**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

**Task:**

class Tatra;

class Kamaz {

private:

string name;

int speed;

public:

Kamaz(string n, int s) {

name = n;

speed = s;

}

friend int FrCreater(Kamaz& k, Tatra& t);

};

class Tatra {

private:

string name;

int speed;

public:

Tatra(string n, int s) {

name = n;

speed = s;

}

friend int FrCreater(Kamaz& k, Tatra& t);

};

int FrCreater(Kamaz& k, Tatra& t) {

if (k.speed > t.speed) return 1;

else if (k.speed == t.speed) return 0;

else return -1;

}

**Functional requirements**

1. The program should allow the user to input data for Kamaz and Tatra.
2. For each truck, the user should provide the name and speed.
3. The program should compare the speeds of Kamaz and Tatra trucks on different types of terrains: plain, mountain, desert.
4. The comparison should be displayed as a table showing the speeds of both trucks on each terrain.
5. The program should determine the winner based on the comparison of speeds.
6. The comparison should be done using a function called **FrCreater()**, which returns:
   * 1 if the Kamaz truck moves faster than the Tatra truck.
   * 0 if their speeds are equal.
   * -1 if the Kamaz truck moves slower than the Tatra truck.
7. The **FrCreater()** function should be made a friend function of the **Kamaz** class and a friend function of both **Kamaz** and **Tatra** classes.
8. The **Tatra** class should be made a friend class of the **Kamaz** class.
9. The program should display the result of the comparison as -1, 0, or 1 along with text describing the meaning of these values.

**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. **Что такое дружественная функция, для чего она нужна?**

**Дружественная функция**—это функция, не являющаяся членом класса, но обеспечивающая доступ ко всем членам класса, для которого она дружественна. Независима от класса **дружественная функция** находится вне класса.

1. **Как она объявляется, каков её синтаксис?**

Но она объявляется внутри класса как обычная функция и перед ней указывается ключевое слово **friend**. Вне класса перед описанием дружественной функции это ключевое слово не ставится

1. **Через что дружественная функция может получить доступ к закрытому полю класса?**

через ключевое слово **friend**

1. **Какой вариант вызова ДФ правильный:**

a.show() или show(a)? правильно будет : show(a)

1. **Наследуется ли дружественная функция?**

Дружественная функция не наследуется

1. **Может ли дружественная функция быть дружественной более чем к одному классу?**

Дружественная функция может быть дружественной более чем к одному классу

1. **Для решения каких задачах есть смысл использовать функцию, дружественную двум и более классам?**

Дружественная функция полезна тогда, когда у двух и более разных классов имеется нечто общее, с чем необходимо произвести какие-то действия, например, сравнение или арифметические действия закрытых полей объектов, принадлежащих разным классам.

1. **Когда нужно использовать дружественные классы?**

Часто дружественные классы используются тогда, когда классу необходим доступ к закрытым данным другого класса.

1. **Какому принципу ООП не соответствуют дружественные функции?**

они не соответствуют принципу ограничения доступа к данным, в соответствии с которым только методы класса могут иметь доступ к сокрытым данным класса. Множество дружественных функций фактически стирают границы между классами, поэтому они должны быть разбросаны по программе и встречаться как можно реже.

1. **Как можно ограничить использование ДФ?**

они должны быть разбросаны по программе и встречаться как можно реже.

1. **В каких крайних случаях нужно использовать ДФ?**

когда нет других адекватных способов обеспечить доступ к закрытым членам класса