Fixing Bugs of the Application :

package com.fifthProject;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;

public class BugsFinding2 {

static Scanner sc = new Scanner(System.in);

public static void main(String[] args) {

/\*System.out.println("Hello World!");\*/

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

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

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

optionsSelection();

}

private static void optionsSelection() {

//displaying options

System.out.println("1. I wish to review my expenditure\n"+"2. I wish to add my expenditure\n"+"3. I wish to delete my expenditure\n"+

"4. I wish to sort the expenditures\n"+"5. I wish to search for a particular expenditure\n"+"6. Close the application");

//adding array elements

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

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

expenses.add(1000);

expenses.add(2300);

expenses.add(45000);

expenses.add(32000);

expenses.add(110);

//expenses.addAll(arrlist);

System.out.println("\nEnter your choice:\t");

int options = sc.nextInt();

switch (options){

case 1:

System.out.println("Your saved expenses are listed below: \n");

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

optionsSelection();

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");

optionsSelection();

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!");

}

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) {

System.out.println("Enter the amount you need to search in your expences:\t");

//Complete the method

int n1 = sc.nextInt(), count=0, temp = 0;

for (int i:arrayList) {

if(i == n1) {

count++;

temp=i;

}

}

if(count==1) {

System.out.println(" Data found at index:"+arrayList.indexOf(temp));

}else {

System.out.println(" Data Not Available in our DataBase");

}

}

private static void sortExpenses(ArrayList<Integer> arrayList) {

int l1 = arrayList.size();

//Complete the method. The expenses should be sorted in ascending order.

int[] arr =new int[l1];

int k=0;

for(int a:arrayList) {

arr[k] = a;

k++;

}

BugsFinding2 ob = new BugsFinding2();

ob.mergeSort(arr, 0, l1-1);

System.out.println("Sorted expenses:\n"+Arrays.toString(arr));

}

void mergeSort(int arr[], int l, int r)

{

if (l < r) {

// Find the middle point

int m =l+ (r-l)/2;

// Sort first and second halves

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

// Merge the sorted halves

merge(arr, l, m, r);

}

}

void merge(int arr[], int l, int m, int r)

{

// Find sizes of two subArrays to be merged

int n1 = m - l + 1;

int n2 = r - m;

int L[] = new int[n1];

int R[] = new int[n2];

for (int i = 0; i < n1; ++i)

L[i] = arr[l + i];

for (int j = 0; j < n2; ++j)

R[j] = arr[m + 1 + j];

int i = 0, j = 0;

int k = l;

while (i < n1 && j < n2) {

if (L[i] <= R[j]) {

arr[k] = L[i];

i++;

}

else {

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1) {

arr[k] = L[i];

i++;

k++;

}

while (j < n2) {

arr[k] = R[j];

j++;

k++;

}

}

}