

Java Objects Part 4 Notes

Team Treehouse

May 25, 2020

1 Exceptions



- *throw new EXCEPTION_NAME*: raises exception *EXCPETION_NAME*
- *try* and *catch*: handles expectations

```
1 public class Game {
2     ...
3     public boolean applyGuess(char letter) {
4         if (misses.indexOf(letter) != -1 || hits.indexOf(letter)
5         != -1) {
6             throw new IllegalArgumentException(letter + " has
7             already been guessed"); // <- this little guy here :)
8         }
9     }
10    ...
11 }
```

```
12 }  
13
```

Listing 1: lesson_01/Game.java

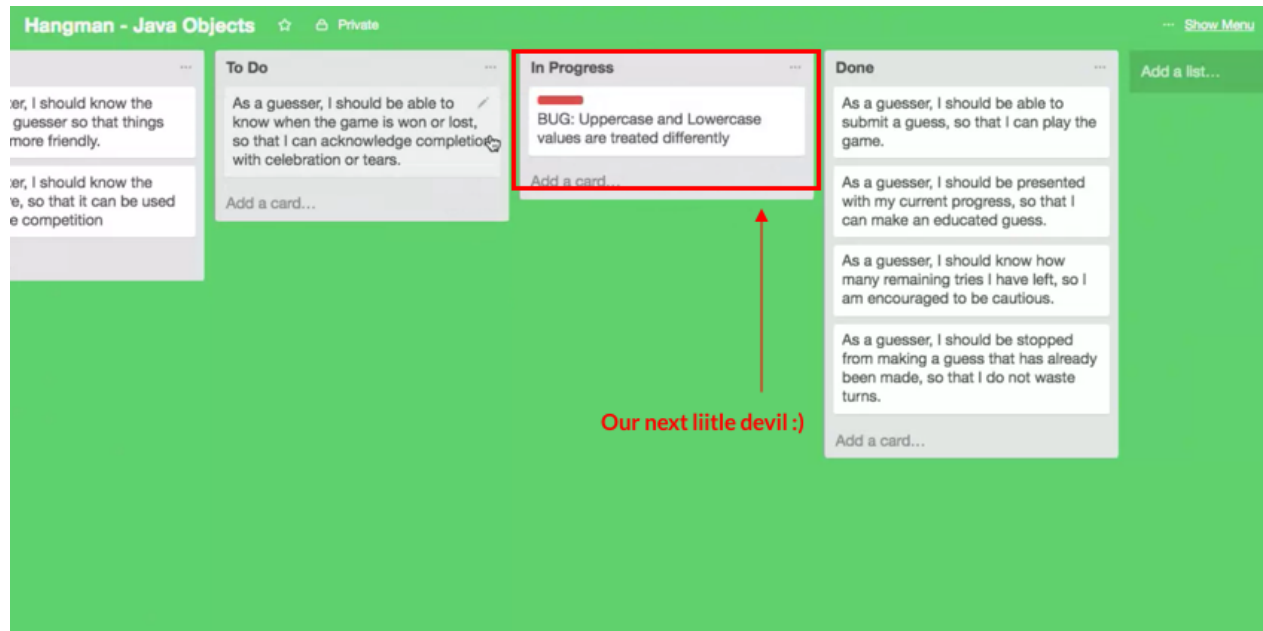
```
1  import java.util.Scanner;  
2  
3  public class Prompter {  
4      ...  
5      public boolean promptForGuess() {  
6          ...  
7          boolean isHit = false;  
8          try { // <- And this little guy here :)  
9              isHit = game.applyGuess(guess);  
10         } catch (IllegalArgumentException iae) {  
11             System.out.println(iae.getMessage());  
12         }  
13  
14         return isHit;  
15     }  
16 }  
17
```

Listing 2: lesson_01/Prompter.java

Notes:

- Files can be compiled and displayed by typing *javac Example.java* && *java Example* in terminal

2 Validating and Normalizing User Input



- *Character.toLowerCase(CHAR_VAR)*: turns value in *CHAR_VAR* to a lowercase character

```

1      public class Game {
2          ...
3          private char normalizeGuess(char letter) {
4              ...
5              letter = Character.toLowerCase(letter); // <- This little
guy here :)
6              ...
7          }
8      }
9

```

Listing 3: lesson.02/Game.java

Notes:

- Files can be compiled and displayed by typing *javac Example.java* && *java Example* in terminal

3 Exercise 2

- Solution included in *exercise_2.java*

4 Using Method Overloading

