Лабораторная работа 15. Классы и объекты, перегрузка операторов

https://en.cppreference.com/w/cpp/language/operators

https://metanit.com/cpp/tutorial/5.14.php?ysclid=m4l57i4g40267868016

I. Доработать класс «Рациональное число» из лабораторной работы 15

1) Для класса «Рациональное число» перегрузить указанные операторы. Каждый оператор протестировать в основной программе (main).

Список операторов для перегрузки:

a) **+, -, *, /** для выполнения арифметических действий с рациональными числам

b) **==, !=, <, >** для сравнения рациональных чисел

c) **>>, <<** для ввода и вывода рациональных чисел через стандартные и файловые потоки

d) **унарный минус -** , должен менять знак у числителя на противоположный;

++ (префиксный), увеличивает значение числа на 1;

-- (постфиксный), уменьшает значение числа на 1;

оператор () без аргументов, должен возвращать модуль числа (менять знак числителя на +, если числитель отрицательный);

оператор [ ] должен возвращать числитель, если аргумент равен 1, или возвращать знаменатель, если аргумент равен 2; для других значений аргумента поведение оператора не определять

2) Написать программу, в которой ввести с клавиатуры массив из 5 рациональных чисел.

Вывести на экран и в файл (имя файла запросить у пользователя) следующие результаты:

a) сумму всех элементов массива

b) количество элементов, которые не равны ½

c) все элементы, большие чем 2/3

d) произведение первого и последнего элемента

e) минимальный элемент в массиве

f) элемент, который ближе всего к числу 5/11 и его индекс, использовать операторы –, () и сравнения

g) сумму всех числителей (использовать оператор [] вместо геттера)

h) новый массив рациональных чисел, элементы которого получены из исходного массива по правилу:

$b_i = \frac{-a_i+1}{a_i}$, где a_i – элементы исходного массива

i) в массиве b из задачи h) все элементы с четными индексами увеличить на 1, а элементы с нечетным индексами уменьшить на 1 (использовать перегруженные ++ и --), вывести полученный массив.

II. Класс Person

Создать класс Person, соответствующий личным данным человека:

- фамилия,
- имя,
- дата рождения (можно использовать time_t и структуру tm или отдельные числовые поля),
- пол (один символ)

Разместить все данные в закрытых полях, для доступа к ним создать геттеры и сеттеры..
При изменении данных о поле организовать проверку на корректность (ограничить допустимые значения символами ‘m’, ‘f’ или ‘м’, ‘ж’ …)

Определить конструкторы, функцию консольного ввода в диалоге с пользователем, перегрузить операторы сравнения и операторы ввода и вывода <<, >>

Создать функцию-член класса, возвращающий возраст человека – количество полных лет.

Написать программу, вводящую данные об указанном пользователем количестве людей в динамический массив. Отсортировать массив по полю возраст в порядке возрастания; использовать собственную реализацию любого алгоритма сортировки.
Вывести для каждого элемента отсортированного массива фамилию, инициалы имени, возраст и пол

III. Класс «точка на плоскости»

Создать класс «точка на плоскости»
Класс должен включать
- поля вещественного типа для хранения декартовых координат точки на плоскости
- функции доступа к полям (геттеры, сеттеры)
- конструкторы
- функции или операторы для консольного ввода и вывода
- функции или операторы для сравнения двух точек (проверка на равенство; большей считать точку, которая дальше от начала координат)
- функцию расчета расстояния между двумя такими точками

Протестировать функции класса на подходящих примерах