Main.java

*/\*  
Rohan Parikh  
Coder Tester lab  
29 September 2020 -  
\*/  
  
  
//Status update for 7 October 2020  
/\*  
Coder sorting is complete, just got to do testing and then let user pick choice  
 \*/  
  
import* java.io.BufferedReader;  
*import* java.io.File;  
*import* java.io.FileReader;  
*import* java.io.IOException;  
*import* java.util.ArrayList;  
*import* java.util.*Comparator*;  
*import* java.util.*List*;  
*import* java.util.Scanner;  
*import* java.util.concurrent.TimeUnit;  
  
*public class* Main {  
 *// Arrays to see if the randomindex has already been used  
 static boolean*[] *usedCoder* = *new boolean*[33];  
 *static boolean*[] *usedTester* = *new boolean*[33];  
  
 *public static void* main(String[] args) {  
 *//printing out to make sure the values are starting out as false  
 // initialize variables and creating array list* File file;  
 *//String TempeoryStorage is for coders first* String tempeoryStorage = *null*;  
 *//String testersTemperory is for testers first* String testersTempeory = *null*;  
  
 *List* <String> studentPairsCodersFirst = *new* ArrayList<>();  
 *List* <String> studentPairsTestersFirst = *new* ArrayList<>();  
 Scanner in = *new* Scanner(System.*in*);  
 *List*<String> allStudents = *new* ArrayList<>();  
  
  
  
  
 BufferedReader br;  
 *int* numOfStudents;  
  
  
 *//Reading file into an array using bufferreader  
  
 try* {  
 file = *new* File("C:\\Users\\mpari\\Documents\\coding projects\\Java\\Coder Tester Software Design Program\\src\\SD\_ClassList.txt");  
 br = *new* BufferedReader(*new* FileReader(String.*valueOf*(file)));  
 *while* (br.ready()) {  
 allStudents.add(br.readLine());  
 }  
 } *catch* (IOException e) {  
 System.*out*.println(e.getMessage());  
 }  
  
  
 *//Asking user for how many students do they want* System.*out*.println("How many students should be in pairs?");  
 numOfStudents = in.nextInt();  
 *if* (numOfStudents > allStudents.size() || numOfStudents <= 0) {  
 System.*out*.println("Sorry, the amount of students you inputted is larger than the students" +  
 " in the file. Input a different number.");  
 numOfStudents = in.nextInt();  
 }  
  
 System.*out*.println("Do you want your list to be sorted coders first or testers first? Note: Choose" +  
 " Testers first if you want it sorted by testers. And vice versa.");  
  
 String codersOrTesters = in.next();  
  
 *//This if statement is coders first  
 if* (codersOrTesters.toLowerCase().equals("coders")) {  
  
 *int* i = 0;  
 *while* (i != numOfStudents) {  
 i++;  
  
 *// method for coders first* tempeoryStorage = ((*studentsAllCoders*(allStudents, numOfStudents)));  
 studentPairsCodersFirst.add(tempeoryStorage);  
 }  
  
 studentPairsCodersFirst.sort(*Comparator*.*comparing*(String::toString));  
 *//Formatting output* String firstLine = String.*format*("%20S %20S ", " Coders", "Testers");  
 System.*out*.println(firstLine);  
 String secondLine = String.*format*("%20S %20S ", " Last Name", " Last Name");  
 System.*out*.println(secondLine);  
 System.*out*.println("-----------------------------------------------");  
 *//for loop to run for amount of coders and then also to split the string and output  
 for* (*int* P = 0; P < studentPairsCodersFirst.size(); P++) {  
 String value = studentPairsCodersFirst.get(P);  
 String[] split = value.split(",");  
 String names = String.*format*("%20S %20S ", split[0], split[1]);  
 System.*out*.println(names);  
 }  
 *try* {  
 TimeUnit.*SECONDS*.sleep(10);  
 } *catch* (InterruptedException e) {  
 e.printStackTrace();  
 }  
 System.*out*.println(" ");  
 System.*out*.println(" ");  
 System.*out*.println(" ");  
 System.*out*.println("Printing Testers first now. ");  
 String firstLine2 = String.*format*("%20S %20S ", " Testers", "Coders");  
 System.*out*.println(firstLine2);  
 String secondLine2 = String.*format*("%20S %20S ", " Last Name", " Last Name");  
 System.*out*.println(secondLine2);  
 System.*out*.println("-----------------------------------------------");  
 *//for loop to run for amount of coders and then also to split the string and output  
 for* (*int* P = 0; P < studentPairsCodersFirst.size(); P++) {  
 String value = studentPairsCodersFirst.get(P);  
 String[] split = value.split(",");  
 String names = String.*format*("%20S %20S ", split[1], split[0]);  
 System.*out*.println(names);  
 }  
 }  
 *else* {  
 *int* i = 0;  
 *while* (i != numOfStudents) {  
 i++;  
  
 *// method for testers first* testersTempeory = ((*studentsAllTesters*(allStudents, numOfStudents)));  
 studentPairsTestersFirst.add(testersTempeory);  
 }  
  
 studentPairsTestersFirst.sort(*Comparator*.*comparing*(String::toString));  
 *//Formatting output but this time testers first* System.*out*.println("Printing testers first then coders. Will let user know about change.");  
 String firstLine = String.*format*("%20S %20S ", " Testers", "Coders");  
 System.*out*.println(firstLine);  
 String secondLine = String.*format*("%20S %20S ", " Last Name", " Last Name");  
 System.*out*.println(secondLine);  
 System.*out*.println("-----------------------------------------------");  
 *//for loop to run for amount of coders and then also to split the string and output  
 for* (*int* P = 0; P < studentPairsTestersFirst.size(); P++) {  
 String value = studentPairsTestersFirst.get(P);  
 String[] split = value.split(",");  
 String names = String.*format*("%20S %20S ", split[0], split[1]);  
 System.*out*.println(names);  
 }  
 *try* {  
 TimeUnit.*SECONDS*.sleep(10);  
 } *catch* (InterruptedException e) {  
 e.printStackTrace();  
 }  
 System.*out*.println(" ");  
 System.*out*.println(" ");  
 System.*out*.println(" ");  
 String firstLine2 = String.*format*("%20S %20S ", " Coders", "Testers");  
 System.*out*.println(firstLine2);  
 String secondLine2 = String.*format*("%20S %20S ", " Last Name", " Last Name");  
 System.*out*.println(secondLine2);  
 System.*out*.println("-----------------------------------------------");  
 *for* (*int* P = 0; P < studentPairsTestersFirst.size(); P++) {  
 String value = studentPairsTestersFirst.get(P);  
 String[] split = value.split(",");  
 String names = String.*format*("%20S %20S ", split[1], split[0]);  
 System.*out*.println(names);  
 }  
  
 }  
 }  
  
 *private static* String studentsAllTesters(*List*<String> allStudents, *int* numOfStudents) {  
 *while* (*true*) {  
 *//common variable for randomindex  
 int* studentsAllLength = allStudents.size();  
 *//random index and inputting arraylist value into a string  
 int* randomIndex = (*int*) (Math.*random*() \* studentsAllLength);  
 *// String randomElement = allStudents.get(randomIndex); Trying to see if using array directly instead of array will work better  
 int* randomIndex2 = (*int*) (Math.*random*() \* studentsAllLength);  
 *//String randomElement2 = allStudents.get(randomIndex2);  
 if* ((!*usedTester*[randomIndex2] && !*usedCoder*[randomIndex]) && randomIndex != randomIndex2) {  
 *// System.out.println(randomIndex);  
 // System.out.println(randomIndex2);  
 // System.out.println(Arrays.toString(usedCoder));  
 // System.out.println(Arrays.toString(usedTester));* String last = allStudents.get(randomIndex2) + "," + allStudents.get(randomIndex);  
 *usedCoder*[randomIndex2] = *true*;  
 *usedTester*[randomIndex] = *true*;  
 *return* last;  
 }  
 }  
 }  
  
 *public static* String studentsAllCoders(*List*<String> allStudents, *int* numOfStudents) {  
 *while* (*true*) {  
 *//common variable for randomindex  
 int* studentsAllLength = allStudents.size();  
 *//random index and inputting arraylist value into a string  
 int* randomIndex = (*int*) (Math.*random*() \* studentsAllLength);  
 *// String randomElement = allStudents.get(randomIndex); Trying to see if using array directly instead of array will work better  
 int* randomIndex2 = (*int*) (Math.*random*() \* studentsAllLength);  
 *//String randomElement2 = allStudents.get(randomIndex2);  
 if* ((!*usedTester*[randomIndex] && !*usedCoder*[randomIndex2]) && randomIndex != randomIndex2)  
 {  
 *// System.out.println(randomIndex);  
 // System.out.println(randomIndex2);  
 // System.out.println(Arrays.toString(usedCoder));  
 // System.out.println(Arrays.toString(usedTester));* String last = allStudents.get(randomIndex) + "," + allStudents.get(randomIndex2);  
 *usedCoder*[randomIndex2] = *true*;  
 *usedTester*[randomIndex] = *true*;  
 *return* last;  
 }  
 }  
 }  
}

Output

C:\Users\mpari\.jdks\liberica-14.0.2\bin\java.exe -javaagent:C:\Users\mpari\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\202.7660.26\lib\idea\_rt.jar=59854:C:\Users\mpari\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\202.7660.26\bin -Dfile.encoding=UTF-8 -classpath "C:\Users\mpari\Documents\coding projects\Java\Coder Tester Software Design Program\out\production\Coder Tester Software Design Program" Main

How many students should be in pairs?

30

Do you want your list to be sorted coders first or testers first? Note: Choose Testers first if you want it sorted by testers. And vice versa.

testers

Printing testers first then coders. Will let user know about change.

TESTERS CODERS

LAST NAME LAST NAME

-----------------------------------------------

BALUSAMY CHITNEEDI

BLANKE CRIMI

BLANKE ELIAS

BLANKE PARRISH-LEWIS

DALAL BORKAR

DALAL MUNOT

DALAL MUZYKA

FAVA GLADSTONE

FAVA MUNOT

GANDHI COUTTS

GANDHI COUTTS

HUANG PARK

HUANG PARRISH-LEWIS

IBRAHIM CRIMI

IBRAHIM SZACILLO

KIM GLADSTONE

KIM PARK

KOROLEV RABTZOW

LUCIANO MUNOT

PAGLINGAYEN GUPTA

PAGLINGAYEN PRZESTRZELSKI

PAGLINGAYEN SZACILLO

PARIKH GLADSTONE

PARIKH RABTZOW

PATEL COUTTS

REVANKAR CRIMI

REVANKAR GUPTA

SONI CRIMI

SONI KUMARAN

SONI SZACILLO

10 second break in between the two (this message is not in the code just wrote it here in word)

CODERS TESTERS

LAST NAME LAST NAME

-----------------------------------------------

CHITNEEDI BALUSAMY

CRIMI BLANKE

ELIAS BLANKE

PARRISH-LEWIS BLANKE

BORKAR DALAL

MUNOT DALAL

MUZYKA DALAL

GLADSTONE FAVA

MUNOT FAVA

COUTTS GANDHI

COUTTS GANDHI

PARK HUANG

PARRISH-LEWIS HUANG

CRIMI IBRAHIM

SZACILLO IBRAHIM

GLADSTONE KIM

PARK KIM

RABTZOW KOROLEV

MUNOT LUCIANO

GUPTA PAGLINGAYEN

PRZESTRZELSKI PAGLINGAYEN

SZACILLO PAGLINGAYEN

GLADSTONE PARIKH

RABTZOW PARIKH

COUTTS PATEL

CRIMI REVANKAR

GUPTA REVANKAR

CRIMI SONI

KUMARAN SONI

SZACILLO SONI

Process finished with exit code 0