

CS 245 Preassignment 2

Magic Array

due before October 31

1 Overview

In this preassignment, you will create an array that holds objects of any class and grows as needed—a magic array. Your class will be similar to `ArrayList`.

Some people claim that we already have these collection classes lying around in our libraries, so why reimplement them? Well, we don't reinvent the wheel because we need new wheels. We reinvent the wheel because we need new inventors—like you. Visiting the underbelly of something you already know from a different angle is not a waste of time.

2 Requirements

1. Write all code in package `pre2`.
2. Write a class named `MagicArray`. It has the following specification, all of whose prescribed methods are `public`:
 - (a) It is a generic class that accepts a single generic parameter.
 - (b) This class uses a traditional Java array as the backing store for the elements. Nowhere in your code can you even mention the classes `ArrayList` or `Vector` or other collection classes.
 - (c) It has a constructor that accepts the initial capacity that the list can hold. `MagicArrays` are born empty.
 - (d) It has a method `get` that accepts an `int` index as a parameter and returns a reference to the object of the generic type at the given index. If the index is invalid, an `IndexOutOfBoundsException` is thrown.
 - (e) It has a method `size` that returns the number of elements in the array.
 - (f) It has a method `add` that accepts a reference to an object of the generic type. It adds the parameter to the backing array, after all previously inserted elements. If there's no room, a new, larger backing array is created, to which all the previously inserted items are first copied.
 - (g) It has a method `remove` that accepts an `int` index of the item to be removed from the array. After removal of index `i`, all elements with index greater than `i` are shifted toward the beginning of the array to maintain contiguousness. If the index is invalid, throw an `IndexOutOfBoundsException`.
 - (h) It has a method `remove` that accepts a reference to an object of the generic type. The first element in the list that matches the parameter is removed. (Use the `equals` method as the criteria for “matches.” Operator `==` tells us whether two references point to the same instance, which is sometimes useful. Here, we just care about equal content.) Any subsequent elements are shifted toward the beginning of the array. (How can you reuse code you've already written?) If no element matches, throw a `NoSuchElementException`.

3 Submission

This is preassignment and is graded with help from the SpecChecker. Your work will also be inspected by human eyes for plagiarism. Please do your own work. Talk about code with your classmates. Ask questions of your instructor or TA. Do not look at others' code. Do not ask questions specific to your homework anywhere online but Piazza. (If you find violators of this rule, please let me know.) Write your own code.

Put the SpecChecker in your Build Path. Run it as a Java Application (not a JUnit Test) and fix problems until all tests pass. Upload the resulting ZIP file to the W drive.

4 Files

- http://www.twodee.org/teaching/cs245/2013C/homework/magicarray/speccheck_pre2.jar