

**Внимание!** "не реализовывать" в этих заданиях означает "поставить в метод заглушки типа `return null` или `return 0`", чтобы метод работал.  
Все классы/интерфейсы пишем в отдельных java-файлах.  
В репозиторий кидаем только java-файлы.

#### **11c**

Реализовать класс `MyArrayCollection`, реализовав его от интерфейса `java.util.Collection<Integer>`. Не реализовывать методы `iterator()` и `toArray`. В качестве реализации внутри взять массив.

#### **11b**

Реализовать класс `MyArrayList`, реализовав его от интерфейса `java.util.List<Integer>` и унаследовав от `MyArrayCollection`. Не реализовывать методы `iterator()`, `listIterator()` и `toArray`. В качестве реализации внутри взять массив.

#### **11a**

Реализовать класс `MyCollection<T>`, реализовав его от интерфейса `java.util.Collection<T>`. Не реализовывать методы `iterator()` и `toArray`. Реализацией внутри берите односвязный линейный список.

#### **12c**

Реализовать класс `MyLinkedCollection`, унаследовав его от `java.util.Collection<Integer>`. Не реализовывать методы `iterator()` и `toArray`. В качестве реализации внутри взять односвязный линейный список.

#### **12b**

Реализовать класс `MyLinkedList`, реализовав его от интерфейса `java.util.List<Integer>` и унаследовав от `MyLinkedCollection`. Не реализовывать методы `iterator()`, `listIterator()` и `toArray`. В качестве реализации внутри взять односвязный линейный список.

#### **12a**

Реализовать класс `MyList<T>`, реализовав его от интерфейса `java.util.List<T>` и унаследовав от реализованного в 11a `MyCollection<T>`. Не реализовывать методы `iterator()`, `listIterator()` и `toArray`. Реализацией внутри берите односвязный линейный список.

#### **13c**

Реализовать интерфейс `MyStack<T>` с методами `push`, `pop`, `isEmpty`. Далее реализовать от него класс `MyLinkedStack<T>`, в котором для хранения элементов используется линейный односвязный список, элементы которого являются объектами класса `Elem<T>` (получается, `Elem` тоже надо параметризовать)

**14c**

С помощью реализованного в 13с `MyLinkedStack<T>` развернуть введенный целочисленный массив.

**14b**

С помощью реализованного в 13с `MyLinkedStack<T>` для введенной строки проверить в ней правильность расстановки скобок. При этом обязательно вывести типа ошибки, если она произошла:

"не все открывающие закрыты", "встретилась лишняя закрывающая", "скобки не соответствуют друг другу"

**14a**

С помощью реализованного в 13с `MyLinkedStack<T>` для введенной строки, представляющей собой обратную польскую запись, написать алгоритм, вычисляющий представленное в ней арифметическое выражения. Считать вводимые данные всегда правильными, лексемы разделены запятой.

Если забыли про то, что это, в Вики есть отличная статья, там есть и набросок алгоритма.

*пример входа: 2,2,\*,3,+,4,-*

*выход: -3*