```
package Practice;


import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

public class FixBugs {

public static void main(String[] args) {

System.out.println("\n*************\n");

System.out.println("\tWelcome to TheDesk \n");

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

optionsSelection();

}

private static void optionsSelection() {

String[] arrMenuOptions = {"1. I wish to review my expenditure",

                           "2. I wish to add my expenditure",

                           "3. I wish to delete my expenditure",

                           "4. I wish to sort the expenditures",

                           "5. I wish to search for a particular expenditure",

                           "6. Close the application"

                            };

int[] arr1 = {1, 2, 3, 4, 5, 6};

ArrayList<Integer>arrlist = new ArrayList<Integer>();

ArrayList<Integer>expenses = new ArrayList<Integer>();

expenses.add(1200);

expenses.add(500);
expenses.add(3000);
expenses.add(7000);
expenses.add(220);
expenses.addAll(arrlist);
Scanner sc = new Scanner(System.in);
int options = 0;
while (options != 6) {
displayMenuOptions(arrMenuOptions);
System.out.println("\nEnter your choice:\t");
options = sc.nextInt();
switch (options) {
case 1:
            System.out.println("Your saved expenses are listed below: \n");
            if (expenses.isEmpty())
            {
            System.out.println("Expenses list is empty\n");
             }
            else
            {
            System.out.println(expenses + "\n");
             }
            break;
case 2:
            System.out.println("Enter the value to add your Expense: \n");
            int value = sc.nextInt();
```

```
                    expenses.add(value);
                    System.out.println("Your value is updated\n");
                    expenses.addAll(arrlist);
                    System.out.println(expenses + "\n");
                    break;
case 3:
                    System.out.println("You are about the delete all your expenses! \nConfirm again by selecting the same option...\n");
                    int con_choice = sc.nextInt();
                    if (con_choice == options)
                    {
                    expenses.clear();
                    System.out.println(expenses + "\n");
                    System.out.println("All your expenses are erased!\n");
                     }
                    else
                    {
                    System.out.println("Oops... try again!");
                      }
                    break;
case 4:
                    sortExpenses(expenses);
                    break;
case 5:
                    searchExpenses(expenses);
                    break;
```

```
case 6:
                    closeApp();
                    break;
default:
                    System.out.println("You have made an invalid choice!\nTry again!\n");
                    break;
                            }
                          }
                       }
private static void displayMenuOptions(String[] arrMenuOptions) {
int slen = arrMenuOptions.length;
for (int i = 0; i<slen; i++)
{
                    // display the all the Strings mentioned in the String array
System.out.println(arrMenuOptions[i]);
 }
   }
private static void closeApp() {
System.out.println("Closing your application... \nThank you!");
   }
private static void searchExpenses(ArrayList<Integer>arrayList) {
int leng = arrayList.size();
System.out.println("Enter the expense you need to search:\t");

//Complete the method
```

```
if (leng> 0) {
Scanner sc = new Scanner(System.in);
int searchExpenseNo = sc.nextInt();
int i = 0;
for (i = 0; i<leng; i++) {
if (arrayList.get(i) == searchExpenseNo) {
break;
                }
            }
if (i != leng) {
System.out.println("Expenditure " + searchExpenseNo + " was found on entry no " + (i + 1) + "\n");
}else {
System.out.println("Expenditure " + searchExpenseNo + " was not found\n");
}
} else {
System.out.println("There are no expenditures available\n");
        }
    }
private static void sortExpenses(ArrayList<Integer>arrayList) {
//Complete the method. The expenses should be sorted in ascending order.
if (arrayList.isEmpty()) {
System.out.println("Sort not performed,\nExpenditure list is empty\n");
        } else {
Collections.sort(arrayList);
System.out.println("Expenditure list sorted");
System.out.println(arrayList + "\n");
        }
    }
}
```