**// linear search and quick sort algorithm is used to finish the searching and sorting methods//**

**import** java.util.ArrayList;

**import** java.util.Scanner;

**import** java.io.\*;

**import** java.util.List;

**public** **class** Main {

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

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

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

System.***out***.println("\tWelcome to TheDesk \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(2300);

expenses.add(45000);

expenses.add(32000);

expenses.add(110);

expenses.addAll(arrlist);

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

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 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) {//linear search

**int** leng = arrayList.size();

**boolean** flag=**true**;

**int**[] arr1=**new** **int**[leng];

**for**(**int** y=0; y<leng; y++) {

arr1[y]=arrayList.get(y);

}

System.***out***.println("Enter the expense you need to search:\t");

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

**int** exp=sc.nextInt();

**for**(**int** q=0; q<leng;q++)

{

**if**(exp==arr1[q])

{

System.***out***.println("expense exists at index "+q);

flag=**false**;

**break**;

}

}

**if**(flag)System.***out***.println("expense doesnt exist");

}

**static** **int** partition(**int** arr[], **int** low, **int** high)

{

**int** pivot = arr[high];

**int** i = (low-1); // index of smaller element

**for** (**int** j=low; j<high; j++)

{

**if** (arr[j] <= pivot)

{

i++;

// swap arr[i] and arr[j]

**int** temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

// swap arr[i+1] and arr[high] (or pivot)

**int** temp = arr[i+1];

arr[i+1] = arr[high];

arr[high] = temp;

**return** i+1;

}

**static** **void** sort(**int** arr[], **int** low, **int** high)

{

**if** (low < high)

{

**int** pi = *partition*(arr, low, high);

*sort*(arr, low, pi-1);

*sort*(arr, pi+1, high);

}

}

**static** **void** printArray(**int** arr[])

{

**int** n = arr.length;

**for** (**int** i=0; i<n; ++i)

System.***out***.print(arr[i]+" ");

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

}

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

//quick sort

**int** arrlength = arrayList.size();

**int**[] arr2=**new** **int**[arrlength];

**for**(**int** x=0; x<arrlength; x++) {

arr2[x]=arrayList.get(x);

}

*sort*(arr2, 0, arrlength-1);

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

*printArray*(arr2);

}

}