**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

Imerov Adilhan

Checked by: Musabaev E.B.

Bishkek 2024

Functional Requirements:

Task 1:

1. The **Int** class should enable four arithmetic operations (+, -, \*, /) with integers.
2. The **Int** class should handle arithmetic overflow by throwing an exception if the result exceeds the boundaries of the **int** type.
3. Overload unary increment operations (postfix and prefix forms) for objects of the **Int** class using internal operator functions.
4. Computations should use the **long double** type to facilitate overflow checking.
5. Utilize the **this** pointer to describe unary operations.

Task 2:

1. The **Fraction** class should have a private integer field to store the fractional value.
2. Overload arithmetic operators for addition, subtraction, multiplication, and division to work with both **Fraction** class objects and numbers.
3. Operator overloading should allow performing arithmetic operations with fractions of different types (e.g., 3/4 + 2/5) and mixed numbers (e.g., 1/2 + 4 or 2 \* 5/6).
4. Overload unary increment operation in prefix or postfix form for incrementing fractions.

Task 3:

1. Overload the **rect\_area()** method in two ways: one method takes both dimensions of the figure (for a rectangle), and the other takes only one dimension (for a square).
2. When called with two arguments, the method should calculate the area of the rectangle, and with one argument, it should calculate the area of the square.
3. If the method is called with an incorrect number of arguments, generate an error message.

Non-functional Requirements:

1. Response Time: The program should quickly calculate results.
2. Availability: The system should work both online and offline.
3. Usability: The interface must be simple and understandable.
4. OS Compatibility: The program should be compatible with Windows 7 and higher.
5. Processor Requirement: The program should run on Intel Core i5 and higher processors.

Answers to Questions:

1. What is polymorphism?
   * Polymorphism is having one interface with multiple implementations.
2. What is operator overloading?
   * Operator overloading is the ability to make the compiler perform standard operations on non-standard (user-defined) data types, optimizing code.
3. What is operator overriding?
   * Operator overriding is the ability to change or extend the standard behavior of operators in programming languages for user-defined types (classes).
4. What is the syntax of an operator function?
5. Which operators can be overloaded, and which cannot?
   * Overloadable operators include arithmetic, logical, bitwise, and comparison operators. Unoverloadable operators include scope resolution (::), member access (.), member pointer access (.\*), and the ternary operator (?:).
6. What is the relationship between the number of arguments in an operator function and the number of operands? Explain this relationship.
   * An overloaded operation always requires one less argument than the number of operands. This is because one of the objects is the object calling the function.
7. How do overloaded operators make code more readable?
   * Overloaded operators allow simplifying code, making it more intuitive. For example, replacing a complex operation in dotted notation:

Cars.car3 = Cars.car2 + Cars.car2;

with a simpler form:

car3 = car1 + car2;

Top of Form