# Контрольная работа #1: «Связные списки»

# Общее в заданиях

1. Класс двусвязный список из целых чисел.
2. Класс должен иметь конструктор по умолчанию и деструктор.
3. Копирование и присваивание запретить.
4. Класс должен иметь метод **insert** для добавления элемента в «хвост» **или** в «голову» списка (порядок следования элементов в списке при этом будет соответствовать порядку их добавления **или** будет обратным; указывается в варианте задания).
5. В соответствии с вариантом задания должен быть реализован один из методов:
6. метод **removeMin**, удаляющий из списка элемент с наименьшим значением (если таких несколько, то удалить надо все)
7. метод **deleteMax**, удаляющий из списка элемент с наибольшим значением (если таких несколько, то удалить надо все)
8. метод **reverse**, переворачивающий список в обратном порядке
9. метод **replaceByAverage**, заменяющий каждую пару последовательных элементов исходного списка одним, содержащим среднее арифметическое этих элементов (если в списке нечетное число элементов, последний удаляется)
10. метод **replaceByMin**, заменяющий каждую пару последовательных элементов исходного списка одним из них, имеющим минимальное значение
11. метод **deleteLess**, удаляющий из списка все элементы, значения которых меньше, хотя бы одного стоящего ранее (просмотр списка начинать с «головы»)
12. метод **merge**, объединяющий два списка в один по правилу: первый элемент первого списка, первый элемент второго; второй элемент первого, второй второго и т. д. Хвост более длинного списка добавить в хвост объединенного. Второй список должен остаться «пустым»
13. метод **reverse**, переворачивающий список в обратном порядке

# Общие требования к реализации

* Перемещение элементов списка в другое место списка должно выполняться **только через изменение указателей.**
* Удаление элементов списка должно включать освобождение памяти.
* При тестировании класса должны учитываться особые случаи (пустые списки, отсутствие искомых элементов и т. п.).

# Примеры вариантов заданий

# Задание 1\_1

Создать класс, реализующий двусвязный список из целых чисел.

Порядок следования чисел в списке должен быть **обратным** порядку добавления элементов, возможны повторяющиеся элементы списка.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* **метод deleteMin**, удаляющий из списка элемент с наименьшим значением

(если таких несколько, то удалить надо все)

# Задание 1\_2

Создать класс, реализующий двусвязный список из целых чисел.

Порядок следования чисел в списке должен **соответствовать** порядку добавления элементов, возможны повторяющиеся элементы.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* **метод deleteMax**, удаляющий из списка элемент с наибольшим значением

(если таких несколько, то удалить надо все)

# Задание 2\_1

Создать класс, реализующий двусвязный список из целых чисел.

Порядок следования чисел в списке должен быть **обратным** порядку добавления элементов, возможны повторяющиеся элементы.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* **метод reverse**, переворачивающий список в обратном порядке.

# Задание 2\_2

Создать класс, реализующий двусвязный список из целых чисел.

Порядок следования чисел в списке должен **соответствовать** порядку добавления элементов, возможны повторяющиеся элементы.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* **метод reverse**, переворачивающий список в обратном порядке.

# Задание 3\_1

Создать класс, реализующий двусвязный список из целых чисел.

Порядок следования чисел в списке должен быть **обратным** порядку добавления элементов, возможны повторяющиеся элементы.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* метод **replaceByAverage,** заменяющий каждую пару последовательных элементов исходного списка одним, содержащим среднее арифметическое этих элементов (если в списке нечетное число элементов, последний удаляется)

# Задание 3\_2

Создать класс, реализующий двусвязный список из целых чисел.

Порядок следования чисел в списке должен **соответствовать** порядку добавления элементов, возможны повторяющиеся элементы.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* **метод replaceByAverage,** заменяющий каждую пару последовательных элементов исходного списка одним, содержащим среднее арифметическое этих элементов (если в списке нечетное число элементов, последний удаляется)

# Задание 4\_1

Создать класс, реализующий **двусвязный** список из целых чисел.

Порядок следования чисел в списке должен быть **обратным** порядку добавления элементов, возможны повторяющиеся элементы.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* **метод replaceByMin,** заменяющий каждую пару последовательных элементов исходного списка одним из них, имеющим минимальное значение (если в списке нечетное число элементов, последний удаляется)

# Задание 4\_2

Создать класс, реализующий **двусвязный** список из целых чисел.

Порядок следования чисел в списке должен соответствовать порядку добавления элементов, возможны повторяющиеся элементы.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* **метод replaceByMin,** заменяющий каждую пару последовательных элементов исходного списка одним из них, имеющим минимальное значение (если в списке нечетное число элементов, последний удаляется)

# Задание 5\_1

Создать класс, реализующий двусвязный список из целых чисел.

Порядок следования чисел в списке должен быть **обратным** порядку добавления элементов, возможны повторяющиеся элементы.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* метод **deleteLess**, удаляющий из списка все элементы, значения которых меньше, хотя бы одного стоящего ранее (просмотр списка начинать с головы).

# Задание 5\_2

Создать класс, реализующий двусвязный список из целых чисел.

Порядок следования чисел в списке должен **соответствовать** порядку добавления элементов, возможны повторяющиеся элементы списка.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* **метод deleteLess**, удаляющий из списка все элементы, значения которых меньше, хотя бы одного стоящего ранее (просмотр списка начинать с головы)

# Задание 6\_1

Создать класс, реализующий двусвязный список из целых чисел.

Порядок следования чисел в списке должен быть **обратным** порядку добавления элементов, возможны повторяющиеся элементы.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* метод **merge**, объединяющий два списка в один по правилу: первый элемент первого списка, первый элемент второго; второй элемент первого, второй второго и т. д. Хвост более длинного списка добавить в хвост объединенного. Второй список должен остаться «пустым».

# Задание 6\_2

Создать класс, реализующий двусвязный список из целых чисел.

Порядок следования чисел в списке должен **соответствовать** порядку добавления элементов, возможны повторяющиеся элементы списка.

Класс должен иметь:

* метод **insert** для добавления элемента в список
* метод **merge**, объединяющий два списка в один по правилу: первый элемент первого списка, первый элемент второго; второй элемент первого, второй второго и т. д. Хвост более длинного списка добавить в хвост объединенного. Второй список должен остаться «пустым».