Assignment Due Date and Time

21 Nov 2015 (Sat) 11:55 pm

Instructions to Students

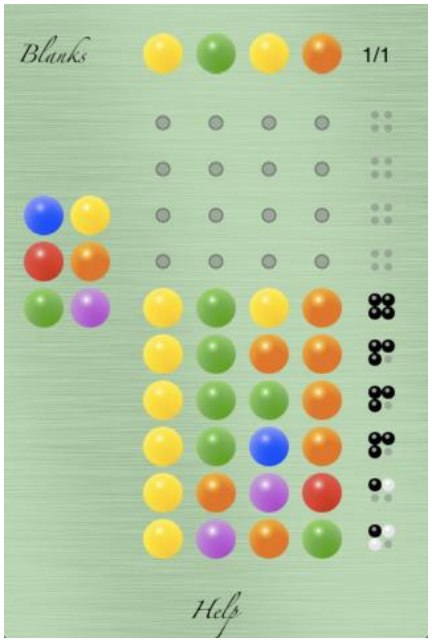
1. This assignment is weighted 40% of the overall Continuous Assessment of this module.
2. This assignment is an individual assignment and should be done by you only. Plagiarism will be treated seriously. Any submitted assignment is found that involved wholly or partly in plagiarism (no matter the assignments are from the original authors or from the plagiarists) will be scored Zero mark and the students involved will be received discipline penalty set by the institute accordingly.
3. Only Java programming language is allowed to develop any required program.
4. Your programs must be well structured. A comment should be included in each class and method to state its main function. Explanation of each variable is also required. The source code must be properly indented. The first few lines in a source file must be a comment stating the name of the source file, your name, class, student number as well as the description of the purposes of the program. Marks will be deducted if any of the above mentioned comment is not included.
5. Grade of your programs will be based on correctness, quality, style, efficiency and the advanced features.
6. You are required to hand in a softcopy of the program source codes upload to Moodle.
7. Remember to backup all your program source codes.
9. Late submission will **NOT** be accepted.

Assignment Specification

You are asked to write a Java Mastermind game.

Information on the Mastermind

The object of this game is to guess the sequence of **four** colored pegs the computer has selected at random from the circled colors ("Blue", "Green", "Orange", "Pink", "Red", "Yellow") in the playing area. ***The colors may be repeated.***



Requirements of the Assignment

Make sure you understand how to play the Mastermind game. For simplicity, we use the numeric digit to represent that color (i.e. 1 - "Blue", 2 - "Green", 3 - "Orange", 4 - "Purple", 5 - "Red", 6 - "Yellow").

Player1 or Computer will set secret pattern and Player2 tries to guess the pattern, in both order and color, within **ten** turns.

Each guess is made by placing a row of code pegs on the decoding board. Once placed, player1 provides feedback by placing from zero to four key pegs in the small holes of the row with the guess. A **black** key peg is placed for each code peg from the guess which is correct in both color and position. A **white** peg indicates the existence of a correct color peg placed in the wrong position.

Functional Requirement

Your program can play in two mode: 'Computer VS Player2' or 'Player1 VS Player2'.

Computer VS Player

Your program will automatically generate an answer for Player2 to guess.

```
Computer VS Player (y/n)? y
--- Player2 Starts to guess ---
Type '0' to give up the game.
Step 1: Please input your guess:
```

Player VS Player

Your program will ask Player1 to input an answer and then Player2 starts to guess.

```
Computer VS Player (y/n)? n
Player1 input the answer: 1123
```

After Player1 input the answer, the screen must be cleared! (You can use a loop to call System.out.println() for 60 times).

Player2 starts to guess

Your program shows a prompt and asks a player to input their guess.

The console will display the first guess with black and white result. If the result is not correct, the program will ask the player to input his/her second guess.

```
--- Player2 Starts to guess ---
Type '0' to give up the game.
Step 1: Please input your guess: 1234
Guess 1: 1234  Black: 1 White: 2
Step 2: Please input your guess: 2211
Guess 2: 2211  Black: 0 White: 3
Step 3: Please input your guess: 2112
Guess 3: 2112  Black: 1 White: 2
Step 4: Please input your guess:
```

If the player's guess is correct, your program indicates that the player wins.

```
Step 4: Please input your guess: 1123
Guess 4: 1123  Black: 4 White: 0
You Win
Do you want to contine (y/n)?
```

The program will keep asking the player to input a guess and display the result until the guess is correct or the number of guesses reaches 10.

If the player cannot get the correct answer after his/her 10th guess, the game shows a message with answer indicating that the player loses the game.

If a game is finished, the program will ask players to continue to play another game or not.

```
Step 8: Please input your guess: 2233
Guess 8: 2233  Black: 1 White: 1
Step 9: Please input your guess: 2211
Guess 9: 2211  Black: 0 White: 3
Step 10: Please input your guess: 2231
Guess 10: 2231  Black: 0 White: 3
You Lose! Correct Answer is 1123
Do you want to continue (y/n)?
```

Player2 is allowed to give up the game by input 0 in the guess and the computer will show he lose the game and give the answer.

```
--- Player2 Starts to guess ---
Type '0' to give up the game.
Step 1: Please input your guess: 1234
Guess 1: 1234  Black: 2 White: 0
Step 2: Please input your guess: 1256
Guess 2: 1256  Black: 1 White: 1
Step 3: Please input your guess: 0
You Lose! Correct Answer is 1631
Do you want to continue (y/n)?
```

Error Handling

Your program must handle following errors:

For the Yes / No question, the program will use y or Y for Yes, n or N for No. If players do not follow this rule, prompt an error message and ask them to input again.

```
Do you want to continue (y/n)?
Please input y or n!
Do you want to continue (y/n)?zzzz
Please input y or n!
Do you want to continue (y/n)?x
Please input y or n!
Do you want to continue (y/n)?
```

For the players input the answer or guess, the input value must be FOUR digits and each digit must be within the range 1 to 6.

```
Player1 input the answer: 398
Use digit '1' to '6' to represent Colour!
Only FOUR pegs are allowed!
Player1 input the answer: 123456
Only FOUR pegs are allowed!
Player1 input the answer: 1237
Use digit '1' to '6' to represent Colour!
Player1 input the answer: abc
Use digit '1' to '6' to represent Colour!
Only FOUR pegs are allowed!
Player1 input the answer:
```

Object Oriented Requirement on Error Handling

You are required to use Object Oriented Programming Skill on Error Handling. Following incomplete codes are the suggestion class and corresponding attributes and methods:

```
public class Error {
  private int errorType;
  private String errorMessage;

  … //missing codes

  public void chkPegsColour(String guess){
    //Error Type = 1,
    //Error Message = Use digit '1' to '6' to represent Colour!
    …
  }

  public void chkPegsNumber(String guess) {
    //Error Type = 2,
    //Error Message = Only FOUR pegs are allowed!
    …
  }

  public void chkYesNoError(String yesno) {
    //Error Type = 3,
    //Error Message = Please input y or n !
    …
  }

  public boolean getError() {
    //return true if any error is occurred
    //return false if no error
    …
  }

  public String getMessage() {
    return errorMessage;
    …
  }

  public String toString() {
    return "Error Type:" + errorType +
            ", Error Message: " + errorMessage;
  }
}
```

Following are the example program to use the above **Error** class:

```
import java.util.Scanner;

public class ErrorTest {
  //Global declaration for Scanner
  public static Scanner sc = new Scanner(System.in);

  public static void main(String args[]) {
    Error err = new Error();
    String choice;
    choice = sc.nextLine();
    err.chkYesNoError(choice);
    if (err.getError()) {
      System.out.println(err.getErrorMessage());
    }
  }

}
```

Testing

You can ease the testing by using 'Copy and Paste' rather than inputting data manually. Prepare a text file, which includes all user inputs in a game run. The following input is an example.

abc
a
n
abc
12345
578
1237
1123
ab345
457
123
12345
2211
2112
1212
1233
1223
1123
a
ab
n

By using Copy and Paste, you can automatically input in the command prompt window.

You can get the result automatically (without the input data echoed).

```
Please input y or n!
Computer VS Player (y/n)? Please input y or n!
Computer VS Player (y/n)? Player1 input the answer: Use digit '1' to
'6' to represent Colour!
Only FOUR pegs are allowed!
Player1 input the answer: Only FOUR pegs are allowed!
Player1 input the answer: Use digit '1' to '6' to represent Colour!
Only FOUR pegs are allowed!
Player1 input the answer: Use digit '1' to '6' to represent Colour!
Player1 input the answer:

 (60 blank lines…)
```

```
--- Player2 Starts to guess ---
Type '0' to give up the game.
Step 1: Please input your guess: Use digit '1' to '6' to represent
Colour!
Only FOUR pegs are allowed!
Step 1: Please input your guess: Use digit '1' to '6' to represent
Colour!
Only FOUR pegs are allowed!
Step 1: Please input your guess: Only FOUR pegs are allowed!
Step 1: Please input your guess: Only FOUR pegs are allowed!
Step 1: Please input your guess: Guess 1: 2211  Black: 0 White: 3
Step 2: Please input your guess: Guess 2: 2112  Black: 1 White: 2
Step 3: Please input your guess: Guess 3: 1212  Black: 1 White: 2
Step 4: Please input your guess: Guess 4: 1233  Black: 2 White: 1
Step 5: Please input your guess: Guess 5: 1223  Black: 3 White: 0
Step 6: Please input your guess: Guess 6: 1123  Black: 4 White: 0
You Win
Do you want to continue (y/n)?Please input y or n!
Do you want to continue (y/n)?Please input y or n!
Do you want to continue (y/n)?Bye Bye!
```

We will use the above method to test your program.

Marking Scheme

Functions

Computer generates the answer	5%
Player1 inputs the answer	5%
Show the guessing results for Player2	
Correct black result	15%
Correct white result	15%
Show Players2 step result	5%
Error Checking	
Yes / No	5%
Invalid digit	5%
Invalid numbers of pegs	5%
OO Requirement	10%
Game finishes within 10 steps	10%
Game correctly shows win/lose message	10%

Style	5%
Documentation (Comments)	5%

Deduction

Cannot pass the Java compiler	-100%
Do not follow the requirement and corresponding output in “Testing”	-30%
More than one Scanner objects in your codes	-20%
Do not use Scanner as the one and only one input method in your codes	-50%
Cannot perform a demonstration to your lecturer	-50%