

Задачи на работу с классами

1. Класс Интервал

Создать класс Interval, представляющий собой отрезок, задаваемый двумя числами на прямой – начало и конец интервала (типа float).

- 1) Написать конструкторы класса
 - a) Конструктор по умолчанию, задающий начало и конец интервала равными нулю
 - b) Конструктор с параметрами, устанавливающий значения начала и конца интервала (в случае если передаваемый конец интервала меньше начала, поменять их местами)
 - c) Конструктор копирования
- 2) Перегрузить оператор присваивания
- 3) Реализовать методы класса
 - a) Для нахождения длины интервала
 - b) Установка и получения значений начала и конца интервала
 - c) Move(float dist); – функция сдвигает начало и конец интервала на расстояние dist с сохранением его длины
 - d) Expand(float scale) - увеличивает длину интервала в scale раз с сохранением его начала
 - e) float Mid() - возвращает точку середину интервала
 - f) bool IsEmpty() – возвращает true в случае, если длина интервала равна 0 (начало совпадает с концом)
- 4) Перегрузить операторы
 - a) & – пересечение двух интервалов. Возвращает интервал, содержащийся в пересечении интервалов, либо пустой интервал, если они не пересекаются.
 - b) | – объединение интервалов. Возвращает интервал, включающий первый и второй интервал, если между ними нет промежутка. В противном случае – пустой интервал.
 - c) void operator+=(float dist) – то, что делает функция Move.
 - d) void operator*=(float scale) – то, что делает функция Expand.
 - e) Операторы записи в поток и чтения из потока. В поток выводить в виде строки "1.2 5.6" , где через пробел идут начало и конец интервала.

2. Класс Стек

Создать класс Stack, реализующий стек целых чисел (int). Основные два метода стека: Push() – поместить элемент в стек и Pop() – забрать элемент из стека и вернуть в функции. Элемент, помещенный в стек последним с помощью Push(), возвращается первым с помощью Pop(). Память под содержимое стека выделять динамически по мере роста стека.

- 1) Написать конструкторы класса
 - a) Конструктор по умолчанию, первоначально выделяющий память под 10 элементов
 - b) Конструктор копирования
- 2) Написать деструктор класса
- 3) Перегрузить оператор присваивания
- 4) Реализовать методы класса
 - 1) void Push(int element); – поместить элемент в стек
 - 2) int Pop(); – извлечь элемент из стека
 - 3) size_t GetCount(); – вернуть количество элементов в стеке

- 4) `int Top();` – возвращает элемент из стека, не извлекая его
- 5) `void Clear();` – очистить стек
- 5) Перегрузить операторы
 - a) `Stack& operator<<(int element)` – помещает элемент в стек (то же самое, что делает `Push`)
 - b) `Stack& operator>>(int& element)` – извлекает элемент из стека (то же самое, что делает `Pop`)

3. Класс Комплексное число

Создать класс `Complex`, реализующий комплексное число https://ru.wikipedia.org/wiki/Комплексное_число. Комплексное число состоит из двух слагаемых – действительной и мнимой частей: $Re + Im \cdot i$, где i – мнимая единица ($i^2 = -1$), Re , Im – действительная и мнимая части соответственно (`float`).

Сумма, разность, произведение и частное комплексных чисел вычисляются следующим образом:

$$(a + b \cdot i) + (c + d \cdot i) = (a + c) + (b + d) \cdot i$$

$$(a + b \cdot i) - (c + d \cdot i) = (a - c) + (b - d) \cdot i$$

$$(a + b \cdot i) \cdot (c + d \cdot i) = (a \cdot c - b \cdot d) + (b \cdot c + a \cdot d) \cdot i$$

$$(a + b \cdot i) / (c + d \cdot i) = (a \cdot c + b \cdot d) / (c^2 + d^2) + (b \cdot c - a \cdot d) / (c^2 + d^2) \cdot i$$

$$\text{Модуль } |a + b \cdot i| = \sqrt{a^2 + b^2}$$

Необходимо

- 1) Написать конструкторы класса
 - a) Конструктор по умолчанию, задающий нули для действительной (`Re`) и мнимой (`Im`) частей.
 - b) Конструктор с 2 параметрами, задающий действительную и мнимую части
 - c) Конструктор с 1 параметром, задающий действительную часть, мнимая часть устанавливается в нуль.
 - d) Конструктор копирования
- 2) Перегрузить оператор присваивания
- 3) Написать методы для установки и получения значений действительной частей
- 4) Написать метод, для вычисления модуля числа
- 5) Перегрузить арифметические операции: `+`, `-`, `*`, `/`
- 6) Перегрузить операторы чтения из потока и записи в поток. Например, число у которого $Re = 4$, $Im = 2$, выводить **"4+2i"**, число, у которого $Re = 3$, $Im = -5$, выводить как **"3-5i"**. В том же формате считывать из потока.