Vectors - Implementing a Vector

Dr. Mark R. Floryan

August 20, 2019

1 SUMMARY

For this homework, you will be writing a vector data structure in Java. Your summary:

1. Download the starter code and import the project into Eclipse

2. Implement the Vector.java class

3. Verify your implementation using the provided tester class

4. **FILES TO DOWNLOAD:** Vectors.zip

5. **FILES TO SUBMIT:** VectorsImplementation.zip

1.1 VECTOR.JAVA

Your only task is to implement all of the methods in Vector.java. This class will implement the provided List interface, which is duplicated for your convenience here. Your task is to implement each of these methods.

```
public interface List<T> {
```

```java
     /**
      * Returns the size of this list, i.e., the number
      * of nodes currently between the head and tail
      * @return
      */
     public int size();

     /**
      * Clears out the entire list
      */
     public void clear() ;

     /**
      * Inserts new data at the end of the
      * list (i.e., just before the dummy tail node)
      * @param data
      */
     public void insertAtTail(T data);

     /**
      * Inserts data at the front of the
      * list (i.e., just after the dummy head node
      * @param data
      */
     public void insertAtHead(T data);

     /**
      * Inserts node such that index becomes the
      * position of the newly inserted data
      * @param data
      * @param index
      */
     public void insertAt(int index, T data);

     public T removeAtTail();

     public T removeAtHead();

     /**
      * Returns index of first occurrence of
      * the data in the list, or -1 if not present
      * @param data
      * @return
      */
```

```
47     public int find(T data);

49     /**
        * Returns the data at the given index, null if
51      * anything goes wrong (index out of bounds, empty list, etc.)
        * @param index
53      * @return
        */
55     public T get(int index);
    }
```

You'll probably want to add a couple of supporting methods. For example, a resize() method is useful for when the underlying array of the vector needs to grow.

1.2 TESTING YOUR IMPLEMENTATION

Once you implement this methods, you can test them by running the main method in the provider TestVectore.java class. This simple tester will execute the methods in your list and compare them to those in Java's built in List to make sure the results are as expected.

One strategy might be to implement your methods in the order that they are tested. That way, you can see the tester say "this method is correct" before moving on to the next one.

Notice that we expect your Vector to be a generic class, meaning any type of Object can be stored within your Vector.