Evens First

A ***queue*** is an abstract data type (ADT) that you see a lot in real life – lines of cars at stoplights, people waiting in line at the bank, documents waiting to be printed – they all follow this "first in, first out" (FIFO) sequence. In a queue, addition only happens at the "rear", and removal only occurs at the "front". Solve the problems below to learn more about queues.

1. Create a class **EvensFirstRunner.java** with a `main` method and a class **EvensFirst.java** that will define the method below.

2. In Java, Queue is an interface, a description of how something that ***is-a*** queue should behave - Queue can NOT be instantiated (unlike the java.util.Stack type, which is concrete).

    a. Use the java.util.LinkedList class to instantiate a "queue object"; this class implements Queue (it will behave like a queue when queue methods are utilized).

    b. Queue methods are as follows:

        i. `peek()` Returns (but does not remove) the value at the head of the queue
        ii. `poll()` Removes ("dequeues") and returns the element at the head of the queue
        iii. `offer()` Adds ("enqueues") an element to the queue (at the tail of the queue)

        There are other methods you can use to achieve similar results, but it's best to use a queue with queue-specific methods to avoid confusion.

3. Complete the method `Queue<Integer> putEvensFirst(Queue<Integer> nums)` that will move the even numbers to the front of the queue. You may use additional queues, but no for-each loops (Iterators).

```
(3, 5, 4, 17, 6, 83, 1, 84, 16, 37) >>> head(4, 6, 84, 16, 3, 5, 17, 83, 1, 37)tail
```