



LAB

JAN 2023

TEB1113

Algorithm & Data Structure

Lab 3

NO.	NAME	STUDENT ID	PROGRAM (IT / IS / CS / BM)
1.	CHENG PIN-JIE	21000548	CS

Assoc. Prof. Dr Manzoor Ahmed Hashmani

Madam Maryam Omar Abdullah Sawad

Homework 1

```
import java.util.Scanner;
```

```
class Bag {
    // Attributes
    private int[] arrBag;
    private int currentItemLoc;

    // Method
    /*
     * Postcondition: The Bag has been initialized
     * and it is now empty.
     */
    public Bag(int size) {
        currentItemLoc = 0;
        arrBag = new int[size];
    }

    /*
     * Precondition: The bag is not full
     * Postcondition: The item (represent in integer) is being
     * added to the array (bag)
     */
    public void add(int entry) {
        if (currentItemLoc < arrBag.length) {
            arrBag[currentItemLoc] = entry;
            currentItemLoc++;
        } else {
            System.out.println("The bag is full");
        }
    }

    /*
     * Precondition: The element must exist inside the bag.
     */
}
```

```

    * Postcondition: If target was in the Bag, then
    * one copy of target has been removed from the
    * Bag, and the return value is true; otherwise the
    * Bag is unchanged and the return value is false.
    */
public boolean remove(int entry) {
    if (countOccur(entry) > 0) {
        for (int i = 0; i < arrBag.length; i++) {
            if (entry == arrBag[i]) {
                for (int j = i; j < currentItemLoc; j++) {
                    arrBag[j] = arrBag[j + 1];
                }
            }
        }
        return true;
    } else {
        return false;
    }
}

/*
    * Precondition: Do not count the empty space.
    * Postcondition: The return value is the number
    * of integers in the Bag.
    */
public int calcSize() {
    int counter = 0;
    for (int i = 0; i < currentItemLoc; i++) {
        if (arrBag[i] != 0) {
            counter++;
        }
    }
    return counter;
}

```

```

/*
 * Postcondition: The return value is the number
 * of copies of target in the Bag.
 */
public int countOccur(int entry) {
    int counter = 0;
    for (int i = 0; i < arrBag.length; i++) {
        if (entry == arrBag[i]) {
            counter++;
        }
    }
    return counter;
}

public void printAllItems() {
    for (int i = 0; i < currentItemLoc; i++) {
        if (arrBag[i] != 0) {
            System.out.println(arrBag[i]);
        }
    }
}

/* Precondition: Which item position should be known
 * Postcondition: return the position of the item;
 */
public int positionOf(int entry) {
    int position = 0;
    for (int i = 0; i < currentItemLoc; i++) {
        if (arrBag[i] == entry) {
            i = position + 1;
            break;
        }
    }
    return position;
}

```

```
}
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Bag bag = new Bag(10);
```

```
        // Use while loop to keep on asking user until user ends.
```

```
        while (true) {
```

```
            System.out.println("_____BAG_____");
```

```
            System.out.println("[1] Add items");
```

```
            System.out.println("[2] Remove items");
```

```
            System.out.println("[3] Size of bag");
```

```
            System.out.println("[4] Find the occurrence of items");
```

```
            System.out.println("[5] Find the position of item");
```

```
            System.out.println("[6] Quit program");
```

```
        // Use scanner to get user input
```

```
        Scanner obj = new Scanner(System.in);
```

```
        System.out.println("Enter your option: ");
```

```
        String in = obj.nextLine();
```

```
        int inInt = Integer.parseInt(in);
```

```
        // Use switch case to alter the option and do the  
        following:
```

```
        switch (inInt) {
```

```
            case 1: // Add item
```

```
                System.out.println("What item you want to add: ");
```

```
                Scanner obj1 = new Scanner(System.in);
```

```
                System.out.println("Enter your item: ");
```

```
                String in1 = obj1.nextLine();
```

```
                int inInt1 = Integer.parseInt(in1);
```

```
                bag.add(inInt1);
```

```
                System.out.println(" ");
```

```
                continue;
```

```

        case 2: // Remove item
            System.out.println("What item you want to remove:
");
            Scanner obj2 = new Scanner(System.in);
            System.out.println("Enter your item: ");
            String in2 = obj2.nextLine();
            int inInt2 = Integer.parseInt(in2);
            bag.remove(inInt2);
            System.out.println(" ");
            continue;
        case 3: // Calculate size of bag
            bag.calcSize();
            System.out.println("The size of the bag now is: " +
bag.calcSize());
            System.out.println(" ");
            continue;
        case 4: // Find occur
            System.out.println("What item you want to count: ");
            Scanner obj4 = new Scanner(System.in);
            System.out.println("Enter your item: ");
            String in4 = obj4.nextLine();
            int inInt4 = Integer.parseInt(in4);
            System.out.println("The occurance of the item is " +
bag.countOccur(inInt4));
            System.out.println(" ");
            continue;
        case 5: // Find position
            System.out.println("What item you want to find: ");
            Scanner obj5 = new Scanner(System.in);
            System.out.println("Enter your item: ");
            String in5 = obj5.nextLine();
            int inInt5 = Integer.parseInt(in5);
            System.out.println("The position of the item is " +
bag.positionOf(inInt5));
            System.out.println(" ");
            continue;

```

```
        case 6: // Quit program
            System.out.println("Thank you!");
            return;
        }
    }
}
```

Homework 2

Program 1 - Reverse within one array

```
import java.util.Scanner;

class Bag {
    // Attributes
    private int[] arrBag;
    private int currentItemLoc;

    // Method
    /*
     * Postcondition: The Bag has been initialized
     * and it is now empty.
     */
    public Bag(int size) {
        currentItemLoc = 0;
        arrBag = new int[size];
    }

    /*
     * Precondition: The bag is not full
     * Postcondition: The item (represent in integer) is being
     * added to the array (bag)
     */
    public void add(int entry) {
        if (currentItemLoc < arrBag.length) {
            arrBag[currentItemLoc] = entry;
            currentItemLoc++;
        } else {
            System.out.println("The bag is full");
        }
    }
}
```



```

/*
 * Precondition: The element must exist inside the bag.
 * Postcondition: If target was in the Bag, then
 * one copy of target has been removed from the
 * Bag, and the return value is true; otherwise the
 * Bag is unchanged and the return value is false.
 */
public boolean remove(int entry) {
    if (countOccur(entry) > 0) {
        for (int i = 0; i < arrBag.length; i++) {
            if (entry == arrBag[i]) {
                for (int j = i; j < currentItemLoc; j++) {
                    arrBag[j] = arrBag[j + 1];
                }
            }
        }
        return true;
    } else {
        return false;
    }
}

```

```

/*
 * Precondition: Do not count the empty space.
 * Postcondition: The return value is the number
 * of integers in the Bag.
 */
public int calcSize() {
    int counter = 0;
    for (int i = 0; i < currentItemLoc; i++) {
        if (arrBag[i] != 0) {
            counter++;
        }
    }
    return counter;
}

```

```
}
```

```
/*
```

```
 * Postcondition: The return value is the number  
 * of copies of target in the Bag.
```

```
*/
```

```
public int countOccur(int entry) {  
    int counter = 0;  
    for (int i = 0; i < arrBag.length; i++) {  
        if (entry == arrBag[i]) {  
            counter++;  
        }  
    }  
    return counter;  
}
```

```
public void printAllItems() {  
    for (int i = 0; i < currentItemLoc; i++) {  
        if (arrBag[i] != 0) {  
            System.out.println(arrBag[i]);  
        }  
    }  
}
```

```
/* Precondition: Item position should be known
```

```
 * Postcondition: return the position of the item;
```

```
*/
```

```
public int positionOf(int entry) {  
    int position = 0;  
    for (int i = 0; i < currentItemLoc; i++) {  
        if (arrBag[i] == entry) {  
            position = i + 1;  
            break;  
        }  
    }  
}
```

```

        return position;
    }

    /* Use a 'temp' variable to store the original array.
     * and then swap the element with the last element of the
     * original element.
     */
    public void reverse(){
        for (int i = 0; i < currentItemLoc/2; i++){
            int temp;
            temp = arrBag[i];
            arrBag[i] = arrBag[currentItemLoc-i-1];
            arrBag[currentItemLoc-i-1] = temp;
        }
    }
}

public class Main {

    public static void main(String[] args) {
        Bag bag = new Bag(10);

        // Use while loop to keep on asking user until user ends.
        while (true) {
            System.out.println("_____BAG_____");
            System.out.println("[1] Add items");
            System.out.println("[2] Remove items");
            System.out.println("[3] Size of bag");
            System.out.println("[4] Find the occurrence of items");
            System.out.println("[5] Find the position of item");
            System.out.println("[6] Reverse the items in the bag");
            System.out.println("[7] Quit program");

            // Use scanner to get user input
            Scanner obj = new Scanner(System.in);

```

```

        System.out.println("Enter your option: ");
        String in = obj.nextLine();
        int inInt = Integer.parseInt(in);

        // Use switch case to alter the option and do the
        following:
        switch (inInt) {
            case 1: // Add item
                System.out.println("What item you want to add: ");
                Scanner obj1 = new Scanner(System.in);
                System.out.println("Enter your item: ");
                String in1 = obj1.nextLine();
                int inInt1 = Integer.parseInt(in1);
                bag.add(inInt1);
                System.out.println(" ");
                continue;
            case 2: // Remove item
                System.out.println("What item you want to remove:
");
                Scanner obj2 = new Scanner(System.in);
                System.out.println("Enter your item: ");
                String in2 = obj2.nextLine();
                int inInt2 = Integer.parseInt(in2);
                bag.remove(inInt2);
                System.out.println(" ");
                continue;
            case 3: // Calculate size of bag
                bag.calcSize();
                System.out.println("The size of the bag now is: " +
bag.calcSize());
                System.out.println(" ");
                continue;
            case 4: // Find occur
                System.out.println("What item you want to count: ");
                Scanner obj4 = new Scanner(System.in);
                System.out.println("Enter your item: ");

```

```

        String in4 = obj4.nextLine();
        int inInt4 = Integer.parseInt(in4);
        System.out.println("The occurrence of the item is " +
bag.countOccur(inInt4));
        System.out.println(" ");
        continue;
    case 5: // Find position
        System.out.println("What item you want to find: ");
        Scanner obj5 = new Scanner(System.in);
        System.out.println("Enter your item: ");
        String in5 = obj5.nextLine();
        int inInt5 = Integer.parseInt(in5);
        System.out.println("The position of the item is " +
bag.positionOf(inInt5));
        System.out.println(" ");
        continue;
    case 6:
        bag.reverse();
        System.out.println("The items in the bag have been
reversed.");
        System.out.println(" ");
        continue;
    case 7: // Quit program
        System.out.println("Thank you!");
        return;
    }
}
}
}
}

```

Program 2 - Reverse using another bag.

```
import java.util.Scanner;

class Bag {
    // Attributes
    private int[] arrBag;
    private int[] arrBag2;
    private int currentItemLoc;

    // Method
    /*
     * Postcondition: The Bag has been initialized
     * and it is now empty.
     */
    public Bag(int size) {
        currentItemLoc = 0;
        arrBag = new int[size];
        arrBag2 = new int[size];
    }

    /*
     * Precondition: The bag is not full
     * Postcondition: The item (represent in integer) is being
     * added to the array (bag)
     */
    public void add(int entry) {
        if (currentItemLoc < arrBag.length) {
            arrBag[currentItemLoc] = entry;
            currentItemLoc++;
        } else {
            System.out.println("The bag is full");
        }
    }
}
```

```
}
```

```
/*
```

```
* Precondition: The element must exist inside the bag.
```

```
* Postcondition: If target was in the Bag, then
```

```
* one copy of target has been removed from the
```

```
* Bag, and the return value is true; otherwise the
```

```
* Bag is unchanged and the return value is false.
```

```
*/
```

```
public boolean remove(int entry) {
```

```
    if (countOccur(entry) > 0) {
```

```
        for (int i = 0; i < arrBag.length; i++) {
```

```
            if (entry == arrBag[i]) {
```

```
                for (int j = i; j < currentItemLoc; j++) {
```

```
                    arrBag[j] = arrBag[j + 1];
```

```
                }
```

```
            }
```

```
        }
```

```
        return true;
```

```
    } else {
```

```
        return false;
```

```
    }
```

```
}
```

```
/*
```

```
* Precondition: Do not count the empty space.
```

```
* Postcondition: The return value is the number
```

```
* of integers in the Bag.
```

```
*/
```

```
public int calcSize() {
```

```
    int counter = 0;
```

```
    for (int i = 0; i < currentItemLoc; i++) {
```

```
        if (arrBag[i] != 0) {
```

```
            counter++;
```

```
        }
```

```

    }
    return counter;
}

/*
 * Postcondition: The return value is the number
 * of copies of target in the Bag.
 */
public int countOccur(int entry) {
    int counter = 0;
    for (int i = 0; i < arrBag.length; i++) {
        if (entry == arrBag[i]) {
            counter++;
        }
    }
    return counter;
}

public void printAllItems() {
    for (int i = 0; i < currentItemLoc; i++) {
        if (arrBag[i] != 0) {
            System.out.println(arrBag[i]);
        }
    }
}

/* Precondition: Item position should be known
 * Postcondition: return the position of the item;
 */
public int positionOf(int entry) {
    int position = 0;
    for (int i = 0; i < currentItemLoc; i++) {
        if (arrBag[i] == entry) {
            position = i + 1;
            break;
        }
    }
}

```



```

        }
    }
    return position;
}

/* Create another abg named arrBag2 to copy the original
 * array reversely.
 */
public void reverse(){
    for (int i = 0; i < currentItemLoc; i++){
        arrBag2[i] = arrBag[currentItemLoc-i];
    }
}
}

public class Main {

    public static void main(String[] args) {
        Bag bag = new Bag(10);

        // Use while loop to keep on asking user until user ends.
        while (true) {
            System.out.println("_____BAG_____");
            System.out.println("[1] Add items");
            System.out.println("[2] Remove items");
            System.out.println("[3] Size of bag");
            System.out.println("[4] Find the occurance of items");
            System.out.println("[5] Find the position of item");
            System.out.println("[6] Reverse the items in the bag");
            System.out.println("[7] Quit program");

            // Use scanner to get user input
            Scanner obj = new Scanner(System.in);
            System.out.println("Enter your option: ");
            String in = obj.nextLine();

```

```

        int inInt = Integer.parseInt(in);

        // Use switch case to alter the option and do the
        following:
        switch (inInt) {
            case 1: // Add item
                System.out.println("What item you want to add: ");
                Scanner obj1 = new Scanner(System.in);
                System.out.println("Enter your item: ");
                String in1 = obj1.nextLine();
                int inInt1 = Integer.parseInt(in1);
                bag.add(inInt1);
                System.out.println(" ");
                continue;
            case 2: // Remove item
                System.out.println("What item you want to remove:
");
                Scanner obj2 = new Scanner(System.in);
                System.out.println("Enter your item: ");
                String in2 = obj2.nextLine();
                int inInt2 = Integer.parseInt(in2);
                bag.remove(inInt2);
                System.out.println(" ");
                continue;
            case 3: // Calculate size of bag
                bag.calcSize();
                System.out.println("The size of the bag now is: " +
bag.calcSize());
                System.out.println(" ");
                continue;
            case 4: // Find occur
                System.out.println("What item you want to count: ");
                Scanner obj4 = new Scanner(System.in);
                System.out.println("Enter your item: ");
                String in4 = obj4.nextLine();
                int inInt4 = Integer.parseInt(in4);

```

```

        System.out.println("The occurrence of the item is " +
bag.countOccur(inInt4));
        System.out.println(" ");
        continue;
    case 5: // Find position
        System.out.println("What item you want to find: ");
        Scanner obj5 = new Scanner(System.in);
        System.out.println("Enter your item: ");
        String in5 = obj5.nextLine();
        int inInt5 = Integer.parseInt(in5);
        System.out.println("The position of the item is " +
bag.positionOf(inInt5));
        System.out.println(" ");
        continue;
    case 6:
        bag.reverse();
        System.out.println("The items in the bag have been
reversed.");
        System.out.println(" ");
        continue;
    case 7: // Quit program
        System.out.println("Thank you!");
        return;
    }
}
}
}
}

```