члены класса с этим идентификатором доступа могут быть свободно использованы из любого места программы.

**private (закрытый):** Члены класса с этим идентификатором доступа доступны только внутри того же класса.

**protected (защищенный):** Члены класса с этим идентификатором доступа доступны внутри класса и его наследников.

Идентификаторы доступа управляют тем, какие члены класса видны и доступны из различных частей программы, обеспечивая контроль доступа и уровень инкапсуляции.

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?**

В главной функции main

Внутри других функций

1. **Каков смысл объявления объектов?**

При определении класса не происходит выделения физической памяти. Память отводится при объявлении (создании) объекта

При объявлении объекта резервируется память под его данные. Это включает в себя выделение места для всех полей класса, которые определены в его описании

Для дальнейшей работы с каждым объектом индивидуально

1. **В какой момент метод готов для вызова объектом?**

Когда объект создается и его конструктор завершен, методы этого объекта считаются готовыми для вызова.

1. **Как называются методы, встроенные в структуру класса?**

Методы класса

1. **Как определяется метод, если внутри класса записан только его заголовок, сам метод определен вне класса?**

class Class {

public:

void Task();

};

**void Class :: Task()** {

Реализация метода}

1. **Если в классе два поля данных и два объекта, сколько полей принадлежит каждому объекту? Совпадет ли имена и значения этих полей для объектов?**

Каждому объекту принадлежат свои уникальные экземпляры полей данных.

Имена полей могут совпадать у разных объектов, но значения будут разными, так как каждый объект имеет свои собственные значения полей.

1. **Тиражируются ли методы класса?**

Да. Методы класса действуют в контексте конкретного экземпляра объекта, и их вызов тиражируется для каждого объекта отдельно

1. **Как в программе написать доступ к открытым и закрытым полям класса?**

Для открытых полей можно обращаться напрямую.

Для закрытых полей можно использовать открытые методы класса (геттеры и сеттеры) или другие открытые методы, предоставляемые классом.

