Problem Set 4 Exercise #32: Set

Reference: Lecture 10 OOP Unit 3 notes

Learning objectives: Object-oriented programming; Array as an attribute

Estimated completion time: 45 minutes

Problem statement:

In mathematics, a set is a group of distinct items.

Write a **`Set`** class with the following two attributes:

- **`private int size;  // the number of data in a set`**
- **`private int[] values; // a collection of data`**

The **`Set`** class should support the following operations:

- **`Set(int num)`** to create a set. Parameter **`num`** indicates the number of data to be inserted into the set (at the moment of creation the set is still empty.).
- **`void add(int data)`** to add **`data`** to the first available (empty) slot in the set. For example, if the set is empty, then **`data`** will be stored in slot #0; if the set already has 2 data (which shall occupy slots #0 and #1), then **`data`** will be stored in slot #2.
- **`boolean contains(int data)`** to check if **`data`** exists in the set.
- **`int getSize()`**to return the size (number of data) of the set.
- **`boolean isSubsetOf(Set another)`** to check if **`this`** set is a subset of **`another`**. Neither set shall be modified by the method call.
- **`boolean equals(Set another)`** to check if **`this`** set has the same content as **`another`**. Neither set shall be modified by the method call.
- **`String toString()`**to return the string representation of data in the set, e.g. return **`"[1, 5, 3, 7, 2]"`**.

Write a user program **PS4_Ex32_Set.java** to read two set of integers and compare them. The user program should contain the following static method in addition to the **`main()`** method:

- **`Set readSet(Scanner scanner, String index)`** that reads and returns a set of integers.

You may assume that user will give distinct data to each set.

The **`main()`** method is complete and you are not supposed to modify it.

Sample run #1:

```
Enter the number of elements in Set 1: 3
Enter 3 element(s): 1 2 3
Enter the number of elements in Set 2: 4
Enter 4 element(s): 4 3 2 1
Set 1 has 3 element(s): [1, 2, 3]
Set 2 has 4 element(s): [4, 3, 2, 1]
Two sets have different values
```

Sample run #2:

```
Enter the number of elements in Set 1: 4
Enter 3 element(s): 4 3 2 1
Enter the number of elements in Set 2: 1
Enter 1 element(s): 1
Set 1 has 4 element(s): [4, 3, 2, 1]
Set 2 has 1 element(s): [1]
Two sets have different values
```

Sample run #3:

```
Enter the number of elements in Set 1: 4
Enter the 3 element(s): 1 2 3 4
Enter the number of elements in Set 2: 4
Enter the 4 element(s): 4 3 2 1
Set 1 has 4 element(s): [1, 2, 3, 4]
Set 2 has 4 element(s): [4, 3, 2, 1]
Two sets are identical
```