**Лабораторная работа №5 «ООП C++. Перегрузка операторов»**

**В каждом задании продемонстрировать работу программ!**

***Вариант 1.***

*Задание 1 Унарная операция*

Создать класс целых чисел. Определить оператор ++, как функцию-член и -- как дружественную функцию.

*Задание 2. Бинарная операция*

Создать объект очередь с перегруженными +, +=, добавление элемента в очередь и сложение очередей, -- для извлечения из очереди, - для вычитания очередей.

***Вариант 2.***

*Задание 1 Унарная операция*

Создать класс целых чисел. Определить оператор --, как функцию-член и ++ как дружественную функцию.

*Задание 2. Бинарная операция*

Создать класс координат. Определить оператор +, как функцию-член и - как дружественную функцию. Сложить и вычесть координаты с друг другом и с числом. Присвоить координаты, сравнить координаты (==, !=).

***Вариант 3.***

*Задание 1 Унарная операция*

Создать объект стек, перегрузив ++ и --. (Как постфиксными так префиксными). ++ Добавляет элемент в стек. -- извлекает элемент из стека.

*Задание 2. Бинарная операция*

Создать класс целых чисел. Определить оператор -, как функцию-член и + как дружественную функцию.

***Вариант 4.***

*Задание 1. Унарная операция*

Создать класс вещественных чисел (double). Определить оператор ++, как функцию-член и -- как дружественную функцию.

*Задание 2. Бинарная операция*

Создать объект типа стек. Перегрузить оператор + как функцию член и \* как дружественную функцию. + складывает элемент в новый стек, \* умножает верхушку стека на параметр. Стеки можно присваивать, проверять на равенство == или !=, вводить и выводить в поток, добавлять += элемент в стек.

***Вариант 5.***

*Задание 1 Унарная операция*

Создать объект типа стек. Перегрузить оператор ++ как функцию член и -- как дружественную функцию. (Как постфиксными так префиксными). ++ добавляет элемент новый в стек, -- удаляет верхушку стека. Оператор ! проверяет стек на пустоту.

*Задание 2. Бинарная операция*

Создать класс вещественных чисел (double). Определить оператор -, как функцию-член и + как дружественную функцию.

***Вариант 6.***

*Задание 1 Унарная операция*

Создать объект - связный двунаправленный список, с перегруженными унарными операциями ++, --, как движение по списку. (Как постфиксными так префиксными).

*Задание 2 Бинарная операция*

Создать класс целых чисел. Определить оператор +, как функцию-член и - как дружественную функцию.

***Вариант 7.***

*Задание 1. Унарная операция*

Создать класс вещественных чисел. Определить оператор ++, как функцию-член и -- как дружественную функцию.

*Задание 2. Бинарная операция*

Создать объект очередь с перегруженными +, +=, добавление элемента в очередь и сложение очередей, -- для извлечения из очереди, - для вычитания очередей.

***Вариант 8.***

*Задание 1. Унарная операция*

Создать класс - координаты с унарным ++ и --, -. ++ и -- постфиксная и префиксная. - меняет знак у обеих координат. ++ как функция-член, -- как дружественная функция.

*Задание 2. Бинарная операция*

Создать класс целых чисел (long). Определить оператор -, как функцию-член и + как дружественную функцию. Оператор присвоения, и сравнений.

***Вариант 9.***

*Задание 1. Унарная операция*

Создать объект типа очередь. Перегрузить оператор ++ как функцию член и -- как дружественную функцию. (Как постфиксными так префиксными). ++ добавляет элемент в очередь (пустой элемент, например int i=0), -- вытаскивает элемент из очереди. Оператор ! проверяет очередь на пустоту.

*Задание 2. Бинарная операция*

Создать класс целых чисел (long). Определить оператор +, как функцию-член и - как дружественную функцию.

***Вариант 10.***

*Задание 1. Унарная операция*

Создать класс вещественных чисел (double). Определить оператор --, как функцию-член и ++ как дружественную функцию.

*Задание 2. Бинарная операция*

Создать объект типа очередь. Перегрузить оператор + как функцию член и \* как дружественную функцию. + добавляет элемент в очередь, \* умножает элемент в очереди. Вытаскивает элемент из очереди --. Очереди можно присваивать, проверять на равенство == или !=, вводить и выводить в поток, добавлять += элемент в очередь.

***Вариант 11.***

*Задание 1. Унарная операция*

Создать класс целых чисел (long). Определить оператор ++, как функцию-член и -- как дружественную функцию.

*Задание 2. Бинарная операция*

Создать объект - однонаправленный список, в котором определены операции, + - добавляет в конец списка, += добавляет в этот же список в конец списка. - удаляет указанный элемент из списка (номер элемента через параметр), = - присвоение списков, сравнение списков ==, !=, >, <, >=, <=, [] получение элемента списка, ++ - устанавливает указатель на следующий элемент. () выдать подсписок от первого до второго элемента.

***Вариант 12.***

*Задание 1. Унарная операция*

Создать класс целых чисел (long). Определить оператор --, как функцию-член и ++ как дружественную функцию.

*Задание 2. Бинарная операция*

Создать объект стек, с перегруженными операциями +, -, =, +=, и для вытаскивания из стека --. () - выдает под-стек.

***Вариант 13.***

*Задание 1. Унарная операция*

Создать объект - очередь с перегруженными операциями ++ как функциями-членами, -- как дружественными функциями. (Как постфиксными так префиксными).

*Задание 2. Бинарная операция*

Создать класс вещественных чисел (double). Определить оператор +, как функцию-член и - как дружественную функцию.

***Вариант 14.***

*Задание 1. Унарная операция*

Создать объект - однонаправленный список, в котором определены операции, ++ - добавляет в конец списка, -- удаляет элемент из списка. (Как постфиксными так префиксными).

*Задание 2. Бинарная операция*

Создать класс вещественных чисел. Определить оператор -, как функцию-член и + как дружественную функцию.

***Вариант 15.***

*Задание 1. Унарная операция*

Создать класс целых чисел (long). Определить оператор ++, как функцию-член и -- как дружественную функцию.

*Задание 2. Бинарная операция*

Создать объект типа стек. Перегрузить оператор ++ как функцию член и -- как дружественную функцию. (Как постфиксными так префиксными). ++ добавляет элемент новый в стек, -- удаляет верхушку стека. Оператор ! проверяет стек на пустоту.

***Вариант 16.***

*Задание 1. Унарная операция*

Создать объект стек, перегрузив ++ и --. (Как постфиксными так префиксными). ++ Добавляет элемент в стек. -- извлекает элемент из стека.

*Задание 2. Бинарная операция*

Создать класс целых чисел (long). Определить оператор +, как функцию-член и \* как дружественную функцию.

***Вариант 17.***

*Задание 1. Унарная операция*

Создать класс - координаты с унарным ++ и --, -. ++ и -- постфиксная и префиксная. - меняет знак у обеих координат. ++ как функция-член, -- как дружественная функция.

*Задание 2. Бинарная операция*

Создать класс вещественных чисел (double). Определить оператор\*, как функцию-член и + как дружественную функцию.

***Вариант 18.***

*Задание 1. Унарная операция*

Создать объект стек, перегрузив ++ и --. (Как постфиксными так префиксными). ++ Добавляет элемент в стек. -- извлекает элемент из стека.

*Задание 2. Бинарная операция*

Создать класс целых чисел. Определить оператор -, как функцию-член и + как дружественную функцию.

***Вариант 19.***

*Задание 1. Унарная операция*

Создать класс вещественных чисел. Определить оператор ++, как функцию-член и - как дружественную функцию.

*Задание 2. Бинарная операция*

Создать объект типа очередь. Перегрузить оператор + как функцию член и \* как дружественную функцию. + добавляет элемент в очередь, \* умножает элемент в очереди. Вытаскивает элемент из очереди --. Очереди можно присваивать, проверять на равенство == или !=, вводить и выводить в поток, добавлять += элемент в очередь.

***Вариант 20.***

*Задание 1. Унарная операция*

Создать класс вещественных чисел. Определить оператор ++, как функцию-член и \* как дружественную функцию.

*Задание 2. Бинарная операция*

Создать объект - однонаправленный список, в котором определены операции, + - добавляет в конец списка, += добавляет в этот же список в конец списка. - удаляет указанный элемент из списка (номер элемента через параметр), = - присвоение списков, сравнение списков ==, !=, >, <, >=, <=, [] получение элемента списка, ++ - устанавливает указатель на следующий элемент. () выдать подсписок от первого до второго элемента.