## GENERICS-1 - Circular Buffer

**Conversation**

Implement a generic circular buffer (<https://en.wikipedia.org/wiki/Circular_buffer>) **based on a simple array** with the following methods. The size of the circular buffer is limited, the limit should be provided for the constructor of the class.

**Methods of CircularBuffer**

|  |
| --- |
| void put(T t);  T get();  Object[] toObjectArray();  T[] toArray();  List<T> asList();  void addAll(List<? extends T> toAdd);  void sort(Comparator<? super T> comparator);  boolean isEmpty(); |

Hint: Check AbstractList.toArray() to implement the toArray() method.

**Confirmation**

The name of the package and the class should be:  com.epam.exercises.circularbuffer.CircularBuffer

Implement a simple program (writing a real unit test is not expected) to demonstrate and test each method of the circular buffer.

void put(T t)

This method inserts t into the buffer at the position of the head. After the insertion, the head points to the next index in the buffer.

The put method throws RUNTIME exception if the buffer is full. The buffer is full when the head and the tail points to the same index, but the buffer is not empty.

T get()

This method gets the value at the tail of the buffer. After that, the tail points to the next index in the buffer.

The get method throws RUNTIME exception if the buffer is empty. The buffer is empty if the head and the tail points to the same index, and the buffer is not full.

Object[] toObjectArray()

This method returns the buffer as an Object array. The size of the array should be equal to the current number of actual elements in the buffer. The first element should be the tail.

T[] toArray()

This method returns the buffer as a type T array. The size of the array should be equal to the current number of actual elements in the buffer. The first element should be the tail.

List<T> asList()

This method returns the buffer as a list with type T. The size of the list should be equal to the current number of actual elements in the buffer. The first element should be the tail.

void addAll(List<? extends T> toAdd)

This method adds all elements from a given list (which can be type T and T's subclasses) to the buffer. It throws an exception if there is not enough free space in the buffer to add all elements.

void sort(Comparator<? super T> comparator)

This method sorts the buffer with the given comparator. As the order of the elements change in the buffer, the positions of the head and the tail should be maintained as well.

boolean isEmpty()

This method returns true if the buffer is empty, false otherwise.