**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

**Functional requirements**

**Task1:**

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

**Task2:**

1. The class Fraction should have one private integer field to store the fractional value.
2. Arithmetic operators for addition, subtraction, multiplication, and division should be overloaded 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. Unary increment operation in prefix or postfix form should be overloaded for incrementing fractions.

**Task3:**

1. The rect\_area() method should be overloaded 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, an error message should be generated.

**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 полиморфизм?**

Полиморфизм – это один интерфейс с множеством реализаций

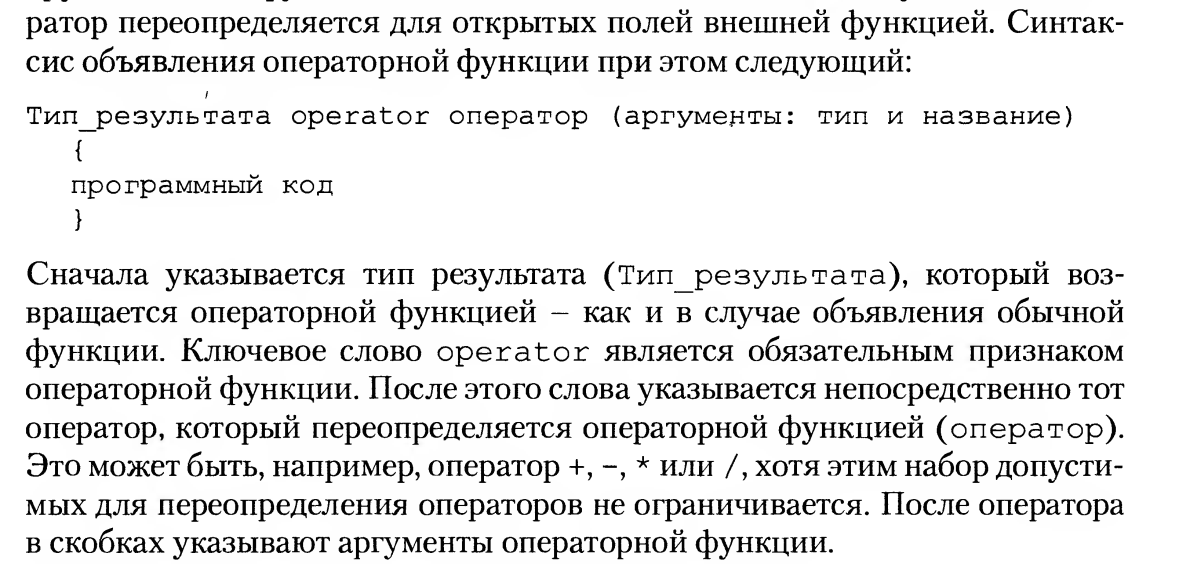
1. **Что такое перегрузка операторов?**

Перегрузка операторов – это возможность заставить компилятор осуществлять стандартные операции над нестандартными (пользовательскими) типами дан-ных, тем самым оптимизировать код программы.

1. **Что такое переопределение операторов?**

Переопределение операторов — это возможность изменять или расширять стандартное поведение операторов в языках программирования для пользовательских типов (классов). Это означает, что разработчики могут задать, каким образом операторы, такие как +, -, \*, /, и другие, будут работать с экземплярами их классов. Таким образом, можно улучшить читаемость кода, делая его более интуитивно понятным

1. **Каков синтаксис операторной функции?**

****

1. **Какие операторы можно перегружать, а какие нельзя?**

**Можно перегружать следующие операторы:**+ - \* / % ^ & | ~ ! = < >  
+= -= \*= /= %= ^= &= |=  
<< >> >>= <<= == != <= >=  
&& || ++ -- ->\* , -> [] ()  
new new[] delete delete[]

**Нельзя перегружать операторы:**

* + - :: (разрешение области видимости)
* .  (доступ к членам класса)
* .\* (выбор члена через указатель на член)
* ?: тернарный оператор

1. **Какова зависимость между количеством аргументов в операторной функции и количеством операндов? Объясните эту зависимость.**

Перегруженной операции всегда требуется количество аргументов на один меньше, чем количество операндов. Это связано с тем, что один из объектов является объектом, вызывающим функцию.

1. **Каким образом перегруженные операции позволяют вид программного кода сделать более читабельным?**

**Например,** код операции сложения объектов в виде в точечном формате:

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

операторная функция дает возможность заменить на более простой:

car3= car1+ car2;