ASSIGNMENT 5: (31-37)

#31 Match Details

Write a java program to parse the fixed size format and display the match details. The matches.txt file contains fixed length values using **trim** () method to trim the blank space in the string and create match objects.

Format of the **matches.txt** file,

Attribute	**Start Index**	**Length**
TeamOne Name	0	5
TeamTwo Name	5	5
Venue	10	20
Match Date	30	10

eg:
CSK RCB Chinnasamy Stadium 25-09-2016
MI KKR Eden Gardens 15-10-2016

Create a main class " **Main.java**"

Create **Match class** with below attributes,

- teamOne - String
- teamTwo - String
- venue - String
- matchDate - String

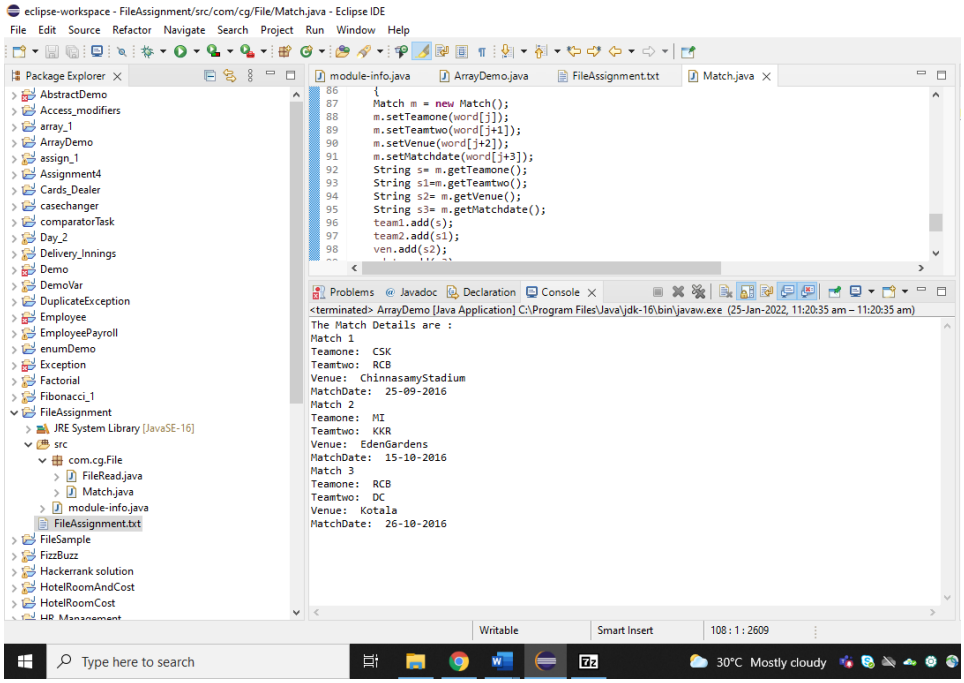
CODE:

```
package com.cg.File;
import java.io.FileReader;
import java.util.*;
import java.io.*;

public class Match {
	private String teamOne;
	private String teamTwo;
	private String venue;
	private String matchDate;

	public void setTeamone(String teamOne)
	{
	this.teamOne=teamOne;
	}
	public void setTeamtwo(String teamTwo)
	{
	this.teamTwo=teamTwo;
	}
	public void setVenue(String venue)
	{
```

```
        this.venue=venue;
        }
        public void setMatchdate(String matchDate)
        {
        this.matchDate=matchDate;
        }
        public String getTeamone()
        {
        return teamOne;
        }
        public String getTeamtwo()
        {
        return teamTwo;
        }
        public String getVenue()
        {
        return venue;
        }
        public String getMatchdate()
        {
        return matchDate;
        }

        public void displayMatch(ArrayList<String> team1,ArrayList<String>
team2,ArrayList<String> ven,ArrayList<String> mdate)
        {
        System.out.println("The Match Details are :");
        int num= team1.size();
        for(int i=0;i<num;i++)
        {
        int a= i+1;
        System.out.println("Match"+" "+ a);
        System.out.println("Teamone: "+ " "+team1.get(i));
        System.out.println("Teamtwo: "+ " "+team2.get(i));
        System.out.println("Venue: "+ " "+ven.get(i));
        System.out.println("MatchDate: "+ " "+mdate.get(i));
        }
        }

        }

        class Main1
        {
        public static void main(String[] args) throws Exception
        {
        File file = new File("FileAssignment.txt");

        BufferedReader br = new BufferedReader(new FileReader(file));

        List<String> lines1 = new ArrayList<String>();
        String line = null;
        while ((line = br.readLine()) != null)
        {
        lines1.add(line);
        }
        br.close();
        String[] lines= lines1.toArray(new String[] {});
        ArrayList<String> team = new ArrayList<String>();
        ArrayList<String> team1 = new ArrayList<String>();
```

```
ArrayList<String> str = new ArrayList<String>();
ArrayList<String> date = new ArrayList<String>();
for(int k=0;k<lines.length;k++)
{
String[] word = lines[k].split("\\s");
for(int j=0;j<word.length;j++)
{
Match m = new Match();
m.setTeamone(word[j]);
m.setTeamtwo(word[j+1]);
m.setVenue(word[j+2]);
m.setMatchdate(word[j+3]);
String obj= m.getTeamone();
String obj1=m.getTeamtwo();
String obj2= m.getVenue();
String obj3= m.getMatchdate();
team.add(obj);
team1.add(obj1);
str.add(obj2);
date.add(obj3);
m=null;
j=word.length;
}
}
Match fin = new Match();
fin.displayMatch(team,team1,str,date);
}
}
```

OUTPUT:



Assignment #32

Sorting Players

Write a Java program to read the player information from text file **(input.txt)** and order the player details based on the cap number and write into another file **(output.csv)** in the same format as the input file.

Input File Format:

<cap number>,<name>,<skill>,<country>,<matches played>

eg:
251,Dhoni,AllRounder,India,443

Create a main class " **Main**. **java**"

Create **Player** class with below attributes,

- capNumber - Integer
- playerName - String
- skill - String
- country - String
- matchesPlayed - Integer

Add appropriate getter and setter methods

Create Player constructor with argument capNumber,playerName, skill, country and matchesPlayed

Create **FileUtility** class to read the information from input.txt file and return the list of player objects

- **List<Player> readFileData(BufferedReader)** - Reads the input file (CSV) parse the information and construct Player objects and return the list of players.

void writeDataToFile(List<Player> playerList) - Write all the players from the player list to the output file after in the same format as the input. Player details string in the CSV format is returned by **toString()** method.

CODE:

```
package com.cg.Fileassignment32;

import java.io.FileReader;

import java.util.*;

import java.io.*;

import java.io.FileNotFoundException;
```

```
import java.io.PrintWriter;
import java.util.*;
import java.io.*;
import java.io.FileNotFoundException;
import java.io.PrintWriter;

//creating player class
public class Sorting {
String capNumber;
String playerName;
String skill;
String country;
String matchesPlayed;

public void setCapnumber(String capNumber)
{
this.capNumber=capNumber;
}
public void setPlayername(String playerName)
{
this.playerName=playerName;
}
public void setSkill(String skill)

{
this.skill=skill;
}
public void setCountry(String country)
{
this.country=country;
}
```

```
public void setMatchesplayed(String matchesPlayed)
{
this.matchesPlayed=matchesPlayed;
}
public String getCapnumber()
{
return capNumber;
}
public String getPlayername()
{
return playerName;
}
public String getSkill()
{
return skill;
}
public String getCountry()
{
return country;
}

public String getMatchesplayed()
{
return matchesPlayed;
}

//creating main method
class FileUtility
{
public static void main(String[] args) throws Exception
{
```

```
FileReader fread = new FileReader("File2.txt");

BufferedReader bread = new BufferedReader(fread);
//Reading the file

List<String> lines1 = new ArrayList<String>();
String line = null;
while ((line = bread.readLine()) != null)
{
lines1.add(line);
}
bread.close();
String[] lines= lines1.toArray(new String[] {});
//To convert the list to array of string to store the information of Match

ArrayList<String> capnumber = new ArrayList<String>();
//Creating different list for storing the information of the team and venue and Matchdate

ArrayList<String> playername = new ArrayList<String>();

ArrayList<String> skill = new ArrayList<String>();
ArrayList<String> country = new ArrayList<String>();
ArrayList<String> matchesplayed = new ArrayList<String>();

for(int k=0;k<lines.length;k++)
{
String[] word = lines[k].split(",");
for(int j=0;j<word.length;j++)
{
Sorting p = new Sorting();
p.setCapnumber(word[j]);
```

```
p.setPlayername(word[j+1]);
p.setSkill(word[j+2]);
p.setCountry(word[j+3]);
p.setMatchesplayed(word[j+4]);
String s= p.getCapnumber();
String s1=p.getPlayername();
String s2= p.getSkill();
String s3= p.getCountry();
String s4= p.getMatchesplayed();
capnumber.add(s);
playername.add(s1);
skill.add(s2);
country.add(s3);
matchesplayed.add(s4);
p=null;
j=word.length;

}
}

try (PrintWriter writer = new PrintWriter("PlayerDetails.csv")) //creating the csv file
{

StringBuilder sb=new StringBuilder();

sb.append(capnumber+","+playername+","+skill+","+country+","+matchesplayed);

writer.write(sb.toString());
}
catch(FileNotFoundException e)
{
```

```
System.out.println(e.getMessage());
}
}
}

}
```

File.txt Output:

```
101,Virat,Allrounder,Pakistan,98
102,Dhoni,Batting,Australia,99
103.Sachin,Bowling,Pakistan,100
```

Assignment #33

Identify Patterns

Write a java program to read the input character stream and identify patterns provided by the user. As the program output display the number of times the pattern occurred in the input character stream.

Read the character stream from the input file (team.txt) that contains the team names and display the output in the console.

Input and Output Format:

- First input corresponds to the number of strings to be searched and followed by each string.
- Refer sample input and output for formatting specifications.
- [All text in bold corresponds to input and the rest corresponds to output]

Sample Input/Output :

Enter number of words

2

Enter the strings to be searched

en

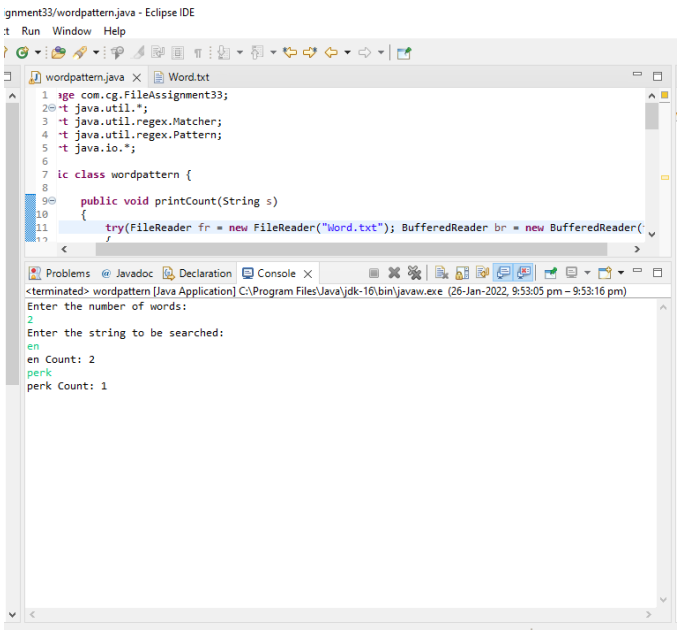
perk

CODE:

```
package com.cg.FileAssignment33;
import java.util.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import java.io.*;

public class wordpattern {

		public void printCount(String str)
		{
			try(FileReader fr = new FileReader("Word.txt"); BufferedReader
br = new BufferedReader(fr))
			{
				String lines = br.readLine();

				int count = 0;
				Matcher match = Pattern.compile(str).matcher(lines);
				while(match.find())
				{
					count++;
				}
```

```
                        System.out.println(str+" Count: "+count);

                }

                catch(IOException e)
                {
                        System.out.println(e);
                }

        }

        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the number of words:");
                int n = sc.nextInt();

                System.out.println("Enter the string to be searched:");
                String s=" ";

                wordpattern iPattern = new wordpattern();
                for(int i=0;i<n;i++) {
                        s = sc.next();
                        iPattern.printCount(s);
                }



        }

}
```

OUTPUT:



Assignment #35

Player Details

Write a java program to record the player details into the file. Get the player details name, teamName and number of matches played from the user and write those information in a comma seperated format (CSV).

Below is the format of the output file.

<name>,<teamName>,<numberOfMatches>

eg: Virat Kohli,Royal Challengers Bangalore,16

Create a main class "Main.java"

Using File class create a new file(player.csv) and write using the OutputStream.

Input and Output Format:

- Get the player details name, teamName and noOfMatches from the user
- [All text in bold corresponds to input and the rest corresponds to output]

Sample Input/Output :

Enter the name of the player

Virat Kohli

Enter the team name

Royal Challengers Banglore

Enter the number of matches played

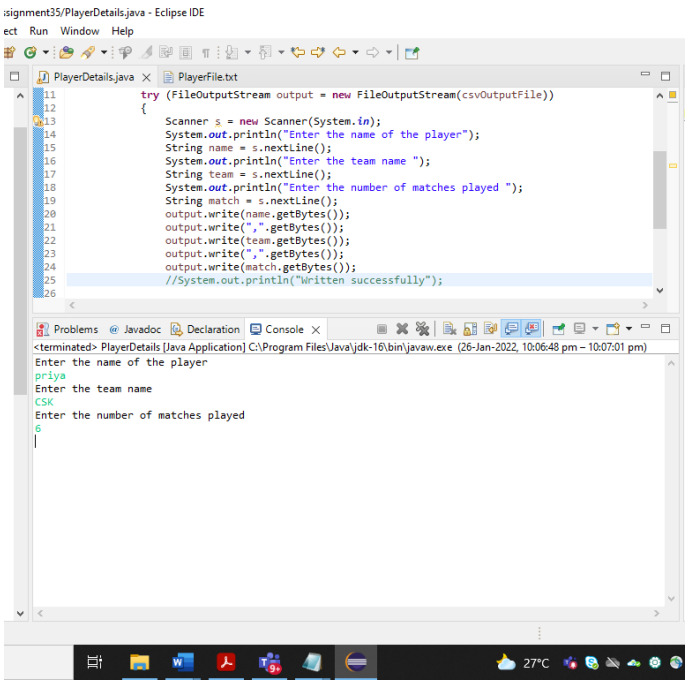
16

CODE:

```
package com.cg.FileAssignment35;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Scanner;

public class PlayerDetails {
		public static void main(String[] args)
		{
			File csvOutputFile = new File("PlayerFile.txt");
			try (FileOutputStream output = new
FileOutputStream(csvOutputFile))
			{
				Scanner str = new Scanner(System.in);
				System.out.println("Enter the name of the player");
```

```
                        String n = str.nextLine();
                        System.out.println("Enter the team name ");
                        String team = str.nextLine();
                        System.out.println("Enter the number of matches played
");
                        String match = str.nextLine();
                        output.write(n.getBytes());
                        output.write(",".getBytes());
                        output.write(team.getBytes());
                        output.write(",".getBytes());
                        output.write(match.getBytes());
                        //System.out.println("Written successfully");

                }
                catch (IOException e)
                {
                        System.out.println(e);
                }
        }

}
```

OUTPUT:



Assignment #37

Sort Scores

Write a Java program to sort scores obtained by the 5 players namely [Virat Kohli, MS Dhoni, Suresh Raina, Gautam Gambir, Ajinkya Rahane] in the recent cricket matches.

Read latest Test, ODI and T20 match scores from the user for each match the scores are provided in a comma separated format. Span a thread for each match and sort the scores in the thread and store the result in the thread object.

Create a main class "Main.java"

Create SortScore class with below attributes,

- matchType - String
- scoreString - String (Comma seperated input scores for a match is stored)
- scores - Integer[]

Create a constructor SortScores with the argument matchType and scoreString.

Extend thread classes and override the run() method

Create a method getScores() to return the scores array.

CODE:

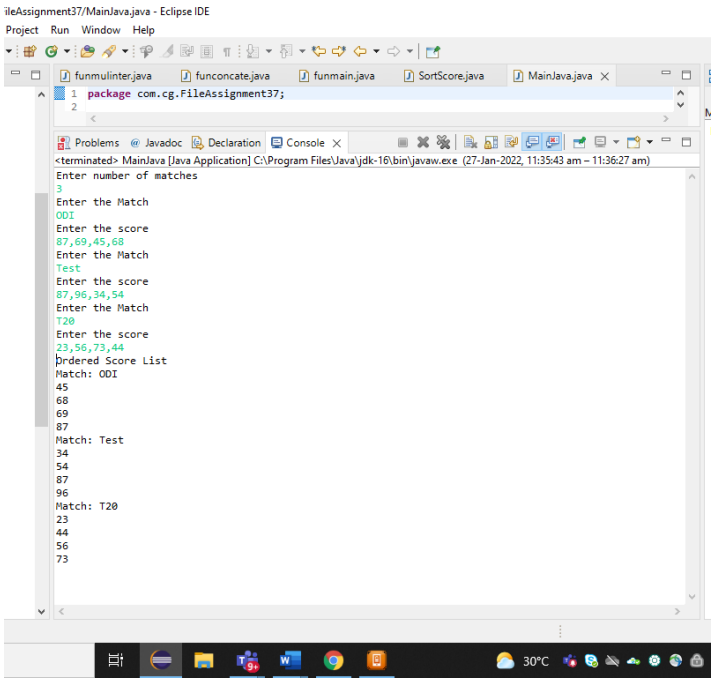
```
package com.cg.FileAssignment37;
import java.util.Arrays;

import java.util.*;

	class SortScore
	{

		public void displayDetails(String[] match,ArrayList<String> score)
		{
			int n=match.length;
			System.out.println("Ordered Score List");

			for(int j=0;j<n;j++)
			{
				System.out.println("Match:"+" "+match[j]);
				ArrayList<String> list = new ArrayList<String>();
				list.add(score.get(j));
				String line[]= list.toArray(new String[] {});

				String input = line[0];
				String[] words = input.split(",");

				int l= words.length;
				int[] arr = new int[l];

				for (int v = 0; v < l; v++) {
					arr[v] = Integer.valueOf(words[v]);
```

```
                        }

                        Arrays.sort(arr);

                        for(int k=0;k<l;k++)
                        {
                              System.out.println(arr[k]);
                        }
                  }

            }
      }
```

Main File:

```
package com.cg.FileAssignment37;

      import java.io.BufferedReader;
      import java.io.IOException;
      import java.io.InputStreamReader;
      import java.util.ArrayList;
      import java.util.Scanner;

      public class MainJava {
            public static void main(String args[]) throws IOException {
                  BufferedReader br =new BufferedReader(new
InputStreamReader(System.in));

                  Scanner sc = new Scanner(System.in);
                  System.out.println("Enter number of matches");
                  int n=sc.nextInt();
                  String[] match = new String[n];
                  ArrayList<String> score = new ArrayList<String>();
                  for(int i=0;i<n;i++) {

                        System.out.println("Enter the Match");
                        match[i]=br.readLine();

                        System.out.println("Enter the score");
                        String scr=br.readLine();
                        score.add(scr);
                  }
                  SortScore str = new SortScore();
                  str.displayDetails(match,score);
                  sc.close();
            }

      }
```

OUTPUT:


ileAssignment37/MainJava.java - Eclipse IDE
Project Run Window Help
funmulinter.java
funconcate.java
funmain.java
SortScore.java
MainJava.java
1 package com.cg.FileAssignment37;
2
Problems Javadoc Declaration Console
<terminated> MainJava [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe (27-Jan-2022, 11:35:43 am – 11:36:27 am)
Enter number of matches
3
Enter the Match
ODI
Enter the score
87,69,45,68
Enter the Match
Test
Enter the score
87,96,34,54
Enter the Match
T20
Enter the score
23,56,73,44
Ordered Score List
Match: ODI
45
68
69
87
Match: Test
34
54
87
96
Match: T20
23
44
56
73
30°C