5. Handling & Exceptions

Exceptions

- What if something goes wrong? → Program Termination?!?! With every Error?
  → Obviously not intelligent!

- Mechanism in Java: Exceptions

- Definition [JTutorial]:
  "An exception is an event, which occurs during the execution of a program,
  that disrupts the normal flow of the program’s instructions."

- Exceptions are like balls that are thrown when something unusual occurs.
  Somebody must catch the ball and handle the exception or the program must terminate.
Exceptions

- Usual case: Methods "try" possibly dangerous code and "catch" (handle) correspondingly resulting exceptions themselves

```java
try {
    // ...
    FileWriter fileWriter = new FileWriter("someFileName.txt");
    fileWriter.write('a');
    // ...
}

} catch (IOException e) {
    // Exception handling code goes here
    e.printStackTrace();
}

} catch (SomeOtherException e) {
    // Exception handling code goes here
    e.printStackTrace();
}

} catch (Exception e) {
    // Exception handling code goes here
    e.printStackTrace();
}
```

Exceptions

- Other possibility: Let others (callers) handle the problem!
  Add `throws` clause to method/constructor definition

```java
class InvalidGearException extends Exception {}
class TireExplodedException extends Exception {}

class Bicycle {
    int gear;

    Bicycle(int initialGear) throws InvalidGearException {
        if (initialGear > 0) {
            gear = initialGear;
        } else {
            throw new InvalidGearException();
        }
    }

    void inflateTires() throws TireExplodedException {
        // ...
    }
}
```

6 Coding & Naming Conventions

Deepening readings:
import java.io.RandomAccessFile;

public class ExceptionsDemo {
    public static void main(String[] args) {
    }
}

import java.io.RandomAccessFile;

public class ExceptionsDemo {
    public static void main(String[] args) {
    }
}
Grundlage ist unser spezielle Fehlerbearbeitung für FileNotFoundException.

Hier ist unsere Fehlerbearbeitung:

```java
public static void main(String[] args) {
    try {
        BufferedReader reader = new BufferedReader(new FileReader("G:\\datei.txt"));
        String line = reader.readLine();
        System.out.println(line);
    }
    catch (FileNotFoundException e) {
        e.printStackTrace();
        System.out.println("Hier ist unsere Fehlerbearbeitung.");
    }
    finally {
        System.out.println("Ende der Datei-Verarbeitung.");
    }
}
```
Deepening readings:

http://www.oracle.com/technetwork/java/codeconv-138413.html
http://geosoft.no/development/javastyle.html
http://docs.oracle.com/javase/1.5.0/docs/guide/javadoc/index.html

**Indentation** (4 spaces) and **line length** (80 characters):

```java
public void initializeMatrix(int[][] someMatrix) {
    for (int i=0; i<someMatrix.length; i++) {
        for (int j=0; j<someMatrix.length; j++) {
            if (i%2==0) {
                if (i==2*j) {
                    someMatrix[i][j]=7;
                } else {
                    someMatrix[i][j] = 3*i+72*j + (int)Math.floor(i/(j+1));
                }
            }
        }
    }
}
System.out.println("This is the end!");
}
```
• **Indentation** (4 spaces) and **line length** (80 characters):

```java
public void initializeMatrix(int[][] someMatrix) {
    for (int i = 0; i < someMatrix.length; i++) {
        for (int j = 0; j < someMatrix.length; j++) {
            if (i % 2 == 0) {
                if (i == 2 * j) {
                    someMatrix[i][j] = 7;
                } else {
                    someMatrix[i][j] = 3 * i + 72 * j +
                        (int)Math.floor(i / (j + 1));
                }
            }
        }
    }
    System.out.println("This is the end!");
}
```

Tip: Use IDE/editor with syntax highlighting!

• **Names of variables and methods**
  • **Short** and **meaningful**
  • **Indicate to the casual observer the intent of its use**
  • **Mixed case with the first letter lowercase, with the first letter of each internal word capitalized**
  • **Booleans with prefix** "is"
  • **Examples**:
    ```java
    int rowIndex
    int columnIndex
    boolean isValid
    static boolean copyFile(URL from, URL to)
    ```

• **Names of constants**
  • **Uppercase with words separated by underscores**
    ```java
    static final int MIN_ALLOWED_VALUE = 42;
    ```

• **Names of classes and interfaces**
  • **Short** and **meaningful**
  • **Nouns in mixed case with the first letter lowercase, with the first letter of each internal word capitalized**
  • **Good style: Use only [a..z] and [A..Z]**
  • **Examples**:
    ```java
    class Bicycle
    class ImageSprite
    interface RasterDelegate
    interface Comparable
    ```

• **Use English for everything**, e.g. names, comments etc.

• **Adhere STRICKLY to common coding styles**
  ➔ **Do not use exotic "own" coding styles!**

• **Use default style of DIE**
  (Integrated Development Environment, e.g. Eclipse)

• **Use comments REGULARLY and THOROUGHLY**

• **Use Javadoc** (not only for larger projects)
7 Using the Java Class Library

Deepening readings (optional):

http://docs.oracle.com/javase/tutorial/java/data/index.html
http://docs.oracle.com/javase/tutorial/java/generics/index.html

Main reference:

http://docs.oracle.com/javase/6/docs/api/

7 Using the Java Class Library

Packages and Imports

- Java's classes and types are organized in hierarchical packages

java.lang.String
java.net.URLConnection
java.util.Collection
javax.xml.parsers.SAXParser
org.w3c.dom.events.DocumentEvent

- All types from package `java.lang` are imported automatically

- Other types need to be imported

import java.net.URLConnection;
import java.util.Collection;
import some.other.package.*;  // * means all types
7 Using the Java Class Library

Packages and Imports

- Java's classes and types are organized in hierarchical packages
  - `java.lang.String`
  - `java.net.URLConnection`
  - `java.util.Collection`
  - `javax.xml.parsers.SAXParser`
  - `org.w3c.dom.events.DocumentEvent`

- All types from package `java.lang` are imported automatically
- Other types need to be imported
  ```java
  import java.net.URLConnection;
  import java.util.Collection;
  import some.other.package.*;  // * means all types
  ```

Generics

- Some types may be parameterized with other types
- Typical examples are classes that implement data structures
- Advantages:
  - Type checks at compile time
  - Programmers can implement algorithms generically
- Examples:
  ```java
  Vector nonGenericVector = new Vector();
  nonGenericVector.add( 1234 );
  nonGenericVector.add( "Hello world" );
  Object typeUnknown = nonGenericVector.get(0);

  Vector<Bicycle> bicycles = new Vector<Bicycle>();
  bicycles.add( new Bicycle() );
  bicycles.add( 123 );  // Compile time error!
  Bicycle typeKnown = bicycles.get(0);
  ```

Wrapper-Classes

- Parameters restricted to reference types (classes, interfaces)
- For each primitive type, a corresponding wrapper class exists
- Examples:
  ```java
  java.lang.Short
  java.lang.Integer
  java.lang.Long
  java.lang.Float
  java.lang.Double
  ```
- Primitive types are automatically boxed and unboxed when necessary
  ```java
  Integer i = 723;
  int j = i;
  ```
8 Solving a Problem

Example Task

- Example task:
  Load an image from the internet and display that image

- How would you start?
  - Google is your friend
  - So is the Java Tutorial
  - Look up corresponding classes, fields and methods in API documentation
  - Write program
  - Test program
  - Repeat all of the above as long as necessary
  - DONE 😊

Deepening readings (optional):

The Internet

Main reference:

http://docs.oracle.com/javase/6/docs/api/

8 Solving a Problem

Example Task

- [Image: Loading an Image using a URL]

- [Image: Reading an Image]

- [Image: Writing an Image]
8 Solving a Problem

Example Task

Reading/Loading an Image

When you think of digital images, you probably think of sampled image formats such as the PNG image format used in digital photography. Most graphics commands used in web pages. All programs that can use these images must first convert from their external format into an internal format.

Java supports loading these external image formats into text-based or raster image formats such as PNG, JPEG, GIF, ICO, BMP, PICT, PDF, and XBM formats. Java also supports the GIF, JPEG, and PNG formats. For example, plug-ins for TIFF and JPEG 2000 are currently available.

To load an image from a specific file, use the following code:

```java
import java.io.File;
import java.io.IOException;

public class ImageLoader {
    public static void main(String[] args) {
        try {
            ImageIO.write(ImagineImageReader.read(new File("example.png")), "png", new File("example.png"));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Image I/O recognizes the contents of the file as a JPEG format image, and writes it into a new image class which can be directly used by Java 2D.

If the code is running on a Mac, there is just one way to obtain the image from the application:

```java
public class ImageLoader {
    public static void main(String[] args) {
        try {
            ImageIO.read(ImagineImageReader.read(new File("example.png")));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

The general-purpose method used in this example returns the URL of the directory containing this applet.

The following example shows how to use the getOutputImage method to load the strawberry.jpg file.
8 Solving a Problem

Example Task

So what about displaying the image we've just loaded?

The previous method used in this example returns the URL of the directory containing this applet.

The following example shows how to use the previous method to load the interactive applet.

```
import java.awt.Image;
import java.awt.Graphics;
import java.net.URL;
import javax.imageio.ImageIO;

public class ImageDemo extends Component {
    public void paint(Graphics g) {
        g.drawImage(ImageIO.read(new URL("image.png"))), 0, 0, null);
    }
    public Dimension getPreferredSize(int i) {
        return new Dimension(50, 50);
    }
    public static void main(String[] args) {
        JFrame frame = new JFrame("Image Demo");
        frame.setSize(500, 500);
        frame.setVisible(true);
    }
}
```
References

- Code examples taken from
  - Stack Overflow, http://stackoverflow.com