Fix Bugs of the Application

DESCRIPTION

Project objective:

As a developer, fix the bugs in the application using the appropriate algorithmic techniques.

Background of the problem statement:

You have been assigned a few tasks during the sprint planning. Solving the bugs raised by the testing team is one among them. You are given the boilerplate code and are asked to complete it by fixing the bugs.

Bugs to be fixed:

- Add the missing source code to the application based on searching technique. Find the appropriate comments to code for the searching technique.
- Write source code for sorting the predefined array and ensure the functionality of the application. Find the appropriate comments to code for sorting the predefined array.
- You can download the boilerplate code by executing the command below in your **git bash**.

git clone https://github.com/Simplilearn-Edu/Full-Stack---The-Desk-Application-.git

You must use the following:

- **Eclipse/IntelliJ:** An IDE to code for the application
- **Java:** A programming language to develop the prototype
- **Git:** To connect and push files from local system to GitHub
- **GitHub:** To store the application code and track its versions
- **Search and Sort techniques:** Select the relevant data structure algorithms to fix the bugs

Following requirements should be met:

- The source code should be pushed to your GitHub repositories. You need to document the steps and write the algorithms in the Google Docs.
- The link of your GitHub repository is must. In order to track your task, you need to share the link of the repository. You can add a section in the Google Docs.
- Document the step-by-step process involved in completing this task.

Code:-

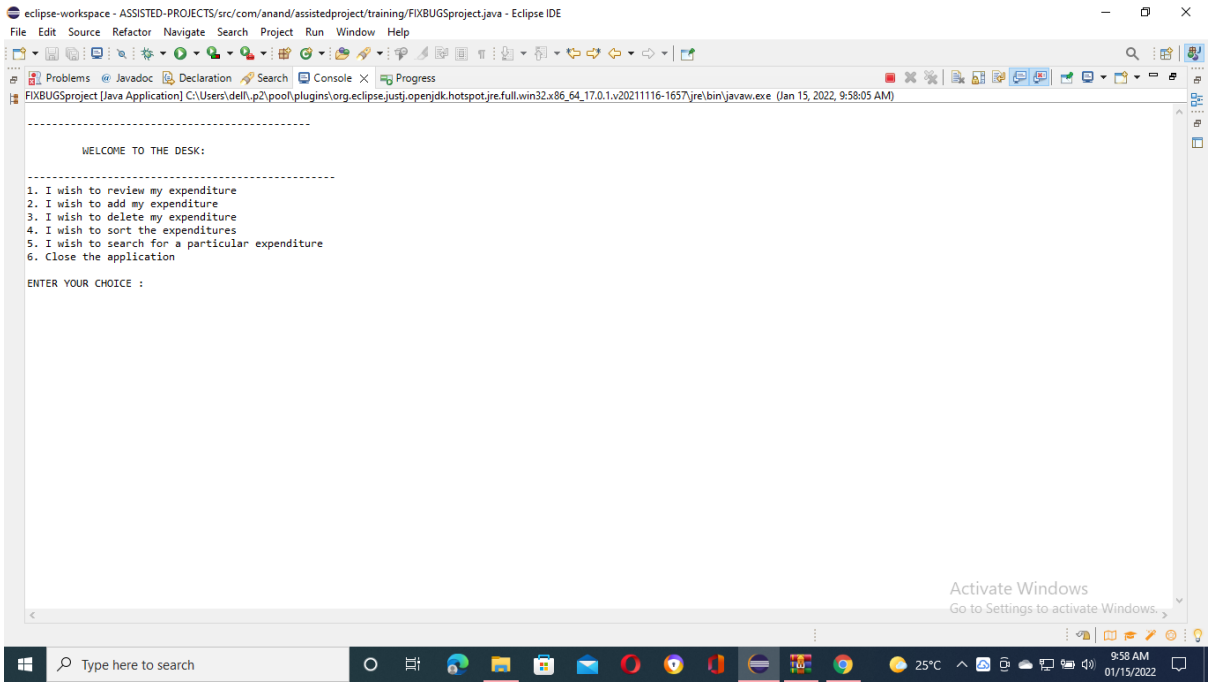
```
package com.anand.assistedproject.training;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;
```

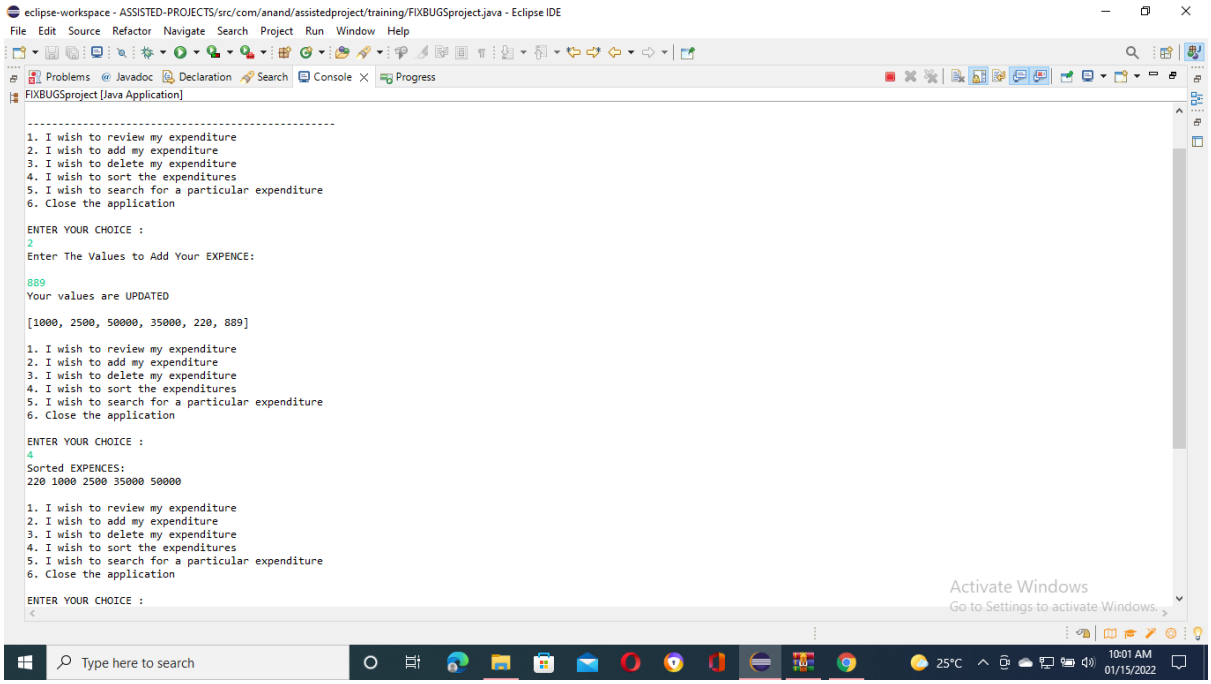
```
public class FIXBUGSproject {
        public static void main(String[] args) {
      //System.out.println("Hello World!");/
      System.out.println("\n-----------------------------------------------\n");
      System.out.println("\t WELCOME TO THE DESK: \n");
      System.out.println("-----------------------------------------------");
      optionsSelection();

   }
   private static void optionsSelection() {
      String[] arr = {"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};
      int  slen = arr1.length;
      for(int i=0; i<slen; i++){
         System.out.println(arr[i]);
         // display the all the Strings mentioned in the String array
      }
      ArrayList<Integer> arrlist = new ArrayList<Integer>();
      ArrayList<Integer> expenses = new ArrayList<Integer>();
      expenses.add(1000);
      expenses.add(2500);
      expenses.add(50000);
      expenses.add(35000);
      expenses.add(220);
      expenses.addAll(arrlist);
      System.out.println("\nENTER YOUR CHOICE :\t");
      @SuppressWarnings("resource")
               Scanner sc = new Scanner(System.in);
      int  options =  sc.nextInt();
      for(int j=1; j<=slen; j++){
         if(options == j){
            switch (options){
               case 1:
                  System.out.println("Your Saved EXPENCES Are Listed Below: \n");
                  System.out.println(expenses+ "\n");
                  optionsSelection();
                  break;

               case 2:
                  System.out.println("Enter The Values to Add Your EXPENCE: \n");
                  int value = sc.nextInt();
                  expenses.add(value);
```

```
                    System.out.println("Your values are UPDATED\n");
                    expenses.addAll(arrlist);
                    System.out.println(expenses+ "\n");
                    optionsSelection();
                    break;

                case 3:
                    System.out.println("You are about the delete all your EXPENCES!! \n Confirm
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 EXPENCES are Erased!\n");
                    } else {
                       System.out.println("Somthing went to Wrong..... try again!");
                    }
                    optionsSelection();
                    break;

                case 4:
                    sortExpenses(expenses);
                    optionsSelection();
                    break;

                case 5:
                    searchExpenses(expenses);
                    optionsSelection();
                    break;

                case 6:
                    closeApp();
                    break;

                default:
                    System.out.println("You have made an invalid choice!");
                    break;
            }
        }
    }

}
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 EXPENCES You Need to Search:\t");
```

```
        //
        @SuppressWarnings("resource")
                Scanner sc = new Scanner(System.in);
        int input = sc.nextInt();
        //Linear Search
        for(int i=0;i<leng;i++) {
           if(arrayList.get(i)==input) {
                System.out.println("Found the EXPENCE " + input + " at " + i + " Position");
           }
        }
    }
    private static void sortExpenses(ArrayList<Integer> arrayList) {
        @SuppressWarnings("unused")
                int arrlength =  arrayList.size();
       //Complete the method. The expenses should be sorted in ascending order.

        Collections.sort(arrayList);
        System.out.println("Sorted EXPENCES: ");
        for(Integer i: arrayList) {
           System.out.print(i + " ");
        }

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

    }
}
```

Output:-




eclipse-workspace - ASSISTED-PROJECTS/src/com/anand/assistedproject/training/FIXBUGSproject.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Search Console Progress
FIXBUGSproject [Java Application]
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
ENTER YOUR CHOICE :
2
Enter The Values to Add Your EXPENCE:
889
Your values are UPDATED
[1000, 2500, 50000, 35000, 220, 889]
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
ENTER YOUR CHOICE :
4
Sorted EXPENCES:
220 1000 2500 35000 50000
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
ENTER YOUR CHOICE :
Activate Windows
Go to Settings to activate Windows.
Type here to search
25°C
10:01 AM
01/15/2022