Esercitazione 6

Programmazione 2

Esercizio 1

Documentare le classi sviluppate per l'esercizio "carte da gioco" con il tool javadoc:

- Aggiungere all'interno dei file sorgenti la documentazione di ciascuna classe, variabile e metodo
- Produrre la documentazione in formato html utilizzando il tool javadoc

Riferimenti javadoc

http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html

Eckel, "Thinking in Java":

-->Everything

is an Object --> Comments and embedded documentation

www.oracle.com: documentazione jdk

Eccezioni

Scopo: gestire le situazioni erronee che sorgono durante l'esecuzione di un programma.

3 tipi di eccezioni:

- checked exception
- error
- exception (unchecked exceptions)

Checked exceptions

Invocare un metodo che dichiara la possibilità di lanciare una *checked exception*, obbliga a:

- gestire l'eccezione o
- propagarla a livello superiore

Sono checked exception gli oggetti di tipo java.lang.Exception che non sono di tipo java.lang.RuntimeException.

Checked exceptions

java.lang

Class Exception

java.lang.Object
  java.lang.Throwable
    java.lang.Exception

```
public class Exception
extends Throwable
```

The class `Exception` and its subclasses are a form of `Throwable` that indicates conditions that a reasonable application might want to catch.

The class `Exception` and any subclasses that are not also subclasses of `RuntimeException` are *checked exceptions*. Checked exceptions need to be declared in a method or constructor's `throws` clause if they can be thrown by the execution of the method or constructor and propagate outside the method or constructor boundary.

fonte: estratto da http://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html

Unchecked exceptions

java.lang

Class RuntimeException

java.lang.Object
  java.lang.Throwable
    java.lang.Exception
      java.lang.RuntimeException

public class **RuntimeException**
extends Exception

RuntimeException is the superclass of those exceptions that can be thrown during the normal operation of the Java Virtual Machine.

RuntimeException and its subclasses are *unchecked exceptions*. Unchecked exceptions do *not* need to be declared in a method or constructor's throws clause if they can be thrown by the execution of the method or constructor and propagate outside the method or constructor boundary.

fonte: estratto dahttp://docs.oracle.com/javase/7/docs/api/java/lang/RuntimeException.html

esempio: classe java.io.File

createNewFile

public boolean createNewFile()
  throws IOException

Atomically creates a new, empty file named by this abstract pathname if and only if a file with this name does not yet exist. The check for the existence of the file and the creation of the file if it does not exist are a single operation that is atomic with respect to all other filesystem activities that might affect the file.

Note: this method should *not* be used for file-locking, as the resulting protocol cannot be made to work reliably. The FileLock facility should be used instead.

Returns:

true if the named file does not exist and was successfully created; false if the named file already exists

Throws:

IOException - If an I/O error occurred

SecurityException - If a security manager exists and its SecurityManager.checkWrite(java.lang.String) method denies write access to the file

Since:

1.2

fonte: estratto da http://docs.oracle.com/javase/7/docs/api/java/io/File.html

Eccezioni

```
/**
 * ritorna un nuovo file di log in /home/logs/log.txt.
 * Se il file e' gia' presente, non viene
 * sovrascritto e il metodo torna null.
 *
 * @return il nuovo file di log o null se il file e' gia'
 *          presente.
 */
private File getLogFile() {
    File logFile = new File("/home/logs/log.txt");
    boolean fileCreato = false;
    fileCreato = logFile.createNewFile();
    if (fileCreato) {
        return logFile;
    } else {
        return null;
    }
}
```

1. gestione dell'eccezione

```
/**
 * ritorna un nuovo file di log in /home/logs/log.txt.
 * Se il file e' gia' presente, non viene
 * sovrascritto e il metodo torna null.
 *
 * @return il nuovo file di log o null se il file e' gia'
 *          presente o non e' possibile crearlo.
 */
private File getLogFile() {
    File logFile = new File("/home/logs/log.txt");
    boolean fileCreato = false;
    try {
        fileCreato = logFile.createNewFile();
    } catch (IOException ex) {
        ex.printStackTrace();
        //istruzioni da eseguire per gestire l'eccezione
        fileCreato = false;
    }
    if (fileCreato) {
        return logFile;
    } else {
        return null;
    }
}
```

2. propagazione dell'eccezione

```
/**
 * ritorna un nuovo file di log in /home/logs/log.txt.
 * Se il file e' gia' presente, non viene
 * sovrascritto e il metodo torna null.
 *
 * @return il nuovo file di log o null se il file e' gia'
 *          presente.
 * @throws IOException se viene generato un errore di I/O
 */
private File getLogFile() throws IOException {
    File logFile = new File("/home/logs/log.txt");
    boolean fileCreato = false;
    fileCreato = logFile.createNewFile();
    if (fileCreato) {
        return logFile;
    } else {
        return null;
    }
}
```

Il metodo che invoca getLogFile é delegato a gestire l'eccezione

Riferimenti

Eckel, Thinking in Java
  --> "Error Handling with Exceptions"

The Java Tutorial (www.oracle.com)
  --> Essential Java Classes
  --> Exceptions

Esercizio

Implementare un sistema di classi che permetta di registrare delle voci composte da: data, descrizione e quantità. Eventuali errori nell'inserimento delle voci (date non valide, …) devono generare un'eccezione (modellare una classe di eccezioni che specializzi java.lang.Exception).

Esercizio

Implementare un'applicazione grafica che permetta all'utente di inserire delle voci composte da: data, descrizione e quantità. Eventuali errori nell'inserimento dei dati (date non valide, …) possono essere intercettati e registrati su un pannello laterale. L'utente deve poter scegliere se registrare gli errori oppure no tramite componente a scelta (radio button, menu' a tendina...).

Esercizio