

Classes Horse FRQ

Consider a software system that models a horse barn. A horse is represented by the following class.

```
class Horse
{
    public String getName()
    { // Implementation not shown. }

    public int getWeight()
    { // Implementation not shown. }

    /* There may be instance variables, constructors, and methods that are not
    shown. */
}
```

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 space 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) */ }

    /* There may be instance variables, constructors, and methods that
    are not shown. */
}
```

(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, the method returns -1.

For example, assume a *HorseBarn* object called *sweetHome* has horses in the following spaces.

0	1	2	3	4	5	6
"Trigger" 1340	<i>null</i>	"Silver" 1210	"Lady" 1575	<i>null</i>	"Patches" 1350	"Duke" 1410

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

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

Information repeated from the beginning of the question.

```
public class Horse
```

```
public String getName()
```

```
public int getWeight()
```

```
public class HorseBarn
```

```
private Horse[] spaces
```

```
public int findHorseSpace(String name)
```

```
public void consolidate()
```

Complete the method *findHorseSpace()* below.

```
/** 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)
```

Testing Code Link:

<https://www.jdoodle.com/a/2WFY>

(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.

For example, assume a barn has horses in the following spaces.

0	1	2	3	4	5	6
"Trigger" 1340	<i>null</i>	"Silver" 1210	<i>null</i>	<i>null</i>	"Patches" 1350	"Duke" 1410

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

0	1	2	3	4	5	6
"Trigger" 1340	"Silver" 1210	"Patches" 1350	"Duke" 1410	<i>null</i>	<i>null</i>	<i>null</i>

Information repeated from the beginning of the question.

```
public class Horse
```

```
public String getName()
```

```
public int getWeight()
```

```
public class HorseBarn
```

```
private Horse[] spaces
```

```
public int findHorseSpace(String name)
```

```
public void consolidate()
```

Complete the method *consolidate()* below.

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

Testing Code Link:

<https://www.jdoodle.com/a/2WfY>