DATA STRUCTURES : Queues

Uses/teaches:
* arraylists
* objects
* queues

This example does not examine all of the intricacies of ArrayLists. To do that, please complete the ArrayList assignment. It's assumed that you know this already.

A disco or dance club (ABBA yay!) wants to program a waitlist for when the dance hall is busy.

This is a data structure called a queue. The first person in the queue is the first person to become allowed in when a space becomes available (someone leaves).

We're not going to worry about how many people are already there, nor if people come in groups. It's just individuals.

We'll have two categories: general reservation and VIP reservations. General reservations cost $25 and VIP cost $60.

We're going to have a Patron object: name, phone number, and ticket type.

Queues can be implemented easily using an arraylist.

1. Make a Patron class.
2. We'll make the Waitlist a class, but since we only have waitlist, we'll make everything static.
3. The waitlist will handle enqueing and dequeuing, as well as printing the whole waitlist
4. The Patron ticket information could easily be handled by another class.
5. We'll set up the Ticket class so that it can only create the two required types of tickets.

Add various numbers of people and periodically dequeue them.
e.g. have a total of 12 people added with 4 dequeues amongst the additions

TODO:
We'll want another method in the Waitlist class that will admit the first 5 VIP ticket holders to the venue.

TODO:
Make another method that allows a person to upgrade ticket from "general" to "VIP", for $100!
You would enter the person's name, it would then recover the correct Patron object from the Waitlist and change the ticket type.