

Слущкий Никита. 053501. Лабораторная Работа №8. Реализация класса Set для double

В основе класса лежит работа с указателями. Мой Set представляет из себя двусвязный список (list), в котором каждый член (Member, который также выделен в отдельный класс) содержит 3 поля: value (непосредственно значение), *next (указатель на следующий после него элемент), *previous (указатель на предшествующий элемент) — поэтому двусвязный.

В классе присутствуют следующие приватные поля:

- *first — хранит адрес текущего первого элемента дека
- *last — соответственно хранит адрес последнего
- *_size — текущее количество элементов дека

Реализован конструктор, просто создающий новый Set.

Реализован деструктор, который вызывает метод Clear () и => удаляет все Member.

Реализованы следующие приватные и публичные (в зависимости от надобности) методы:

- Find () — линейно ищет значение по ключу и возвращает его порядковый номер (в случае отсутствия: -1). // Приватный
- Size () — возвращает количество элементов в текущем множестве
- Contains () — возвращает логическое значение в зависимости от того, присутствует ли элемент в множестве. Ссылается на внутренний приватный Find ()
- Add () — добавляет элемент в множество. В обычном массиве операция бы занимала 3N (поиск, копирование туда/сюда), тут же — N (только поиск. Вставка в конец за константу работает)
- Remove () — удаляет элемент в множестве. На проверку + на доступ уходит 2N (можно объединить и сделать N на самом деле). Удаление само по себе тратит константное количество операций.
- Clear () — очищает множество. Удаляет *first до тех пор, пока не удалит все Member. Если без очистки памяти от прошлых значений, то вообще за 3 операции можно сделать

Перегружены некоторые операторы:

- Оператор потокового вывода cout<<. По аналогии в JavaScript, где можно вывести сет (console.log(set)), он выводит размер и элементы в скобках
- Оператор доступа по индексу [] (как и в обычном массиве или векторе) // приватный. Для внутренней работы, так как оператор cout<< без него не хотел сам то же самое внутри себя делать