Object Oriented Design Pattern HW2

21400358 백선화

|  |  |  |
| --- | --- | --- |
| 1  2  3  4 | import java.util.\*;  import java.io.FileInputStream;  import java.lang.String;  import java.io.PrintWriter; | [cs](http://colorscripter.com/info#e) |

First, I imported “java.io.FileInputStream” and “java.io.PrintWriter” for using file stream. Instead of using “java.io.FileOutputStream”, I use “PrintWriter” library because it is easier to write user defined string value.

|  |  |  |
| --- | --- | --- |
| 1  2 | FileInputStream inputStream=null;          PrintWriter pw = null; | [cs](http://colorscripter.com/info#e) |

I define “FileInputStream”, “PrintWriter”.

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | try {              inputStream = new FileInputStream("C:/Users/user/workspace/score.txt");              pw = new PrintWriter("C:/Users/user/workspace/output.txt");                                  ...                                  ...          }          catch (Exception e)          {              System.out.println("File input/output error!"+e);          }          finally          {              try              {                  inputStream.close();                  pw.close();              }              catch(Exception e)              {                  System.out.println("File closing error!"+e);              }          }  [Colored by Color Scripter](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

By using try ~ catch, I used exception handling process. While trying input and output streaming process, when error comes out, “catch” statement runs. It means that “File input/output error!” and error information “e” is printed in the console.

And “finally” statements are always executed. If we use input or output stream, stream have to be closed. Therefore after trying and catching inputStream and printWriter which is outputStream should be closed. And it also uses try ~ catch. If program fails to closing each stream, it catches error by printing “file closing error” and error information “e”.

|  |  |  |
| --- | --- | --- |
| 1  2 | inputStream = new FileInputStream("C:/Users/user/workspace/score.txt");              pw = new PrintWriter("C:/Users/user/workspace/output.txt"); | [cs](http://colorscripter.com/info#e) |

I designate input file path and output file path.

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9 | byte[] readBuffer = new byte[inputStream.available()];              while(inputStream.read(readBuffer, 0, readBuffer.length)!=-1){}              String buffer = new String(readBuffer);                String[] line = buffer.split("\n");                for(int i=0;i<line.length;i++){                  System.out.println(line[i]);              }  [Colored by Color Scripter](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

IO stream recognize only byte form. I allocate new byte “readBuffer” in the size of inputStream by using available function. With while loop, variable “readBuffer” reads inputStream as a unit of byte form. After while loop, variable “readBuffer” contains whole contents from inputStream. And “readBuffer” byte variable is converted to string variable “buffer”. Finally, I split “buffer” in new line unit and put buffer onto string array variable “line”. After that, “line[i]” variable contains i-th line of inputStream. (for example, if I print line[1], “sunhwa:baek:80:70:90:78:90” are printed.)

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | float[] total= new float[line.length];              String[] grade = new String[line.length];                for(int i=0;i<line.length;i++){                  String[] word = line[i].split(":");                  total[i] = Float.parseFloat(word[2])\*(float)0.1                          + Float.parseFloat(word[3])\*(float)0.1                          + Float.parseFloat(word[4])\*(float)0.1                          + Float.parseFloat(word[5])\*(float)0.3                          + Float.parseFloat(word[6])\*(float)0.4;                  if(total[i]>=90)                  {                      grade[i]="A";                  }                  else if(total[i]<90 && total[i]>=80)                  {                      grade[i]="B";                  }                  else if(total[i]<80 && total[i]>=70)                  {                      grade[i]="C";                  }                  else if(total[i]<70 && total[i]>=60)                  {                      grade[i]="D";                  }                  else                  {                      grade[i]="F";                  }                }  [Colored by Color Scripter](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

This is (a) problem in assignment. I assigned float value “total” as array with size of length of line. String array variable “grade” also is defined with size of length of line. This is because each line includes one student information and total and grade value should contain each student’s total, grade information.

With for loop from 0 to length of line, I split one line based on “:” and assigned one word unit to string array “word”. After that, for example, string array “word[0]” contains “sunhwa” and “word[2]” contains “90”. And I calculates total score value. (c.f. total[1] contains second line student’s total score.) With if and else~if statement, I stores value of each student’s grade in string array “array”. (c.f. string array “grade[2]” contains grade of third student, such as ”A”.)

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67 | float[] project1 = new float[3];              float[] project2 = new float[3];              float[] project3 = new float[3];              float[] midterm = new float[3];              float[] finalexam = new float[3];              float[] totalscore = new float[3];                  float max=0;              float min=10000000;              float sum=0;                for(int j=2;j<=6;j++){                  max=0;min=1000000;sum=0;                  for(int i=0;i<line.length;i++){                      String[] word = line[i].split(":");                      if(Float.parseFloat(word[j])>max)                      {                          max = Float.parseFloat(word[j]);                      }                      if(Float.parseFloat(word[j])<min)                      {                          min = Float.parseFloat(word[j]);                      }                      sum =sum + Float.parseFloat(word[j]);                  }                  if (j==2){                      project1[0]=max;                      project1[1]=min;                      project1[2]=sum/line.length;                  }                  else if(j==3){                      project2[0]=max;                      project2[1]=min;                      project2[2]=sum/line.length;                  }                  else if(j==4){                      project3[0]=max;                      project3[1]=min;                      project3[2]=sum/line.length;                  }                  else if(j==5){                      midterm[0]=max;                      midterm[1]=min;                      midterm[2]=sum/line.length;                  }                  else if(j==6){                      finalexam[0]=max;                      finalexam[1]=min;                      finalexam[2]=sum/line.length;                  }              }              for(int i=0;i<line.length;i++){                  max=0;min=1000000;sum=0;                  if(total[i]>max)                  {                      max = total[i];                  }                  if(total[i]<min)                  {                      min = total[i];                  }                  sum =sum + total[i];              }              totalscore[0]=max;              totalscore[1]=min;              totalscore[2]=sum/line.length; | [cs](http://colorscripter.com/info#e) |

This is source code for (b) problem. Each float array variable contains highest, lowest and average scores which means that for each float variable, first value contains highest value, second value contains lowest value and third value contains average value. (c.f. midterm[1]contains midterm lowest value.)

As I mentioned upper code, I split each line to unit of word in string array variable “word”. And I find minimum(variable min), maximum(variable max) and summation of scores(variable sum) value. For loop is executed from 2 to 6 because in array “word[]”, 2~6 value contains the score information. After for loop, with if statement I assigned highest(variable max), lowest(variable min) and average(sum/line.length) value to each project and exam array in sequential order. (c.f. “prject3[1]” contains the lowest score of third project.) In the same way, I find the value of total score.

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | int[] numGun = new int[5];              for(int i=0;i<line.length;i++){                  if(grade[i].equals("A"))                      numGun[0]++;                  else if(grade[i].equals("B"))                      numGun[1]++;                  else if(grade[i].equals("C"))                      numGun[2]++;                  else if(grade[i].equals("D"))                      numGun[3]++;                  else                      numGun[4]++;              } | [cs](http://colorscripter.com/info#e) |

This is code for problem (c). Integer array “numGun” contains the number of grade for each student. For example, if two students get “A” grade, numGun[0] has integer value “2” and if five students get “C” grade, numGun[2] has integer value “5”.