Assignment\_1\_Java\_Exception

Due date: 10/26 2016 Wednesday

Course: NJIT Java, Android, Data structures IT114

Instructor: ilker Kiris

Student: <your Name and Last Name goes here>

**SETUP and how to submit**

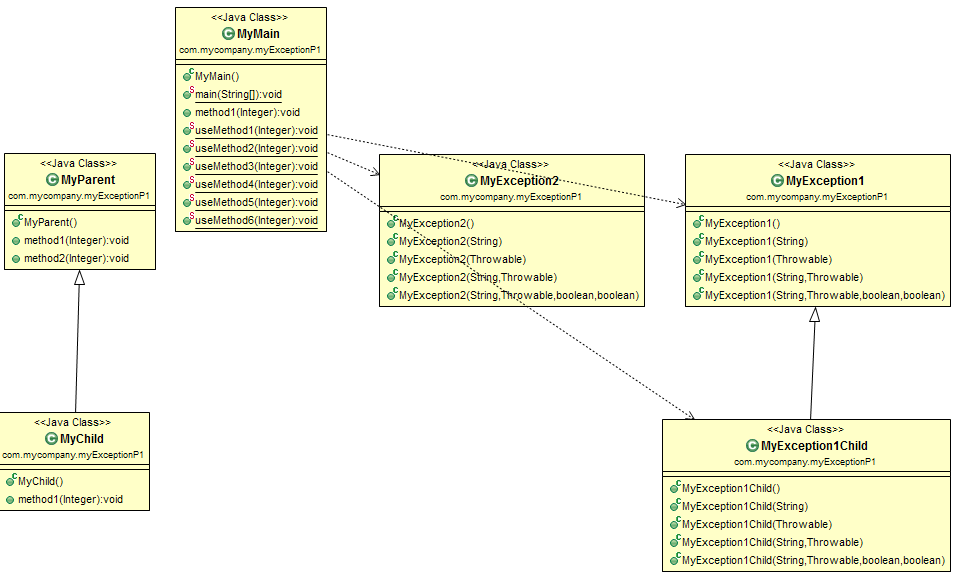
1. Create Java Project (highly suggesting to create it in STS) and put the java packages into a source directory called src
2. Add README.md file at root (top) level of project explaining what is happening
3. Create a class diagram (highly suggesting to install ObjectAid eclipse plugin to STS and using that to create class diagram and an image of the class diagram and put them to directory called *design* under root level)
4. Create github account, create an organization called it114-assignments-yournameLastname
5. Create a github repository called assignment-java-exception-yournameLastname in above organization
6. Create a git repository of your project, commit your code (first add to index then commit) to local repository, then set above created github repository as remote repository for the project and finally push to remote github repository
7. Email ilker the questions and your answers. And link to your above created github repository.

**CODING**

Create a Java project with classes like depicted in below class diagram, that;

1. create Checked Exceptions (like MyException1, MyException2, MyException1Child),
2. create Checked Exception that is child of another Checked Exception,
3. have a method that throw these checked Exceptions (like Main.method1),
4. handle checked Exceptions thrown by called methods via either try/catch or throws,
5. use checked Exception inheritance in overwriting methods in Child classes that throws these Exceptions,
6. wrap an Exception into another Exception (via its constructor that takes a Throwable) and throw that

**Example class diagram**;



**Questions**;

1. **What are 3 top classes used in Java to represent failures?**
2. Errors, Runtime(Un-checked) Exceptions, Checked Exceptions
3. **What is the difference between checked and unchecked exceptions?**
4. Runtime (Un-checked) Exceptions are errors that occur due to normal operations. For example when the user enters a string in lieu of an expected integer input.
5. **What are Errors? Give couple of example Java Error classes.**
6. Can be syntax error, For example naming java file different name from class file.
7. **What are unchecked Exceptions? Give couple of example Java unchecked exceptions**
8. Apart from the example above, it can also occur due to programing error, for example: having an array of set integer yet calling out of box index;
9. **What are checked Exceptions? Give couple of example Java checked exceptions**
10. import java.io.File;
11. import java.io.FileReader;
12. public class FilenotFound\_Demo {
13. public static void main(String args[]) {
14. File file = new File("E://file.txt");
15. FileReader fr = new FileReader(file);
16. }
17. }
18. **How can you know an Exception is unchecked Exception?**
19. If the error persists at Runtime
20. **How can you know a failure is representing an Error?**
21. Usually the program terminates
22. **How can you know an Exception is checked Exception?**
23. When it occurs at compile time – they cannot be ignored.
24. **What are 2 ways to “handle” a checked Exception that can be thrown by block of code in a method?**
26. public Throwable getCause()
28. public String getMessage()
29. **How can a method signal a failure to its caller?**
30. **When body of a method can throw a checked Exception, does its signature reflect that? How?**
31. Yes, via try / return value
32. **What are “catch” blocks for?**
33. if the block enclosed by the try generates an exception of this type, it is caught and handled here
34. **Does the order of Exceptions specified in a try/catch block matter?**
35. Yes, the order must start with ‘try’
36. **When a child class is overwriting a method of parent that throws a checked Exception, can it throw a more generic Exception?**
37. Can you chain Exceptions?