My Project

AUTHOR
Version
10/04/2017

Table of Contents

Table of contents

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Boys .....
breakup_couple (Contain breakup details) .....
Couple 2
csv_gen .....
gift .....
Girls .....
MyLogger .....
q6 (Main class q4) .....

Class Documentation

Boys Class Reference

Detailed Description

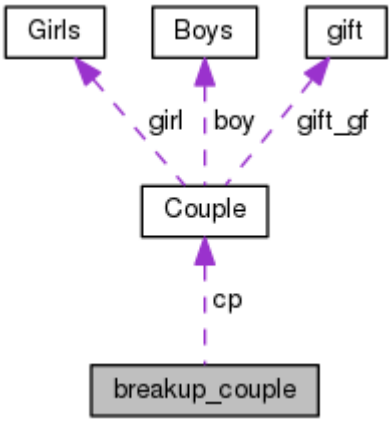
The documentation for this class was generated from the following file:

1 Boys.java

breakup_couple Class Reference

Contain breakup details.

Collaboration diagram for breakup_couple:



IMAGE

Detailed Description

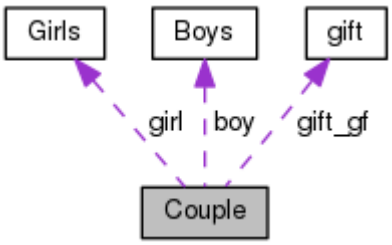
Contain breakup details.

The documentation for this class was generated from the following file:

2 breakup_couple.java

Couple Class Reference

Collaboration diagram for Couple:



IMAGE

Detailed Description

The documentation for this class was generated from the following file:

3 Couple.java

---

csv_gen Class Reference

Detailed Description

The documentation for this class was generated from the following file:

4 csv_gen.java

---

gift Class Reference

Detailed Description

The documentation for this class was generated from the following file:

5 gift.java

Girls Class Reference

Detailed Description

The documentation for this class was generated from the following file:
6 Girls.java

MyLogger Class Reference

Detailed Description

The documentation for this class was generated from the following file:
7 MyLogger.java

q6 Class Reference

Main class q4.

Static Public Member Functions

static void main (String args[]) throws IOException

Detailed Description

Main class q4.

Member Function Documentation

static void q6.main (String *args*[]) throws IOException[static]

Sorting Gift according to price

```
 8                                                                {
 9         int i,j;
10         i=0;
11         j=0;
12         int m=0;
13         Boys b[] = new Boys[1000];
14         csv_gen csv_files = new csv_gen();
15         csv_files.csv();
16         String csvFile = "boy.csv";
17         File out=new File("log.log");
18         String line = "";
19         String csvSplit = ",";
20         BufferedReader buff = null;
21         try{
22             buff = new BufferedReader(new FileReader(csvFile));
23             while((line = buff.readLine() )!= null){
24                 String[] boys_table = line.split(csvSplit);
25
26                 b[i] = new Boys();
27                 b[i].name = boys_table[0];
28                 b[i].intelligence = Integer.parseInt(boys_table[1]);
29                 b[i].attractivness = Integer.parseInt(boys_table[2]);
30                 b[i].min_attr_req = Integer.parseInt(boys_table[3]);
31                 b[i].budget = Integer.parseInt(boys_table[4]);
32                 b[i].status = boys_table[5];
33                 b[i].boy_type= boys_table[6];
34                 b[i].girlf="";
35                 b[i].happiness_boy = 0;
36                 i++;
```

```
            }
        }

        catch(FileNotFoundException e){
            e.printStackTrace();
        }catch(IOException e){
            e.printStackTrace();
        }

        Girls g[] = new Girls[1000];
        csvFile = "girl.csv";
        line = "";
        csvSplit = ",";
        buff = null;


        try{
            buff = new BufferedReader(new FileReader(csvFile));
            while((line = buff.readLine() )!= null){
                String[] girls_table = line.split(csvSplit);

                g[j] = new Girls();
                g[j].name = girls_table[0];
                g[j].attar = Integer.parseInt(girls_table[1]);
                g[j].expense = Integer.parseInt(girls_table[2]);
                g[j].intelligence = Integer.parseInt(girls_table[3]);
                g[j].status = girls_table[4];
                g[j].girl_type = girls_table[5];
                g[j].boyf = "";
                g[j].happiness_girl= 0;

                j++;
            }
        }
        catch(FileNotFoundException e){
            e.printStackTrace();
        }catch(IOException e){
            e.printStackTrace();
        }finally{
                if(buff != null){
                    try{
                        buff.close();
                    }catch(IOException e){
                    e.printStackTrace();
                    }
                }
        }


        gift gf[] = new gift[1000];
        csvFile = "gift.csv";
        line = "";
        csvSplit = ",";
        buff = null;


        try{
            buff = new BufferedReader(new FileReader(csvFile));
            while((line = buff.readLine() )!= null){
                String[] gifts_table = line.split(csvSplit);
            //  System.out.println(boy[0] + boy[1] + boy[2] + boy [3]);
                gf[m] = new gift();
                gf[m].gift_name = gifts_table[0];
                gf[m].gift_price = Integer.parseInt(gifts_table[1]);

                gf[m].gift_value = Integer.parseInt(gifts_table[2]);
                gf[m].gift_type = gifts_table[3];

                m++;

            }
        }
        catch(FileNotFoundException e){
            e.printStackTrace();
        }catch(IOException e){
            e.printStackTrace();
        }finally{
```

```
                if(buff != null){
                    try{
                        buff.close();
                    }catch(IOException e){
                    e.printStackTrace();
                    }
                }
        }

        gift tem;
        int m1,n;
        for(m1=0;m1<m;m1++){
            for(n=m1+1;n<m;n++){
                if(gf[m1].gift_price > gf[n].gift_price){
                    tem= gf[m1];
                    gf[m1]=gf[n];
                    gf[n]=tem;
                }
            }
        }



        int count = 0;
        String output = "";
        int c = 0,d = 0;
        Couple gbcouple[] = new Couple[100];
        for(c=0;c<i;c++){
            for(d=0;d<j;d++){
                if(b[c].is_eligible(g[d].expense,g[d].attar) &&
b[c].status.equals("Single") && g[d].status.equals("Single")){
                    b[c].girlf = g[d].name;
                    g[d].boyf = b[c].name;
                    b[c].status = "Is_committed";
                    g[d].status = "Is_committed";
                    count++;
                    String df= new SimpleDateFormat("dd/MM/yy
HH:mm:ss").format(new Time(System.currentTimeMillis()));
                    output =  df+" "+"Boy : "+b[c].name + " is Committed
with " +"Girl : "+g[d].name + "\n" ;
                    System.out.println(output);
                    try(BufferedWriter buffer=new BufferedWriter(new
FileWriter(out,true))){
                        buffer.write(output);

                    }
                    gbcouple[count-1] = new Couple(b[c],g[d]);

                }
            }

        }
        int o;
        int u;
        Random rn = new Random();
        o = rn.nextInt(count);
        Random rand = new Random();
        u = rand.nextInt(30)+1;
        breakup_couple bk = new breakup_couple(gbcouple, count,
o,output,u);// Least n happy couples broke up
        String outp = "";
        outp = bk.break_print();
        for(i =0 ; i < o; i++)
        {
            for(j = 0;  j < 20; j++)
            {
                if(b[j].status == "Single" && gbcouple[i].girl.status ==
"Single"&& !((b[j].name).equals(gbcouple[i].boy.name)) && b[j].budget >=
gbcouple[i].girl.expense && gbcouple[i].girl.attar >= b[j].min_attr_req) {
                    b[j].status = "Is_committed";
                    gbcouple[i].girl.status = "Is_committed";
                    String fd= new SimpleDateFormat("dd/MM/yy
HH:mm:ss").format(new Time(System.currentTimeMillis()));

                    outp = outp.concat("\n"+fd+ " Day "+ 30/u +" "+"New
Commitment : "+gbcouple[i].girl.name +"   and  "+b[j].name+" \n\n");//New
commitments of broke up girls
```

```
  182
  183                     //  System.out.println("New Commitment after break up:
"+gbcouple[i].girl.name +"-"+b[j].name );
  184                         gbcouple[i].boy = b[j];
  185                         break;
  186                 }
  187             }
  188         }
  189         System.out.println(outp);
  190         try(BufferedWriter buffer=new BufferedWriter(new
FileWriter(out,true))){
  191         buffer.write(outp);
  192
  193         }
  194     }
```

The documentation for this class was generated from the following file:

8 q6.java

Index

INDE