*Main.java*

*/\*  
Rohan Parikh  
Coder Tester lab  
29 September 2020 - 19 October 2020  
\*/  
  
  
import* java.io.BufferedReader;  
*import* java.io.File;  
*import* java.io.FileReader;  
*import* java.io.IOException;  
*import* java.util.\*;  
  
*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 = *null*;  
  
  
 *List*<String> studentPairsCodersFirst = *new* ArrayList<>();  
  
 *List*<String> testersGroupInCodersFirstRun = *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();  
 }  
  
  
  
 *// method for coders first* String codersFirst = " ";  
 String testersFirst = " ";  
 *int* i = 0;  
 *while* (i != numOfStudents) {  
 i++;  
 tempeoryStorage = ((*studentsAllCoders*(allStudents, numOfStudents, codersFirst, testersFirst, studentPairsCodersFirst, testersGroupInCodersFirstRun)));  
 }  
 *// studentPairsCodersFirst.add(tempeoryStorage);  
 //testersGroupInCodersFirstRun.add(tempeoryStorage);* System.*out*.println("Print testers first or coders first?");  
 String choice = in.next();  
 *if* (choice.toLowerCase().equals("coders")) {  
 studentPairsCodersFirst.sort(*Comparator*.*comparing*(String::toString));  
 *//Formatting output* String firstLine = String.*format*("%10S %19S ", " Coders", "Testers");  
 System.*out*.println(firstLine);  
 String secondLine = String.*format*("%10S %19S ", " 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*("%10S %19S ", split[0], split[1]);  
 System.*out*.println(names);  
 }  
  
 System.*out*.println(" ");  
 System.*out*.println(" ");  
  
 Collections.*sort*(testersGroupInCodersFirstRun);  
 firstLine = String.*format*("%10S %19S ", " Testers", "Coders");  
 System.*out*.println(firstLine);  
 secondLine = String.*format*("%10S %19S ", " Last Name", " Last Name");  
 System.*out*.println(secondLine);  
 System.*out*.println("-----------------------------------------------");  
 *for* (*int* P = 0; P < testersGroupInCodersFirstRun.size(); P++) {  
 String value = testersGroupInCodersFirstRun.get(P);  
 String[] split = value.split(",");  
 String names = String.*format*("%10S %19S ", split[0], split[1]);  
 System.*out*.println(names);  
 }  
 }  
  
  
 *else* {  
 testersGroupInCodersFirstRun.sort(*Comparator*.*comparing*(String::toString));  
 *//Formatting output* String firstLine = String.*format*("%10S %19S ", " Testers", "Coders");  
 System.*out*.println(firstLine);  
 String secondLine = String.*format*("%10S %19S ", " 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 < testersGroupInCodersFirstRun.size(); P++) {  
 String value = testersGroupInCodersFirstRun.get(P);  
 String[] split = value.split(",");  
 String names = String.*format*("%10S %19S ", split[0], split[1]);  
 System.*out*.println(names);  
 }  
  
 System.*out*.println(" ");  
 System.*out*.println(" ");  
  
 Collections.*sort*(studentPairsCodersFirst);  
 firstLine = String.*format*("%10S %19S ", " Coders", "Testers");  
 System.*out*.println(firstLine);  
 secondLine = String.*format*("%10S %19S ", " Last Name", " Last Name");  
 System.*out*.println(secondLine);  
 System.*out*.println("-----------------------------------------------");  
 *for* (*int* P = 0; P < studentPairsCodersFirst.size(); P++) {  
 String value = studentPairsCodersFirst.get(P);  
 String[] split = value.split(",");  
 String names = String.*format*("%10S %19S ", split[0], split[1]);  
 System.*out*.println(names);  
 }  
 }  
 }  
  
  
  
 *public static* String studentsAllCoders(*List*<String> allStudents, *int* numOfStudents, String codersFirst, String testersFirst, *List*<String> studentPairsCodersFirst, *List*<String> testersGroupInCodersFirstRun) {  
 *int* i = 0;  
 *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);  
 *int* randomIndex2 = (*int*) (Math.*random*() \* studentsAllLength);  
 *if* ((!*usedTester*[randomIndex] && !*usedCoder*[randomIndex2]) && randomIndex != randomIndex2) {  
 codersFirst = allStudents.get(randomIndex2) + "," + allStudents.get(randomIndex);  
 *usedCoder*[randomIndex2] = *true*;  
 *usedTester*[randomIndex] = *true*;  
 studentPairsCodersFirst.add(codersFirst);  
 testersFirst = allStudents.get(randomIndex) + "," + allStudents.get(randomIndex2);  
 testersGroupInCodersFirstRun.add(testersFirst);  
 *return* codersFirst;  
 }  
 }  
 }  
}