

Реализовать недостающие методы класса `DoubleLinkedList`:

Методы для корректного копирования и перемещения:

1. Конструктор копирования
2. Оператор копирующего присваивания (использовать идиому `copy-and-swap` <https://ru.wikipedia.org/wiki/Copy-and-swap>)
3. Конструктор перемещения
4. Оператор перемещающего присваивания

Методы, объявленные, но не реализованные в коде:

5. `insertTail(Node* x)` – вставка сформированного узла в хвост списка
6. `replaceNode(Node* x, int item)` – замена значения узла на новое
7. `deleteTail()` – удаление узла из хвоста списка
8. `deleteItem(const int item, bool all = false)` – удаление узла (или узлов) с указанным значением
9. `replaceItem(int itemOld, int itemNew, bool all = false)` – замена значения узла (или узлов) на новое

Методы и функции, указанные в презентации «Практика #1»:

10. Перегруженный оператор вывода `<<` (дружественная функция)
11. Перегруженный оператор сравнения `==` (метод класса)
12. Метод для добавления в конец исходного списка элементов списка-параметра