AP Computer Science A@Beijing National Day School

Lab 15: `HorseBarn`

Due date: Thursday, April 18, 2019
Instructor: Mr. Alwin Tareen
Total Points: 15

Task Overview

- Implement a program that simulates a horse barn that contains a series of stalls.

Background

- Consider a software system that models a horse barn. Classes that represent horses implement the following interface:

```
public interface Horse
{
    /** @return The horse's name */
    String getName();

    /** @return The horse's weight */
    int getWeight();
}
```

- A horse barn consists of N numbered spaces. Each space can hold at most one horse. The spaces are indexed starting from 0; the index of the last space is $N - 1$. No two horses in the barn have the same name.
- The declaration of the `HorseBarn` class is shown below. You will write two unrelated methods of the `HorseBarn` class.

```
public class HorseBarn
{
    /** The spaces in the barn. Each array element holds a reference to the horse that is
     *  currently occupying the space. A null value indicates an empty space.
     */
    private Horse[] spaces;

    /** Returns the index of the space that contains the horse with the specified name.
     *  Precondition: No two horses in the barn have the same name.
     *  @param name the name of the horse to find
     *  @return the index of the space containing the horse with the specified name;
     *  -1 if no horse with the specified name is in the barn.
     */
    public int findHorseSpace(String name)
    {
        // to be implemented in part (a)
    }

    /** Consolidates the barn by moving horses so that the horses are in adjacent spaces,
     *  starting at index 0, with no empty spaces between any two horses.
     *  Postcondition: The order of the horses is the same as before the consolidation.
     */
    public void consolidate()
    {
        // to be implemented in part (b)
    }
}
```

The `findHorseSpace()` Method

(a) Write the `HorseBarn` method `findHorseSpace()`. This method returns the index of the space in which the horse with the specified name is located. If there is no horse with the specified name in the barn, then the method returns `-1`.

For example, assume that a `HorseBarn` object called `sweetHome` has horses in the following spaces:

0	1	2	3	4	5	6
"Trigger" 1340	`null`	"Silver" 1210	"Lady" 1575	`null`	"Patches" 1350	"Duke" 1410

The following table shows the results of several calls to the `findHorseSpace()` method.

Method Call	Value Returned	Reason
`sweetHome.findHorseSpace("Trigger")`	0	A horse named Trigger is in space 0.
`sweetHome.findHorseSpace("Silver")`	2	A horse named Silver is in space 2.
`sweetHome.findHorseSpace("Coco")`	-1	A horse named Coco is not in the barn.

The `consolidate()` Method

(b) Write the `HorseBarn` method `consolidate()`. This method consolidates the barn by moving horses, so that the horses are in adjacent spaces, starting at index `0`, with no empty spaces between any two horses. After the barn is consolidated, the horses are in the same order as they were before the consolidation.

0	1	2	3	4	5	6
"Trigger" 1340	`null`	"Silver" 1210	`null`	`null`	"Patches" 1350	"Duke" 1410

The following table shows the arrangement of the horses after the `consolidate()` method is called.

0	1	2	3	4	5	6
"Trigger" 1340	"Silver" 1210	"Patches" 1350	"Duke" 1410	`null`	`null`	`null`

Specification

The Information Box Which Includes Your Name[5 points]

- Type your English and Pinyin name into the `Author` field, where it says: `YOUR NAME HERE`

Manage the Horses in a Horse Barn [10 points]

- Write a `Java` program in the file `HorseBarn.java` that manages the horses in a horse barn.
- You will write your solution in a class called: `public class HorseBarn` right below the place where it says: `YOUR CODE HERE`.
- Make sure that you run your `Java` program, and ensure that it is free of errors.

Testing

- The file `HorseBarnJUnitTest.java` contains the `JUnit` test cases which verify the correct functionality of the program.

Submission

- Submit your `Java` program by uploading it to the `Web-CAT` automated grading platform: `http://ec2-54-65-207-33.ap-northeast-1.compute.amazonaws.com:8080/Web-CAT/WebObjects/Web-CAT.woa`