**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

Rami Alshalabi

Checked by: Musabaev E.B.

Bishkek 2024

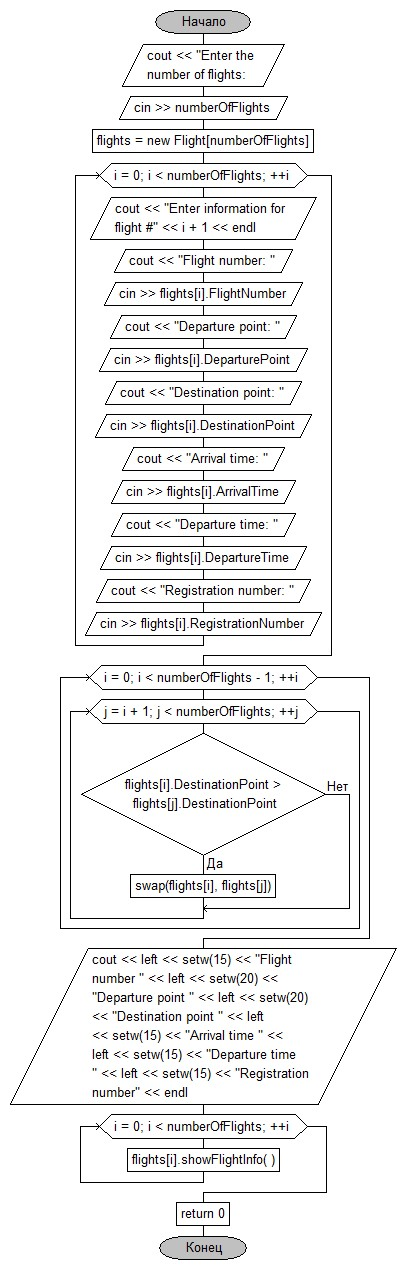
**Task1**

**Class Flight**

class Flight  
{  
public:  
 short int FlightNumber;  
 string DeparturePoint;   
 string DestinationPoint;   
 string ArrivalTime;   
 string DepartureTime;   
 int RegistrationNumber;

**Functional requirements**

1. Ability to input values for the class Flight 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**

class Flight  
{  
public:  
 short int FlightNumber;  
 string DeparturePoint;   
 string DestinationPoint;   
 string ArrivalTime;   
 string DepartureTime;   
 int RegistrationNumber;   
  
 void setFlightInfo(short int number, string departurePoint, string destinationPoint,  
 string arrivalTime, string departureTime, int regNumber)  
 {  
 FlightNumber = number;  
 DeparturePoint = departurePoint;  
 DestinationPoint = destinationPoint;  
 ArrivalTime = arrivalTime;  
 DepartureTime = departureTime;  
 RegistrationNumber = regNumber;  
 }  
  
 void showFlightInfo()  
 {  
 cout << left << setw(15) << FlightNumber  
 << left << setw(20) << DeparturePoint  
 << left << setw(20) << DestinationPoint  
 << left << setw(15) << ArrivalTime  
 << left << setw(15) << DepartureTime  
 << left << setw(15) << RegistrationNumber << endl;  
 }  
};

**Task 2**

**Class Students**

class Student

{

public:

string Surname;

string Name;

string DateOfBirth;

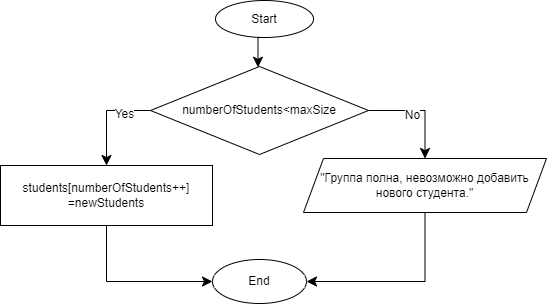
string PhoneNumber;}

**Functional requirements**

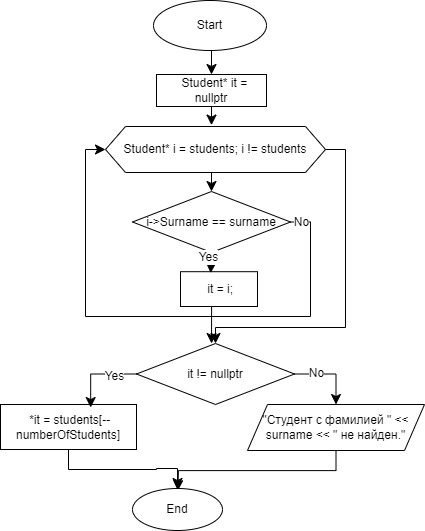
1. The system should handle a variable number of student records.
2. Users should be able to add or remove student records dynamically.
3. Users should be able to search for a student by various attributes, such as last name, date of birth, or phone number.
4. The system should allow users to sort the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

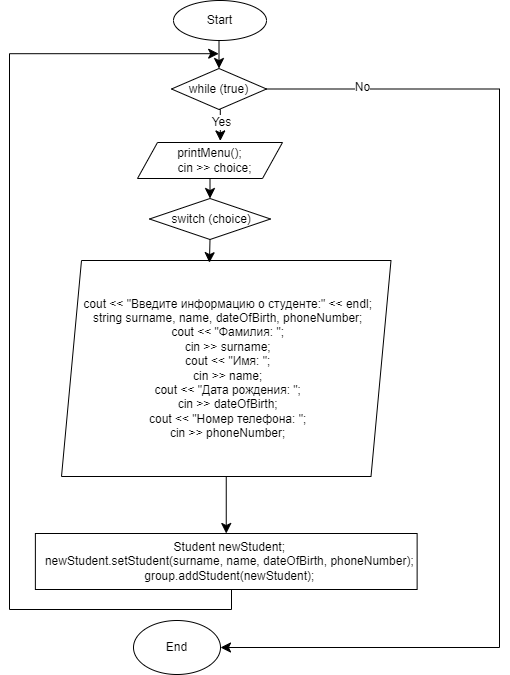
**Flowchart:**

**AddStudent**

****

**RemoveStudent**





**Task 3**

class Wagon {

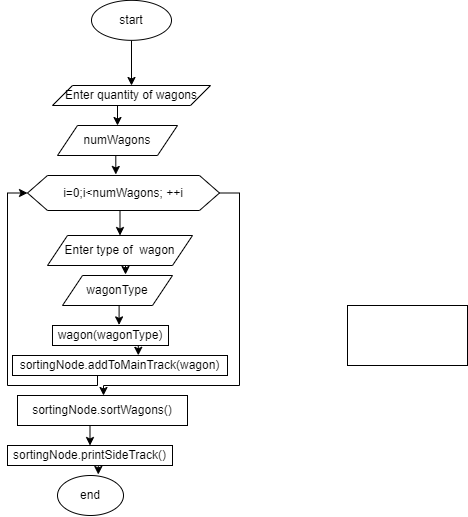
public:

int type; // type: 0 or 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. **How do you understand the concepts of class and object? Provide examples.**
   * Class: A description of characteristics.
   * Object: A specific instance where each characteristic has its own value.Example:

Class: Person { Name Surname Age } Object: { Anna Johns 17 }

1. **What is the syntax and meaning of class definition? What are the components of a class called?**

class ClassName { private: // private fields and methods of the class public: // public fields and methods of the class } [object list];

**Components of a class: private and public members.**

1. **What are access specifiers in a class? Provide definitions.**
   * Public: Members with this access specifier can be freely used from any part of the program.
   * Private: Members with this access specifier are only accessible within the same class.
   * Protected: Members with this access specifier are accessible within the class and its derived classes. Access specifiers control the visibility and accessibility of class members from different parts of the program, ensuring access control and encapsulation.
2. **What is encapsulation?** Encapsulation is a mechanism that combines data and the code that manipulates that data, protecting both from external interference or misuse. It hides the specific implementation of a class, making debugging and program modification easier.
3. **Where in a program with classes can you create objects?**
   * In the main function (main)
   * Inside other functions
4. **What is the significance of declaring objects? When a class is defined, no physical** memory is allocated. Memory is reserved when an object is declared (created). Object declaration allocates memory for its data, including space for all class fields defined in its description, facilitating individual interaction with each object.
5. **At what point is a method ready for invocation by an object? When an object is** created, and its constructor is completed, the methods of that object are considered ready for invocation.
6. **What are methods built into the structure of a class called?** Class methods.
7. **How is a method defined if only its declaration is inside the class, and the method itself is defined outside the class?**

class Class { public: void Task(); }; void Class::Task() { // Method implementation }

1. **If a class has two data fields and two objects, how many fields belong to each object?** Will the names and values of these fields match for objects? Each object has its unique instances of data fields. Names of fields may coincide across different objects, but values will be different since each object has its own set of field values.
2. **Are class methods overloaded? Yes, class methods are overloaded. They operate in** the context of a specific object instance, and their invocation is specific to each object.
3. **How can you access public and private class fields in a program?**
   * For public fields, you can directly access them.
   * For private fields, you can use public methods of the class (getters and setters) or other public methods provided by the class.

**Top of Form**