Corso di Programmazione
Anno Accademico 2014–2015
Prof. Giovanni Pighizzini

Note ed esercizi aggiuntivi

12. Ancora eccezioni

Esempio. Esempio di codice che può sollevare svariati problemi in esecuzione (individuarli tutti!)

```
class Prova {
  public static void main(String[] a) {
    int somma = 0;
    for (String s: a)
      somma = somma + s.length();
    System.out.println(somma);
    System.out.println(somma / a.length);
    Integer n = new Integer(a[1]);
    System.out.println(10 / n);
  }
}
```

Lo stesso codice con un blocco `try-catch` per intercettare le singole eccezioni.

```
class Prova {
  public static void main(String[] a) {
    int somma = 0;
    for (String s: a)
      somma = somma + s.length();
    System.out.println(somma);
    try {
      System.out.println(somma / a.length);
      Integer n = new Integer(a[1]);
      System.out.println(10 / n);
    } catch (ArithmeticException e) {
      if (a.length > 0)
        System.out.println("Il secondo argomento deve essere un numero diverso da 0");
      else
        System.out.println("Sono richiesti almeno due argomenti");
    } catch (NumberFormatException e) {
      System.out.println("Il secondo argomento deve essere un numero");
    } catch (ArrayIndexOutOfBoundsException e) {
      System.out.println("Sono richiesti almeno due argomenti");
    }
  }
}
```

Esempio. Visualizza sul monitor il contenuto di un file di caratteri, il cui nome viene specificato sulla riga di comando.

```
import java.io.FileReader;
import java.io.IOException;

class VisualizzaFile {

  public static void main(String[] args) throws IOException {
    //costruzione dello stream di caratteri
    FileReader frd = new FileReader(args[0]);

    int i;
    //lettura e visualizzazione
    while ((i = frd.read()) != -1)
      System.out.print((char)i);

    //chiusura dello stream
    frd.close();
  }
}
```

Note
Il costruttore di `FileReader` può sollevare una `FileNotFoundException` (sottoclasse di `IOException`). Il metodo `read` e il metodo `close` possono sollevare una `IOException`. Poiché si tratta di *eccezioni controllate*, `main` deve intercettarle con un costrutto `try-catch` oppure delegarle esplicitamente al chiamante. Qui si è scelta la seconda alternativa: la delega esplicita al chiamante è espressa scrivendo `throws IOException` alla fine dell'intestazione di `main`.

Esercizio 12.1
Modificate l'applicazione `VisualizzaFile` in modo che nel caso il file non esista o vi siano problemi durante la lettura, visualizzi un messaggio di errore e termini. A tale scopo intercettate opportunamente le eccezioni.