

Script generated by TTT

Title: Lehmann: Uebung_Einf_HF (06.07.2012)
Date: Fri Jul 06 09:16:59 CEST 2012
Duration: 77:26 min
Pages: 84

The screenshot shows a Microsoft Word document window titled 'ttt/TTT' with the file name 'Introduction to Java Basics.pptx'. The slide is titled '5 Error Handling & Exceptions'. Below the title, there is a section titled 'Deepening readings:' with four links:
http://en.wikipedia.org/wiki/Exception_handling
<http://docs.oracle.com/javase/tutorial/essential/exceptions/>
<http://docs.oracle.com/javase/tutorial/essential/exceptions/runtime.html>
<http://docs.oracle.com/javase/tutorial/essential/exceptions/advantages.html>

5 Handling & Exceptions

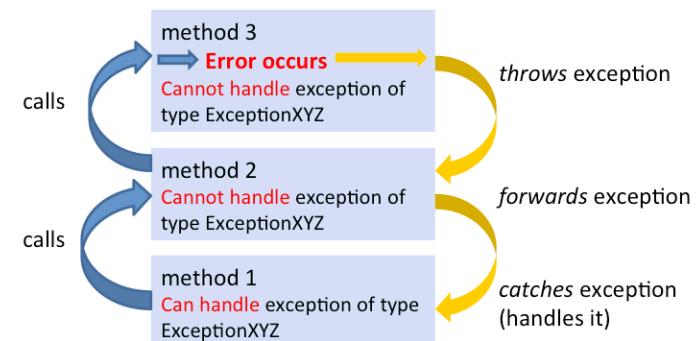
Exceptions

- 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.

5 Handling & Exceptions

Exceptions

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



5 Handling & Exceptions

Exceptions

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

```
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();
}
```

5 Handling & Exceptions

Exceptions

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

```
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 {
        // ...
    }
}
```



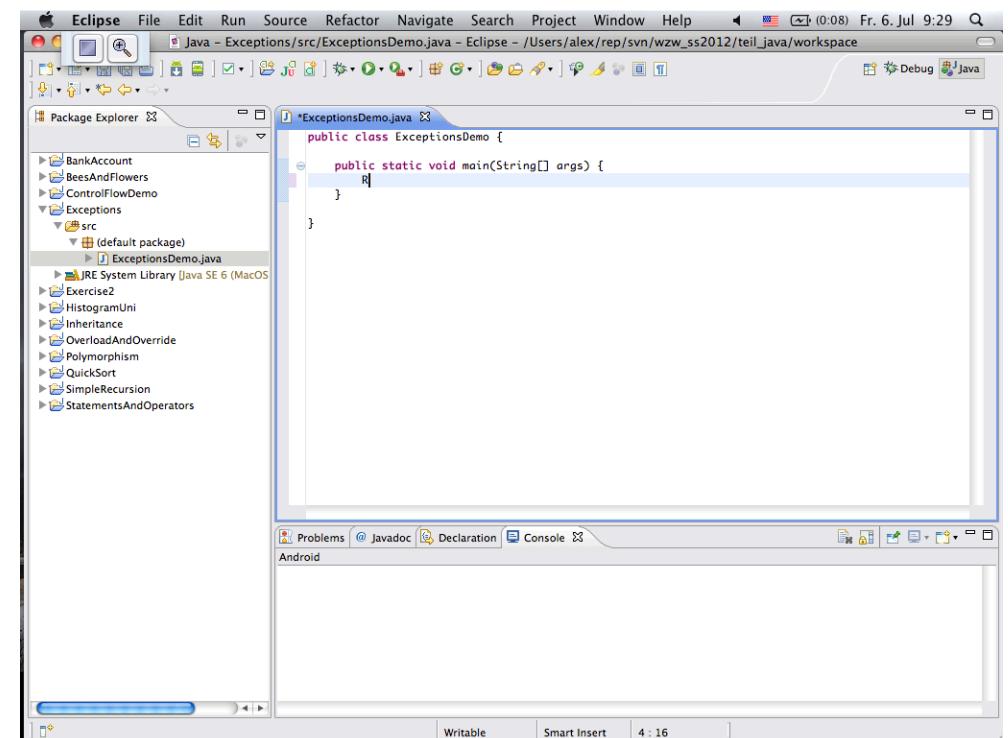
6 Coding & Naming Conventions

Deepening readings:

```
    if (initialGear > 0) {
        gear = initialGear;
    } else {
        throw new InvalidGearException();
    }

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

}
```



Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:30

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.RandomAccessFile;
public class ExceptionsDemo {
    public static void main(String[] args) {
        RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt");
    }
}
```

Problems Javadoc Declaration Console

Android

Writable Smart Insert 6 : 83

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:30

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.RandomAccessFile;
public class ExceptionsDemo {
    public static void main(String[] args) {
        RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt");
    }
}
```

Problems Javadoc Declaration Console

Android

Unhandled exception type FileNotFoundException

Writable Smart Insert 6 : 86

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:31

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.RandomAccessFile;
public class ExceptionsDemo {
    public static void main(String[] args) {
        RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");
    }
}
```

Problems Javadoc Declaration Console

Android

Unhandled exception type FileNotFoundException

Writable Smart Insert 6 : 87

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:31

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;
import java.io.RandomAccessFile;

public class ExceptionsDemo {
    public static void main(String[] args) {
        try {
            RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");
        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

Problems Javadoc Declaration Console

Android

Writable Smart Insert 10 : 16

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:33

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;
import java.io.RandomAccessFile;

public class ExceptionsDemo {
    public static void main(String[] args) {
        try {
            RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere Fehlerbehandlung.");
        }
    }
}
```

Problems

```
<terminated> ExceptionsDemo [Java Application] /System/Library/java/javaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
java.io.FileNotFoundException: DieseDateiGibtEsNicht.txt (No such file or directory)
at java.io.RandomAccessFile.open(Native Method)
at java.io.RandomAccessFile.<init>(RandomAccessFile.java:216)
at java.io.RandomAccessFile.<init>(RandomAccessFile.java:101)
at ExceptionsDemo.main(ExceptionsDemo.java:8)
```

Hier ist unsere Fehlerbehandlung.

The value of the local variable file is not used

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:33

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;
import java.io.RandomAccessFile;

public class ExceptionsDemo {
    public static void main(String[] args) {
        try {
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere Fehlerbehandlung.");
        }
    }
}
```

Problems

```
<terminated> ExceptionsDemo [Java Application] /System/Library/java/javaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
java.io.FileNotFoundException: DieseDateiGibtEsNicht.txt (No such file or directory)
at java.io.RandomAccessFile.open(Native Method)
at java.io.RandomAccessFile.<init>(RandomAccessFile.java:216)
at java.io.RandomAccessFile.<init>(RandomAccessFile.java:101)
at ExceptionsDemo.main(ExceptionsDemo.java:8)
```

Hier ist unsere Fehlerbehandlung.

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:34

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;

public class ExceptionsDemo {
    public static void main(String[] args) {
        try {
            RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");
            String line = file.readLine();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere Fehlerbehandlung.");
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere spezielle Fehlerbehandlung fuer IOException.");
        }
    }
}
```

Problems

```
<terminated> ExceptionsDemo [Java Application] /System/Library/java/javaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
java.io.FileNotFoundException: DieseDateiGibtEsNicht.txt (No such file or directory)
at java.io.RandomAccessFile.open(Native Method)
at java.io.RandomAccessFile.<init>(RandomAccessFile.java:216)
at java.io.RandomAccessFile.<init>(RandomAccessFile.java:101)
at ExceptionsDemo.main(ExceptionsDemo.java:8)
```

Hier ist unsere Fehlerbehandlung.

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:34

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;

public class ExceptionsDemo {
    public static void main(String[] args) {
        try {
            RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");
            String line = file.readLine();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere Fehlerbehandlung.");
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere spezielle Fehlerbehandlung fuer IOException.");
        }
    }
}
```

Problems

```
<terminated> ExceptionsDemo [Java Application] /System/Library/java/javaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
java.io.FileNotFoundException: DieseDateiGibtEsNicht.txt (No such file or directory)
at java.io.RandomAccessFile.open(Native Method)
at java.io.RandomAccessFile.<init>(RandomAccessFile.java:216)
at java.io.RandomAccessFile.<init>(RandomAccessFile.java:101)
at ExceptionsDemo.main(ExceptionsDemo.java:8)
```

Hier ist unsere Fehlerbehandlung.

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:35

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;

public class ExceptionsDemo {

    public static void main(String[] args) {
        try {
            RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");
            String line = file.readLine();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere Fehlerbehandlung.");
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere spezielle Fehlerbehandlung fuer IOException.");
        }

        int[] numbers = { 1, 2, 3, 4, 5 };
        for (int i = 0; i < numbers.length; i++) {
            System.out.println(numbers[i]);
        }
    }
}
```

Problems

```
<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
java.io.FileNotFoundException: DieseDateiGibtEsNicht.txt (No such file or directory)
at java.io.RandomAccessFile.open(Native Method)
at java.io.RandomAccessFile.<init>(RandomAccessFile.java:216)
at java.io.RandomAccessFile.<init>(RandomAccessFile.java:181)
at ExceptionsDemo.main(ExceptionsDemo.java:8)

Hier ist unsere Fehlerbehandlung.
```

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:37

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;

public class ExceptionsDemo {

    public static void main(String[] args) {
        try {
            RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");
            String line = file.readLine();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere Fehlerbehandlung.");
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere spezielle Fehlerbehandlung fuer IOException.");
        }

        int[] numbers = { 1, 2, 3, 4, 5 };
        for (int i = 0; i < numbers.length; i++) {
            System.out.println(numbers[i]);
        }
    }
}
```

Problems

```
<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
1
2
3
4
5
```

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:37

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;

public class ExceptionsDemo {

    public static void main(String[] args) {
        try {
            RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");
            String line = file.readLine();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere Fehlerbehandlung.");
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere spezielle Fehlerbehandlung fuer IOException.");
        }

        int[] numbers = { 1, 2, 3, 4, 5 };
        for (int i = 0; i < numbers.length; i++) {
            System.out.println(numbers[i]);
        }
    }
}
```

Problems

```
<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
1
2
3
4
5
```

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:37

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;

public class ExceptionsDemo {

    public static void main(String[] args) {
        try {
            RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");
            String line = file.readLine();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere Fehlerbehandlung.");
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Hier ist unsere spezielle Fehlerbehandlung fuer IOException.");
        }

        int[] numbers = { 1, 2, 3, 4, 5 };
        for (int i = 0; i < numbers.length; i++) {
            System.out.println(numbers[i]);
        }
    }
}
```

Problems

```
<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
1
2
3
4
5
```

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (0:12) Fr. 6. Jul 9:38

Java - Exceptions/src/Bicycle.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
public class Bicycle {  
    private int gear;  
  
    public Bicycle(int initialGear) {  
    }  
}
```

Problems Javadoc Declaration Console

```
<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)  
1  
2  
3  
4  
5
```

Writable Smart Insert 6 : 9

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (0:12) Fr. 6. Jul 9:38

Java - Exceptions/src/Bicycle.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
public class Bicycle {  
    private int gear;  
  
    public Bicycle(int initialGear) {  
        if ()  
    }  
}
```

Problems Javadoc Declaration Console

```
<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)  
1  
2  
3  
4  
5
```

Writable Smart Insert 6 : 13

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 6. Jul 9:39

Java - Exceptions/src/Bicycle.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
public class Bicycle {  
    private int gear;  
  
    public Bicycle(int initialGear) {  
        if (initialGear <= 0) {  
            throw new Exception("D");  
        } else {  
            gear = initialGear;  
        }  
    }  
}
```

Problems Javadoc Declaration Console

```
<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)  
1  
2  
3  
4  
5
```

Unhandled exception type Exception

Writable Smart Insert 7 : 34

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 6. Jul 9:39

Java - Exceptions/src/Bicycle.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
public class Bicycle {  
    private int gear;  
  
    public Bicycle(int initialGear) {  
        if (initialGear <= 0) {  
            Unhandled exception type Exception on ("initialGear >= 0");  
        } else {  
            gear = initialGear;  
        }  
    }  
}
```

Problems Javadoc Declaration Console

```
<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)  
1  
2  
3  
4  
5
```

Unhandled exception type Exception

Writable Smart Insert 7 : 50

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:39 Q

Java - Exceptions/src/Bicycle.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
public class Bicycle {  
    private int gear;  
  
    public Bicycle(int initialGear) {  
        if (initialGear <= 0) {  
            throw new Exception("InitialGear >= 0!");  
        } else {  
            gear = initialGear;  
        }  
    }  
}
```

ExceptionsDemo.java *Bicycle.java

Problems @ Javadoc Declaration Console

```
<terminated> ExceptionsDemo [Java Application] /System/Library/java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 201  
1  
2  
3  
4  
5
```

Writable Smart Insert 5 : 37

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:40 Q

Java - Exceptions/src/Bicycle.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
public class Bicycle {  
    private int gear;  
  
    public Bicycle(int initialGear) throws Exception {  
        if (initialGear <= 0) {  
            throw new Exception("InitialGear >= 0!");  
        } else {  
            gear = initialGear;  
        }  
    }  
}
```

ExceptionsDemo.java *Bicycle.java

Problems @ Javadoc Declaration Console

```
<terminated> ExceptionsDemo [Java Application] /System/Library/java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 201  
1  
2  
3  
4  
5
```

Writable Smart Insert 5 : 54

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:41 Q

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.io.RandomAccessFile;  
  
public class ExceptionsDemo {  
  
    public static void main(String[] args) {  
        try {  
            RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");  
            String line = file.readLine();  
        } catch (FileNotFoundException e) {  
            e.printStackTrace();  
            System.out.println("Hier ist unsere Fehlerbehandlung.");  
        } catch (IOException e) {  
            e.printStackTrace();  
            System.out.println("Hier ist unsere spezielle Fehlerbehandlung fuer IOException.");  
        }  
  
        int[] numbers = { 1, 2, 3, 4, 5 };  
        for (int i = 0; i < numbers.length; i++) {  
            System.out.println(numbers[i]);  
        }  
    }  
}
```

ExceptionsDemo.java *Bicycle.java *InvalidGearException.java

Problems @ Javadoc Declaration Console

```
<terminated> ExceptionsDemo [Java Application] /System/Library/java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 201  
1  
2  
3  
4  
5
```

Writable Smart Insert 22 : 10

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 6. Jul 9:42 Q

Java - Exceptions/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.io.RandomAccessFile;  
  
public class ExceptionsDemo {  
  
    public static void main(String[] args) throws InvalidGearException {  
        try {  
            RandomAccessFile file = new RandomAccessFile("DieseDateiGibtEsNicht.txt", "r");  
            String line = file.readLine();  
        } catch (FileNotFoundException e) {  
            e.printStackTrace();  
            System.out.println("Hier ist unsere Fehlerbehandlung.");  
        } catch (IOException e) {  
            e.printStackTrace();  
            System.out.println("Hier ist unsere spezielle Fehlerbehandlung fuer IOException.");  
        }  
  
        int[] numbers = { 1, 2, 3, 4, 5 };  
        for (int i = 0; i < numbers.length; i++) {  
            System.out.println(numbers[i]);  
        }  
  
        Bicycle bike = new Bicycle(-3);  
    }  
}
```

ExceptionsDemo.java *Bicycle.java *InvalidGearException.java

Problems @ Javadoc Declaration Console

```
<terminated> ExceptionsDemo [Java Application] /System/Library/java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 201  
Exception in thread "main" InvalidGearException  
at Bicycle.<init>(Bicycle.java:7)  
at ExceptionsDemo.main(ExceptionsDemo.java:24)
```

Writable Smart Insert 7 : 72



6 Coding & Naming Conventions

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>



6 Coding & Naming Conventions

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>

6 Coding & Naming Conventions

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

```
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!");
        }
    }
}
```

6 Coding & Naming Conventions

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

```
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!");
}
```

6 Coding & Naming Conventions

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

```
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!



6 Coding & Naming Conventions

- 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:

```
int rowIndex  
int columnIndex  
boolean isValid  
static boolean copyFile(URL from, URL to)
```

- Names of constants
 - Uppercase with words separated by underscores

```
static final int MIN_ALLOWED_VALUE = 42;
```

6 Coding & Naming Conventions

- 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:

```
class Bicycle  
class ImageSprite  
interface RasterDelegate  
interface Comparable
```

6 Coding & Naming Conventions

- Use English for everything, e.g. names, comments etc.
- Adhere STRICTLY to common coding styles
 - Do not use exotic "own" coding styles!
- Use default style of IDE
 - (Integrated Development Environment, e.g. Eclipse)
- Use comments REGULARLY and THOROUGHLY
- Use Javadoc (not only for larger projects)



7 Using the Java Class Library

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/>

- So far:
 - Classes, objects, methods, fields, variables, control flow statements etc.
 - These are only *basic tools*
- Advanced problems require a multitude of specialized types (classes etc.)
- Write them all by yourself?
 - No! Java's strength: Types for almost EVERYTHING exist
- "Class Library"
 - Documentation at <http://java.sun.com/javase/6/docs/api/>
- Similar class libraries exist for .NET, C++, and other languages/platforms

7 Using the Java Class Library

Packages	Description
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent

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.netURLConnection;
import java.util.Collection;
import some.other.package.*; // * means all types
```

7 Using the Java Class Library

The screenshot shows a web browser displaying the Java API documentation. The title bar reads "Overview (Java Platform SE 6)" and the address bar shows "http://docs.oracle.com/javase/6/docs/api/". The main content area is titled "Java™ Platform, Standard Edition 6 API Specification". It includes a sidebar with "All Classes" and "Packages" sections, and a central table listing various Java packages with their descriptions.

Packages	
java.awt	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt.color	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.datatransfer	Provides classes for color spaces.
java.awt.dnd	Provides interfaces and classes for transferring data between and within applications.
java.awt.event	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.font	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.geom	Provides classes and interface relating to fonts.
java.awt.image	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.image.renderable	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image	Provides classes for creating and modifying images.
java.awt.layout	Provides classes and interfaces for producing rendering-independent

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.netURLConnection;  
import java.util.Collection;  
import some.other.package.*; // * means all types
```

7 Using the Java Class Library

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:**

```
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);
```

7 Using the Java Class Library

Wrapper-Classes

- Parameters restricted to **reference types** (classes, interfaces)
- For each **primitive type**, a corresponding **wrapper class** exists
- Examples:

```
java.lang.Short  
java.lang.Integer  
java.lang.Long  
java.lang.Float  
java.lang.Double
```

etc.

- Primitive types are automatically **boxed** and **unboxed** when necessary

```
Integer i = 723;  
int j = i;
```



8 Solving a Problem

Example Task

8 Solving a Problem

Deepening readings (optional):

[The Internet](#)

Main reference:

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

- **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 ☺

8 Solving a Problem

Example Task

java load picture from url and display - Google-Suche
http://www.google.de/search?client=safari&rls=en&q=java+load+picture+from+url+and+display...C

Suche: ungefähr 32.800.000 Ergebnisse (0.31 Sekunden)

Alles Bilder Maps Videos News Shopping Mehr

how to load a image from web in java - Stack Overflow
stackoverflow.com/.../how-to-load-a-image-from... - Diese Seite übersetzen
4 Antworten · 29. Mai 2009

I need to load an image from a web in a simple Java stand alone application. ... Linked.
Load image from URL and limit the size of the file ...

Reading/Loading an Image (The Java™ Tutorials > 2D Graphics)
docs.oracle.com/javase/.../images/loadimage.htm... - Diese Seite übersetzen
LoadingImageApp.java shows how to display this image ... The getCodeBase method used in this example returns the URL of the directory containing this applet.

Learning Java - Chapter 6 : Java
www.particlekit.se/.../JavaCourse.../Java... - Diese Seite übersetzen
8 Nov 2008 - As of Java 1.4 you can load and draw image files encoded as JPEG, GIF and PNG ... use one of the overloaded getimage() methods for loading from a URL ... that provides various resources and tools for the display system.

Getting an Image from a URL Example Depot
www.exampledepot.com/.../java.net.URL... - Diese Seite übersetzen
try { // Create a URL for the image's location URL url = new URL("http://hostname:80/Image.gif"); // Get the image java.awt.Image image = java.awt.Toolkit.getDefaultToolkit().getDesktopImage(); ... It's a good article.Thank you for show to us.

Reading an Image from a File, InputStream, or URL Example Depot
www.exampledepot.com/.../java.awt.Image... - Diese Seite übersetzen
Java Code Examples from The Java Developers' Almanac ... The example also demonstrates how to display the image on the screen ... I'm wondering how can I load an image from any directory of the disc, not from the root of the project.

ithx but how do i display an image in java | DaniWeb
www.daniweb.com/.../Java... - Diese Seite übersetzen
Software Development > Java > ithx but how do i display an image in java ... load

8 Solving a Problem

Example Task

Reading/Loading an Image (The Java™ Tutorials > 2D Graphics > Working with Images)
http://docs.oracle.com/javase/tutorial/2d/images/loadimage.html

The Java™ Tutorials

Working with Images
Reading/Loading an Image
Drawing an Image
Creating and Drawing to an Image
Writing/Saving an Image

When you think of digital images, you probably think of sampled image formats such as the JPEG image format used in digital photography, or GIF images commonly used on web pages. All programs that can use these images must first convert them from their external format into an internal format.

Java 2D™ supports loading these external image formats into its `BufferedImage` format using its Image API which is in the `javax.image` package. Image I/O has built-in support for GIF, PNG, JPEG, BMP, and WBMP. Image I/O is also extensible so that developers or administrators can "plug-in" support for additional formats. For example, plug-ins for TIFF and JPEG 2000 are separately available.

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

```
BufferedImage img = null;
try {
    img = ImageIO.read(new File("strawberry.jpg"));
} catch (IOException e) {
}
```

Image I/O recognises the contents of the file as a JPEG format image, and decodes it into a `BufferedImage` which can be directly used by Java 2D.

>LoadingImageApp.java shows how to display this image.

If the code is running in an applet, then it's just as easy to obtain the image from the applet codebase :

```
try {
    URL url = new URL(getCodeBase(),
                      "strawberry.jpg");
    img = ImageIO.read(url);
} catch (IOException e) {
}
```

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

The following example shows how to use the `getCodeBase` method to load the strawberry.jpg file.

Introduction to Java Basics.pptx

8 Solving a Problem

Example Task

Reading/Loading an Image (The Java™ Tutorials > 2D Graphics > Working with Images)

The Java™ Tutorials

Working with Images
Reading/Loading an Image
Drawing an Image
Creating and Drawing to an Image
Writing/Saving an Image

← Previous • Trail • Next → Home Page > 2D Graphics > Working with Images

Reading/Loading an Image

When you think of digital images, you probably think of sampled image formats such as the JPEG image format used in digital photography, or GIF images commonly used on web pages. All programs that can use these images must first convert them from that external format into an internal format.

Java 2D™ supports loading these external image formats into its `BufferedImage` format using its Image I/O API which is in the `javax.imageio` package. Image I/O has built-in support for GIF, PNG, JPEG, BMP, and WBMP. Image I/O is also extensible so that developers or administrators can "plug-in" support for additional formats. For example, plug-ins for TIFF and JPEG 2000 are separately available.

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

```
BufferedImage img = null;
try {
    img = ImageIO.read(new File("strawberry.jpg"));
} catch (IOException e) {
```

Image I/O recognises the contents of the file as a JPEG format image, and decodes it into a `BufferedImage` which can be directly used by Java 2D.

`LoadImageApp.java` shows how to display this image.

If the code is running in an applet, then it's just as easy to obtain the image from the applet codebase :

```
try {
    URL url = new URL(getCodeBase(),
        "strawberry.jpg");
    img = ImageIO.read(url);
} catch (IOException e) {
```

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

The following example shows how to use the `getcodebase` method to load the strawberry.jpg file.

8 Solving a Problem

Example Task

Reading/Loading an Image (The Java™ Tutorials > 2D Graphics > Working with Images)

The Java™ Tutorials

Working with Images
Reading/Loading an Image
Drawing an Image
Creating and Drawing to an Image
Writing/Saving an Image

← Previous • Trail • Next → Home Page > 2D Graphics > Working with Images

Reading/Loading an Image

When you think of digital images, you probably think of sampled image formats such as the JPEG image format used in digital photography, or GIF images commonly used on web pages. All programs that can use these images must first convert them from that external format into an internal format.

Java 2D™ supports loading these external image formats into its `BufferedImage` format using its Image I/O API which is in the `javax.imageio` package. Image I/O has built-in support for GIF, PNG, JPEG, BMP, and WBMP. Image I/O is also extensible so that developers or administrators can "plug-in" support for additional formats. For example, plug-ins for TIFF and JPEG 2000 are separately available.

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

```
BufferedImage img = null;
try {
    img = ImageIO.read(new File("strawberry.jpg"));
} catch (IOException e) {
```

Image I/O recognises the contents of the file as a JPEG format image, and decodes it into a `BufferedImage` which can be directly used by Java 2D.

`LoadImageApp.java` shows how to display this image.

If the code is running in an applet, then it's just as easy to obtain the image from the applet codebase :

```
try {
    URL url = new URL(getCodeBase(),
        "strawberry.jpg");
    img = ImageIO.read(url);
} catch (IOException e) {
```

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

The following example shows how to use the `getcodebase` method to load the strawberry.jpg file.

8 Solving a Problem

Example Task

Reading/Loading an Image (The Java™ Tutorials > 2D Graphics > Working with Images)

The Java™ Tutorials

Working with Images
Reading/Loading an Image
Drawing an Image
Creating and Drawing to an Image
Writing/Saving an Image

← Previous • Trail • Next → Home Page > 2D Graphics > Working with Images

Reading/Loading an Image

When you think of digital images, you probably think of sampled image formats such as the JPEG image format used in digital photography, or GIF images commonly used on web pages. All programs that can use these images must first convert them from that external format into an internal format.

Java 2D™ supports loading these external image formats into its `BufferedImage` format using its Image I/O API which is in the `javax.imageio` package. Image I/O has built-in support for GIF, PNG, JPEG, BMP, and WBMP. Image I/O is also extensible so that developers or administrators can "plug-in" support for additional formats. For example, plug-ins for TIFF and JPEG 2000 are separately available.

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

```
BufferedImage img = null;
try {
    img = ImageIO.read(new File("strawberry.jpg"));
} catch (IOException e) {
```

Image I/O recognises the contents of the file as a JPEG format image, and decodes it into a `BufferedImage` which can be directly used by Java 2D.

`LoadImageApp.java` shows how to display this image.

If the code is running in an applet, then it's just as easy to obtain the image from the applet codebase :

```
try {
    URL url = new URL(getCodeBase(),
        "strawberry.jpg");
    img = ImageIO.read(url);
} catch (IOException e) {
```

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

The following example shows how to use the `getcodebase` method to load the strawberry.jpg file.

8 Solving a Problem

Example Task

URL (Java Platform SE 6)

http://docs.oracle.com/javase/6/docs/api/java/net/URL.html

Overview Package Class Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

java.net

Class URL

`java.lang.Object`
↳ `java.net.URL`

All Implemented Interfaces:
`Serializable`

public final class URL
extends Object
implements Serializable

Class `URL` represents a Uniform Resource Locator, a pointer to a "resource" on the World Wide Web. A resource can be something as simple as a file or a directory, or it can be a reference to a more complicated object, such as a query to a database or to a search engine. More information on the types of URLs and their formats can be found at:

<http://www.socs.uts.edu.au/MosaicDocs-old/primer.html>

In general, a URL can be broken into several parts. The previous example of a URL indicates that the protocol to use is `http` (HyperText Transfer Protocol) and that the information resides on a host machine named `socscs.uts.edu.au`. The information on that host machine is named `/MosaicDocs-old/url-primer.html`. The exact meaning of this name on the host machine is both protocol dependent and host dependent. The information normally resides in a file, but it could be generated on the fly. This component of the URL is called the `path` component.

A URL can optionally specify a "port", which is the port number to which the TCP connection is made on the remote host machine. If the port is not specified, the default port for the protocol is used instead. For example, the default port for `http` is `80`. An alternative port could be specified as:

<http://www.socs.uts.edu.au:80/MosaicDocs-old/url-primer.html>

The syntax of `url`, defined by [RFC 3986: Uniform Resource Identifiers \(URI\): Generic Syntax](http://www.ietf.org/rfc/rfc3986.txt), amended by [RFC 2732: Format for Literal IPv6 Addresses in URLs](http://www.ietf.org/rfc/rfc2732.txt). The Literal IPv6 address format also supports scope_ids. The syntax and usage of scope_ids is described [here](#).

A URL may have appended to it a "fragment", also known as a "ref" or a "reference". The fragment is indicated by the sharp sign character "#" followed by more characters. For example,

8 Solving a Problem

Example Task

URL (Java Platform SE 6)
http://docs.oracle.com/javase/6/docs/api/java/net/URL.html
C Qr java 6 api url

http://foo.com/hello%20world and http://foo.com/hello%20world

would be considered not equal to each other.

Note, the `URL` class does perform escaping of its component fields in certain circumstances. The recommended way to manage the encoding and decoding of URLs is to use `URL`, and to convert between these two classes using `toURI()` and `URI.toURL()`.

The `URLEncoder` and `URLDecoder` classes can also be used, but only for HTML form encoding, which is not the same as the encoding scheme defined in RFC2396.

Since:
JDK 1.0

See Also:
[Serialized Form](#)

Constructor Summary

<code>URL(String spec)</code>	Creates a <code>URL</code> object from the <code>String</code> representation.
<code>URL(String protocol, String host, int port, String file)</code>	Creates a <code>URL</code> object from the specified protocol, host, port number, and file.
<code>URL(String protocol, String host, int port, String file, URLStreamHandler handler)</code>	Creates a <code>URL</code> object from the specified protocol, host, port number, file, and handler.
<code>URL(String protocol, String host, String file)</code>	Creates a URL from the specified protocol name, host name, and file name.
<code>URL(URL context, String spec)</code>	Creates a URL by parsing the given spec within a specified context.
<code>URL(URL context, String spec, URLStreamHandler handler)</code>	Creates a URL by parsing the given spec with the specified handler within a specified context.

Method Summary

<code>boolean equals(Object obj)</code>	Compares this URL for equality with another object.
<code>String getAuthority()</code>	Gets the authority part of this <code>URL</code> .
<code>Object getContent()</code>	Gets the content of this <code>URL</code> .

8 Solving a Problem

Example Task

URL (Java Platform SE 6) [< Back](http://docs.oracle.com/javase/6/docs/api/java/net/URL.html#URL(java.lang.String)) [java 6 api url](#)

URL

```
public URL(String spec)
throws MalformedURLException
```

Creates a URL object from the String representation.

This constructor is equivalent to a call to the two-argument constructor with a null first argument.

Parameters:
spec - the string to parse as a URL.

Throws:
`MalformedURLException` - If the string specifies an unknown protocol.

See Also:
`URL(java.net.URL, java.lang.String)`

URL

```
public URL(URL context,
          String spec)
throws MalformedURLException
```

Creates a URL by parsing the given spec within a specified context. The new URL is created from the given context URL and the spec argument as described in RFC2396 "Uniform Resource Identifiers : Generic " Syntax":

```
<scheme>://<authority><path><?query#<fragment>
```

The reference is parsed into the scheme, authority, path, query and fragment parts. If the path component is empty and the scheme, authority, and query components are undefined, then the new URL is a reference to the current document. Otherwise, the fragment and query parts present in the spec are used in the new URL.

If the scheme component is defined in the given spec and does not match the scheme of the context, then the new URL is created as an absolute URL based on the spec alone. Otherwise the scheme component is inherited from the context URL.

If the authority component is present in the spec then the spec is treated as absolute and the spec authority and path will replace the context authority and path. If the authority component is absent in the spec then the authority of the new URL will be inherited from the context.

If the spec's path component begins with a slash character "/" then the path is treated as absolute and the spec path replaces the context path.

Otherwise, the path is treated as a relative path and is appended to the context path, as described in RFC2396. Also, in this case, the path is canonicalized through the removal of directory changes made by occurrences of "..".

Introduction to Java Basics.pptx

In der Präsentation suchen

Start Designs Tabellen Diagramme SmartArt Übergänge Animationen Bildschirmpäsentation Überprüfen

Folien Schriftart Absatz Einfügen Anordnen Schnellformatvorlagen

Format

Neue Folie

144

145

146

147

148

149

150

151

8 Solving a Problem

Example Task

URL (Java Platform SE 6)

[http://docs.oracle.com/javase/6/docs/api/java/net/URL.html#URL\(java.lang.String\)](http://docs.oracle.com/javase/6/docs/api/java/net/URL.html#URL(java.lang.String))

URL

```
public URL(String spec)
throws MalformedURLException
Creates a URL object from the given representation.
```

This constructor is equivalent to a call to the two-argument constructor with a null first argument.

Parameters:

- spec - The string to parse as a URL.

Themes:

- MalformedURLException - If the string specifies an unknown protocol.

See Also:

- [URL class](#)
- [URLConnection class](#)

URL

```
public URL(URL context,
           String spec,
           throws MalformedURLException
Creates a URL by using the given spec within a specified context. The new URL is created from the given context URL, and the spec argument as described in the URL class.
```

Parameters:

- context - A URL by which the given spec within a specified context. The new URL is created from the given context URL, and the spec argument as described in the URL class.
- spec - A string to parse as a URL.

The reference is parsed into the scheme, authority, path, query and fragment parts. If the path component is empty and the scheme, authority, and query components are undefined, then the new URL is a reference to the context document. Otherwise, the fragment and query parts present in the spec are used in the new URL.

If the scheme component is defined in the given spec and does not match the scheme of the context, then the new URL is created as an absolute URL based on the spec alone. Otherwise the scheme component is taken from the context URL.

If the authority component is defined in the given spec, then the spec's authority and path will replace the context authority and path. If the authority component is undefined in the spec then the authority of the new URL will be inherited from the context.

If the spec's path component begins with a slash character "/" then the path is treated as absolute and the spec path replaces the context path.

Otherwise, the path is treated as a relative path and is appended to the context path, as described in RFC2396. Also, in this case, the path is canonicalized through the removal of directory changes made by occurrences of ".." and ":".

Notizen durch Klicken hinzufügen

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Eclipse - Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace. It also displays the date and time: Fr. 6. Jul 10:15.
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project navigation.
- Package Explorer View:** Shows the project structure. The **ImageDemo** project is expanded, showing its **src** folder which contains a package named **(default package)** with a file named **ImageDemo.java**.
- Editor View:** Displays the Java code for **ImageDemo.java**. The code defines a public static void main method that creates a URL object pointing to a specific image URL.
- Console View:** Shows the output of the application's execution. It indicates the application terminated with an exception. The stack trace shows an **InvalidGearException** occurring at **Bicycle.<init>(Bicycle.java:24)**, which is part of the **ExceptionsDemo.main** method.

Eclipse File Edit Run Source Navigate Project Refactor Window Help (Charged) Fr. 6. Jul 10:15

Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

*ImageDemo.java

```
public class ImageDemo {  
    public static void main(String[] args) {  
        URL url = new URL("http://vmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");  
    }  
}
```

Problems Javadoc Declaration Console

<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
Exception in thread "main" InvalidGearException
at Bicycle.<init>(Bicycle.java:7)
at ExceptionsDemo.main(ExceptionsDemo.java:24)

Writable Smart Insert 4 : 9

Eclipse File Edit Run Source Navigate Project Refactor Window Help (Charged) Fr. 6. Jul 10:16

Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

*ImageDemo.java

```
public class ImageDemo {  
    public static void main(String[] args) {  
        URL url = new URL("http://vmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");  
    }  
}
```

Organize Imports

Choose type to import:

Page 1 of 1

java.net.URL
javax.print.DocFlavor.URL

Problems Javadoc Declaration Console

<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
Exception in thread "main" InvalidGearException
at Bicycle.<init>(Bicycle.java:7)
at ExceptionsDemo.main(ExceptionsDemo.java:24)

URL cannot be resolved to a type Writable Smart Insert 4 : 12

Eclipse File Edit Run Source Navigate Project Refactor Window Help (Charged) Fr. 6. Jul 10:16

Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

*ImageDemo.java

```
import java.net.URL;  
  
public class ImageDemo {  
    public static void main(String[] args) {  
        URL url = new URL("http://vmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");  
    }  
}
```

Problems Javadoc Declaration Console

<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
Exception in thread "main" InvalidGearException
at Bicycle.<init>(Bicycle.java:7)
at ExceptionsDemo.main(ExceptionsDemo.java:24)

Writable Smart Insert 6 : 9

Eclipse File Edit Run Source Navigate Project Refactor Window Help (Charged) Fr. 6. Jul 10:16

Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

*ImageDemo.java

```
import java.net.URL;  
  
public class ImageDemo {  
    public static void main(String[] args) {  
        try {  
            URL url = new URL("http://vmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");  
        }  
    }  
}
```

Problems Javadoc Declaration Console

<terminated> ExceptionsDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2011)
Exception in thread "main" InvalidGearException
at Bicycle.<init>(Bicycle.java:7)
at ExceptionsDemo.main(ExceptionsDemo.java:24)

Writable Smart Insert 6 : 13

Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 6. Jul 10:18 Q

Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.net.URL;
public class ImageDemo {
    public static void main(String[] args) {
        try {
            URL url = new URL("http://vmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Problems @ Javadoc Declaration Console

<terminated> ImageDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2012 10:18:21)

Writable Smart Insert 9 : 35

Introduction to Java Basics.pptx

Start Designs Tabellen Diagramme SmartArt Übergänge Animationen Bildschirmpresentation Überprüfen Einfügen Anordnen Schnellformatvorlagen

Folien

144 (highlighted)

145

146

147

148

149

150

151

8 Solving a Problem

Example Task

The Java™ Tutorials
Reading/Loading an Image (The Java™ Tutorials > 2D Graphics > Working with Images)

Reading/Loading an Image

When you think of digital images, you probably think of compressed image formats such as the JPEG image format used in digital photography, or GIF images commonly used on web pages. All programs that can see these images must first convert them from the external formats into an internal format.

Java 2D® supports loading these external image formats into its internal format using the ImageIO API, which is part of the standard Java API. This API has support for reading JPEG, PNG, JPEG2000, and BMP. ImageIO is also extensible so that developers or administrators can "plug-in" support for additional formats. For example, plug-ins for TIFF and JPEG2000 are separately available.

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

```
Image img = null;
try {
    img = ImageIO.read(new File("strawberry.jpg"));
} catch (IOException e) {
    e.printStackTrace();
}
```

ImageIO recognizes the extension of the file as a JPEG format image, and decodes it into a BufferedImage which can be directly used by Java 2D.

LoadImageApp.java shows how to display this image.

If the code is running in an applet, then it's just as easy to obtain the image from the applet codebase:

```
try {
    URL url = new URL(getCodeBase());
    img = ImageIO.read(url);
} catch (IOException e) {
    e.printStackTrace();
}
```

The getCodeBase() method used in this example returns the URL of the directory containing this applet.

The following example shows how to use the getCodeBase() method to load the strawberry.jpg file.

Notizen durch Klicken hinzufügen

144 von 158 89%

Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 6. Jul 10:19 Q

Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.awt.image.BufferedImage;
import java.net.URL;
import javax.imageio.ImageIO;

public class ImageDemo {
    public static void main(String[] args) {
        try {
            URL url = new URL("http://vmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");
            BufferedImage image = ImageIO.read(url);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Problems @ Javadoc Declaration Console

<terminated> ImageDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2012 10:18:21)

Writable Smart Insert 13 : 53

Eclipse File Edit Run Source Navigate Project Refactor Window Help (Charged) Fr. 6. Jul 10:19 Q

Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

```
import java.awt.image.BufferedImage;
import java.net.URL;
import javax.imageio.ImageIO;

public class ImageDemo {
    public static void main(String[] args) {
        try {
            URL url = new URL("http://vmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");
            URL url = new URL("file:///tmp/ImageDemo/morningDew.jpg");
            BufferedImage image = ImageIO.read(url);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Problems @ Javadoc Declaration Console

<terminated> ImageDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2012 10:18:21)

Writable Smart Insert 14 : 13

The screenshot shows the Eclipse IDE interface. On the left is the Package Explorer with a tree view of Java projects and source files. In the center is the Java editor window titled "ImageDemo.java" containing the following code:

```

import java.awt.image.BufferedImage;
import java.net.URL;

import javax.imageio.ImageIO;

public class ImageDemo {
    public static void main(String[] args) {
        try {
            URL url = new URL("http://vmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");
            BufferedImage image = ImageIO.read(url);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Below the editor are tabs for "Problems", "@ Javadoc", "Declaration", and "Console". The "Console" tab shows the message "<terminated> ImageDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2012 10:14:13)".

The screenshot shows a Microsoft PowerPoint slide titled "8 Solving a Problem" under the heading "Example Task". The slide contains a code snippet from "URL.java" and a callout bubble pointing to it.

```

public URL(URL url, String spec)
throws MalformedURLException {
    Create a URL object from the given spec representation.
    This constructor is equivalent to a call to the two-argument constructor with a null first argument.
    Parameters:
    url - the URL to parse
    spec - the string to parse as a URL.
    Throws:
    MalformedURLException - if the string specifies an unknown protocol.
    See Also:
    URL(URL url, String spec)

```

The callout bubble contains the following text:

Note: If the reference is parsed into the scheme, authority, path, query and fragment parts. If the path component is empty and the scheme, authority, and query components are undefined, then the new URL is a reference to the current document. Otherwise, the fragment and query parts present in the spec are used in the new URL.

If the scheme component is defined in the given spec and does not match the scheme of the context, then the new URL is created as an absolute URL based on the spec alone. Otherwise the scheme component is inherited from the context URL.

If the authority component is present in the spec then the spec is treated as absolute and the spec authority and path will replace the context authority and path. If the authority component is absent in the spec then the authority of the new URL will be inherited from the context.

If the spec's path component begins with a slash character "/" then the path is treated as absolute and the spec path replaces the context path.

Otherwise, the path is treated as a relative path and is appended to the context path, as described in RFC2396. Also, in this case, the path is canonized through the removal of directory changes made by occurrences of .. and .

8 Solving a Problem

Example Task

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

The screenshot shows a Java applet window titled "Reading/Loading an Image (The Java™ Tutorials > 2D Graphics > Working with Images)". The page content discusses loading images from URLs and displays a sample code snippet. A red arrow points to the code block.

Note:

If you don't see the applet running, you need to install [release 6 \(or later\)](#) of the Java SE Development Kit (JDK).

LoadImageApp.java contains the complete code for this example and this applet requires the [strawberry.jpg](#) image file.

In addition to reading from files or URLs, Image I/O can read from other sources, such as an `InputStream`. `ImageIO.read()` is the most straightforward convenience API for most applications, but the `javax.imageio.ImageIO` class provides many more static methods for more advanced usages of the Image I/O API. The collection of methods on this class represent just a subset of the rich set of APIs for discovering information about the images and for controlling the image decoding (reading) process.

We will explore some of the other capabilities of Image I/O later in the [Writing/Saving an Image](#) section.

8 Solving a Problem

Example Task

The screenshot shows a Java application window titled "LoadImageApp.java" running in a browser. The code is identical to the one shown in the previous screenshot. A large blue thought bubble with a sad face is overlaid on the right side of the window.

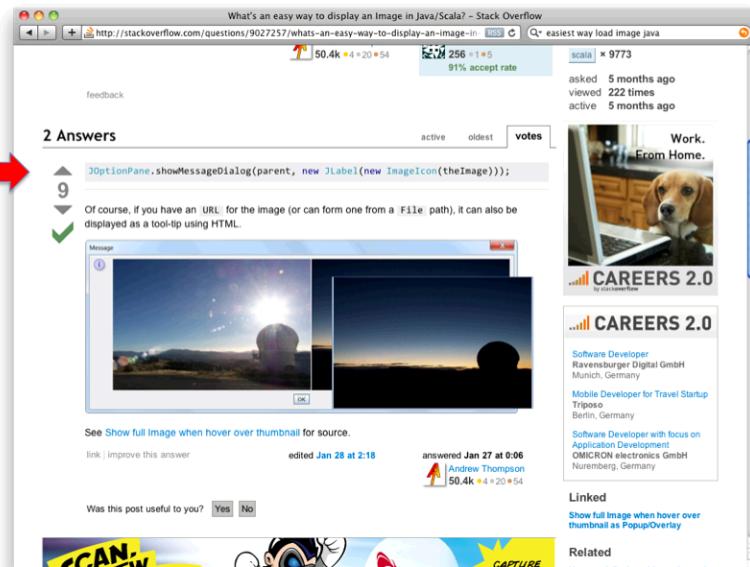
```

public class LoadImageApp extends Component {
    BufferedImage img;
    public void paint(Graphics g) {
        g.drawImage(img, 0, 0, null);
    }
    public LoadImageApp() {
        try {
            img = ImageIO.read(new File("strawberry.jpg"));
        } catch (IOException e) {
        }
    }
    public Dimension getPreferredSize() {
        if (img == null) {
            return new Dimension(100,100);
        } else {
            return new Dimension(img.getWidth(), img.getHeight());
        }
    }
    public static void main(String[] args) {
        JFrame f = new JFrame("Load Image Sample");
        f.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        f.add(new LoadImageApp());
        f.pack();
        f.setVisible(true);
    }
}

```

8 Solving a Problem

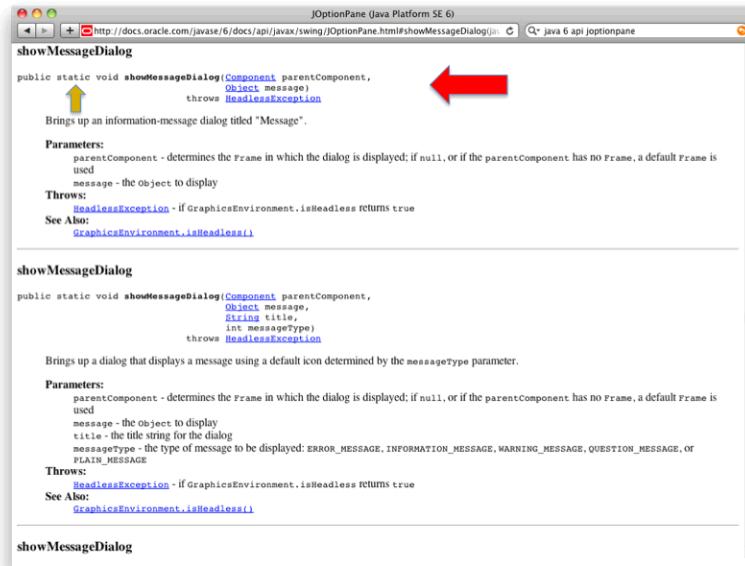
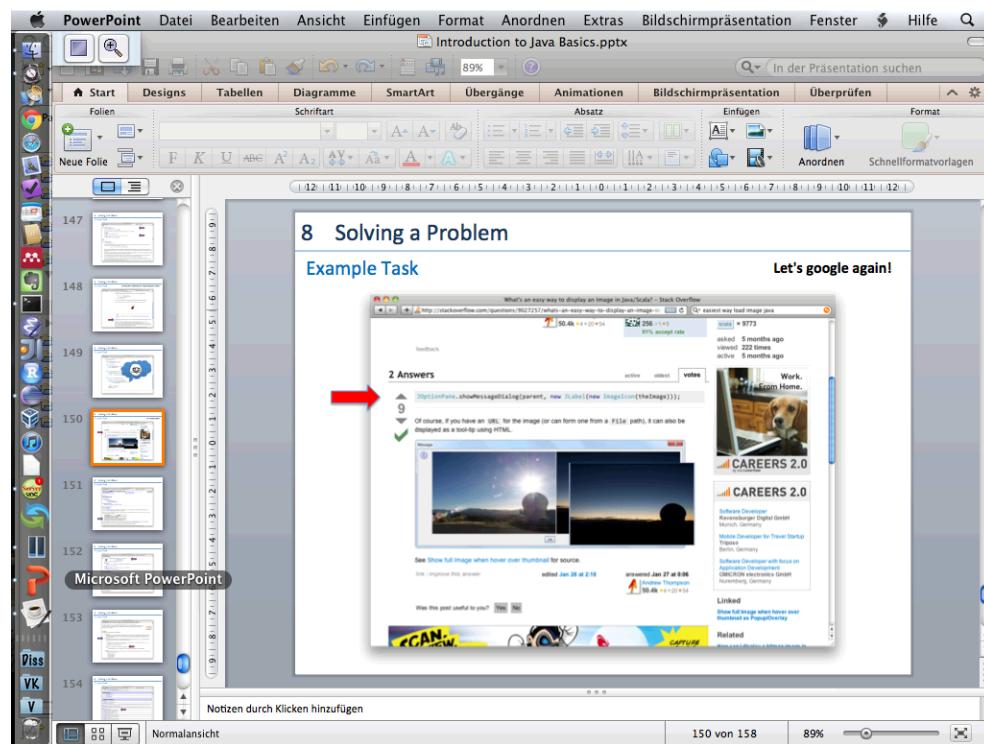
Example Task



Let's google again!

8 Solving a Problem

Example Task



Eclipse File Edit Run Source Navigate Project Refactor Window Help (Charged) Fr. 6. Jul 10:26

Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

- BankAccount
- BeesAndFlowers
- ControlFlowDemo
- Exceptions
- Exercise2
- HistogramUni
- ImageDemo
 - src
 - (default package)
 - ImageDemo.java
- JRE System Library [Java SE 6 (MacOS X)]
- Inheritance
- OverloadAndOverride
- Polymorphism
- QuickSort
- SimpleRecursion
- StatementsAndOperators

*ImageDemo.java

```
import java.awt.image.BufferedImage;
import java.net.URL;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

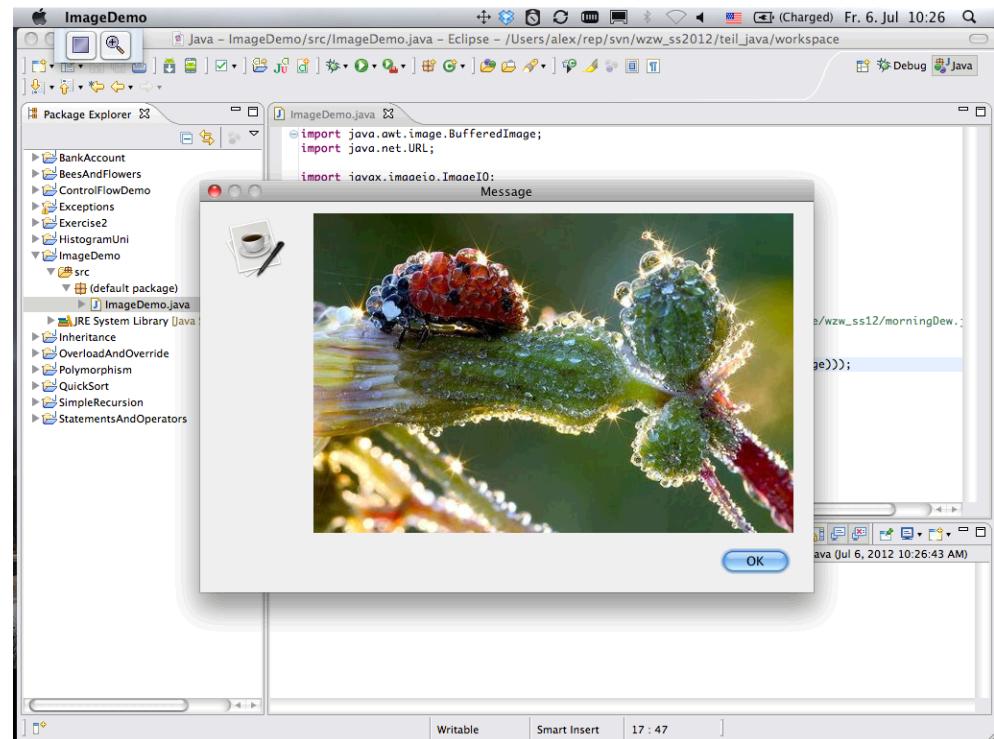
public class ImageDemo {

    public static void main(String[] args) {
        try {
            URL url = new URL("http://vmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");
            BufferedImage image = ImageIO.read(url);
            JOptionPane.showMessageDialog(null, new JLabel(new ImageIcon(image)));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Problems Javadoc Declaration Console

<terminated> ImageDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2012 10:26:43 AM)

Writable Smart Insert 17 : 85



Eclipse File Edit Run Source Navigate Project Refactor Window Help (Charged) Fr. 6. Jul 10:27

Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Package Explorer

- BankAccount
- BeesAndFlowers
- ControlFlowDemo
- Exceptions
- Exercise2
- HistogramUni
- ImageDemo
 - src
 - (default package)
 - ImageDemo.java
- JRE System Library [Java SE 6 (MacOS X)]
- Inheritance
- OverloadAndOverride
- Polymorphism
- QuickSort
- SimpleRecursion
- StatementsAndOperators

*ImageDemo.java

```
import java.awt.image.BufferedImage;
import java.net.URL;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

public class ImageDemo {

    public static void main(String[] args) {
        try {
            URL url = new URL("http://vmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");
            BufferedImage image = ImageIO.read(url);
            JOptionPane.showMessageDialog(null, new JLabel(new ImageIcon(image)));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Problems Javadoc Declaration Console

<terminated> ImageDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 6, 2012 10:26:43 AM)

Writable Smart Insert 17 : 47

8 Solving a Problem

Example Task

showMessageDialog

```
public static void showMessageDialog(Component parentComponent,
                                    Object message,
                                    String title,
                                    int messageType)
                                    throws HeadlessException
```

Brings up an information-message dialog titled "Message".

Parameters:

- parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is used.
- message - the object to display

Throws:

- HeadlessException - if GraphicsEnvironment.isHeadless returns true

See Also:

- GraphicsEnvironment.isHeadless()

showMessageDialog

```
public static void showMessageDialog(Component parentComponent,
                                    Object message,
                                    String title,
                                    int messageType)
                                    throws HeadlessException
```

Brings up a dialog that displays a message using a default icon determined by the messageType parameter.

Parameters:

- parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is used.
- message - the object to display
- title - the title string for the dialog
- messageType - the type of message to be displayed: ERROR_MESSAGE, INFORMATION_MESSAGE, WARNING_MESSAGE, QUESTION_MESSAGE, or PLAIN_MESSAGE

Throws:

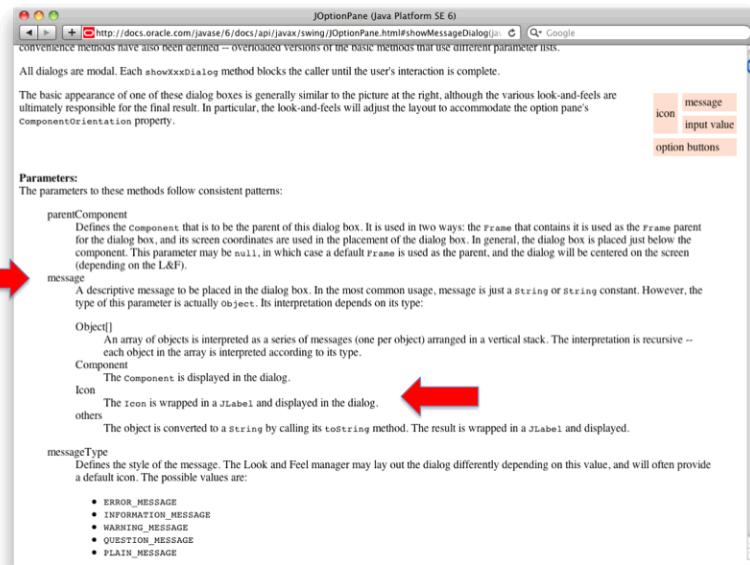
- HeadlessException - if GraphicsEnvironment.isHeadless returns true

See Also:

- GraphicsEnvironment.isHeadless()

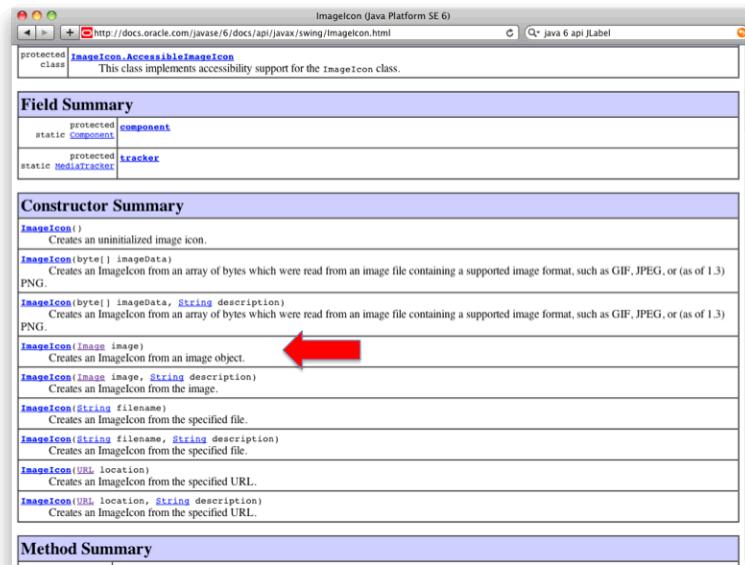
8 Solving a Problem

Example Task



8 Solving a Problem

Example Task



```

Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 6. Jul 10:29 Q
Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Java - ImageDemo/src/ImageDemo.java

import java.awt.image.BufferedImage;
import java.net.URL;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

public class ImageDemo {

    public static void main(String[] args) {
        try {
            // URL url = new URL("http://wmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");
            URL url = new URL("file:///tmp/ImageDemo/morningDew.jpg");
            BufferedImage image = ImageIO.read(url);
            ImageIcon imageIcon = new ImageIcon(image);
            JOptionPane.showMessageDialog(null, new JLabel(new ImageIcon(image)));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 6. Jul 10:30 Q
Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Java - ImageDemo/src/ImageDemo.java

import java.awt.image.BufferedImage;
import java.net.URL;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

public class ImageDemo {

    public static void main(String[] args) {
        try {
            // URL url = new URL("http://wmschlichter7.informatik.tu-muenchen.de/wzw_ss12/morningDew.jpg");
            URL url = new URL("file:///tmp/ImageDemo/morningDew.jpg");
            BufferedImage image = ImageIO.read(url);
            ImageIcon imageIcon = new ImageIcon(image);
            JOptionPane.showMessageDialog(null, new JLabel(new ImageIcon(image)));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

8 Solving a Problem

Example Task



java.awt.image
Class BufferedImage

All Implemented Interfaces:
 `RenderedImage, WritableRenderedImage, Transparency`

This class describes an `Image` with an accessible buffer of image data. A `BufferedImage` is comprised of a `ColorModel` and a `Raster` of image data. The number and types of bands in the `Raster` must match the number and types required by the `ColorModel` to represent its color and alpha components. All `BufferedImage` objects have an upper left corner coordinate of (0, 0). Any `Raster` used to construct a `BufferedImage` must therefore have `minX=0` and `minY=0`.

This class relies on the data fetching and setting methods of `Raster`, and on the color characterization methods of `ColorModel`.

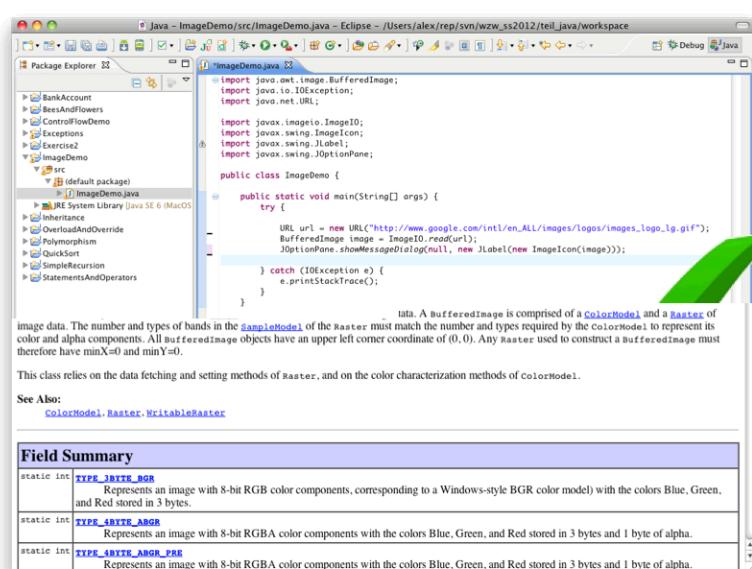
See Also:
 `ColorModel, Raster, WritableRaster`

Field Summary

<code>static int TYPE_BYTE_BGR</code>	Represents an image with 8-bit RGB color components, corresponding to a Windows-style BGR color model with the colors Blue, Green, and Red stored in 3 bytes.
<code>static int TYPE_BYTE_ABGR</code>	Represents an image with 8-bit RGBA color components with the colors Blue, Green, and Red stored in 3 bytes and 1 byte of alpha.
<code>static int TYPE_BYTE_ABGR_PFA</code>	Represents an image with 8-bit RGBA color components with the colors Blue, Green, and Red stored in 3 bytes and 1 byte of alpha.

8 Solving a Problem

Example Task



```
ImageDemo.java
```

```
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.net.URL;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

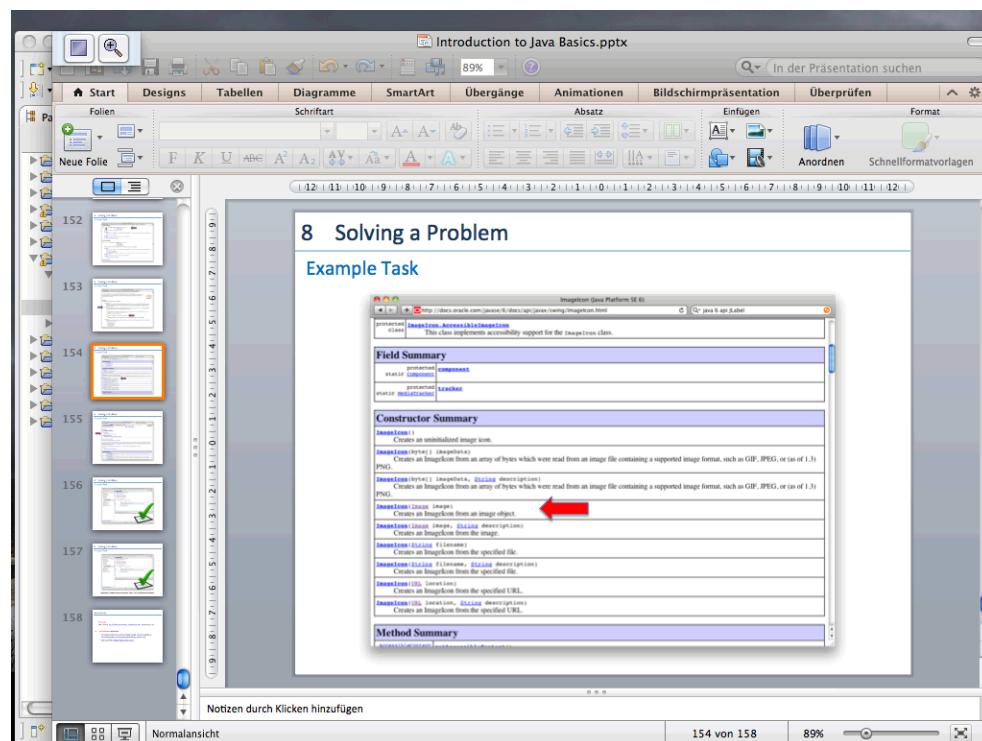
public class ImageDemo {
    public static void main(String[] args) {
        try {
            URL url = new URL("http://www.google.com/intl/en_ALL/images/logos/images_logo_lg.gif");
            BufferedImage image = ImageIO.read(url);
            JOptionPane.showMessageDialog(null, new JLabel(new ImageIcon(image)));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

This class relies on the data fetching and setting methods of `Raster`, and on the color characterization methods of `ColorModel`.

See Also:
 `ColorModel, Raster, WritableRaster`

Field Summary

<code>static int TYPE_BYTE_BGR</code>	Represents an image with 8-bit RGB color components, corresponding to a Windows-style BGR color model with the colors Blue, Green, and Red stored in 3 bytes.
<code>static int TYPE_BYTE_ABGR</code>	Represents an image with 8-bit RGBA color components with the colors Blue, Green, and Red stored in 3 bytes and 1 byte of alpha.
<code>static int TYPE_BYTE_ABGR_PFA</code>	Represents an image with 8-bit RGBA color components with the colors Blue, Green, and Red stored in 3 bytes and 1 byte of alpha.



Introduction to Java Basics.pptx

8 Solving a Problem

Example Task

8 Solving a Problem

Example Task

ImageIcon (Java Platform SE 6)

`protected ImageIcon(Icon icon)`
This class implements accessibility support for the `ImageIcon` class.

Field Summary

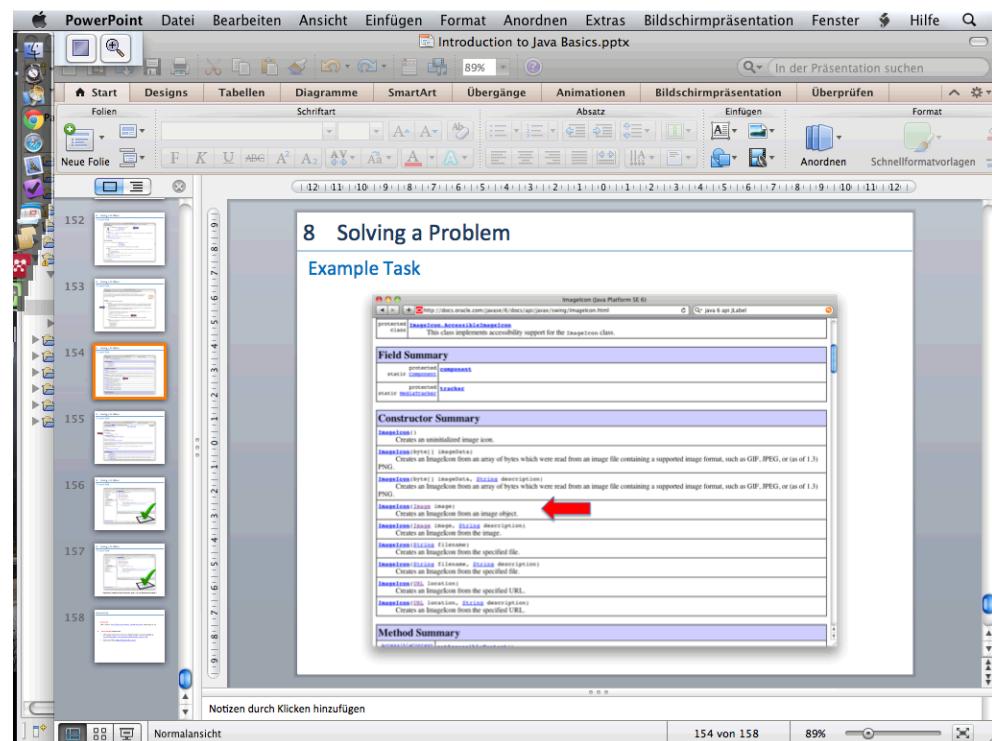
<code>protected component</code>	
<code>protected Icon icon</code>	

Constructor Summary

<code>ImageIcon()</code>	Creates an uninitialized image icon.
<code>ImageIcon(Icon icon)</code>	Creates an <code>ImageIcon</code> from an array of bytes which were read from an image file containing a supported image format, such as GIF, JPEG, or (as of 1.3) PNG.
<code>ImageIcon(Icon icon, String description)</code>	Creates an <code>ImageIcon</code> from an array of bytes which were read from an image file containing a supported image format, such as GIF, JPEG, or (as of 1.3) PNG.
<code>ImageIcon(Image image)</code>	Creates an <code>ImageIcon</code> from an image object.
<code>ImageIcon(Image image, String description)</code>	Creates an <code>ImageIcon</code> from the image.
<code>ImageIcon(String location)</code>	Creates an <code>ImageIcon</code> from the specified file.
<code>ImageIcon(File file, String description)</code>	Creates an <code>ImageIcon</code> from the specified file.
<code>ImageIcon(URL location)</code>	Creates an <code>ImageIcon</code> from the specified URL.
<code>ImageIcon(URL location, String description)</code>	Creates an <code>ImageIcon</code> from the specified URL.

Method Summary

<code>ImageIcon(Icon icon)</code>	Creates an <code>ImageIcon</code> from the specified icon.
<code>ImageIcon(String location)</code>	Creates an <code>ImageIcon</code> from the specified location.
<code>ImageIcon(URL location)</code>	Creates an <code>ImageIcon</code> from the specified URL.



Introduction to Java Basics.pptx

8 Solving a Problem

Example Task

8 Solving a Problem

Example Task

ImageIcon (Java Platform SE 6)

`protected ImageIcon(Icon icon)`
This class implements accessibility support for the `ImageIcon` class.

Field Summary

<code>protected component</code>	
<code>protected Icon icon</code>	

Constructor Summary

<code>ImageIcon()</code>	Creates an uninitialized image icon.
<code>ImageIcon(Icon icon)</code>	Creates an <code>ImageIcon</code> from an array of bytes which were read from an image file containing a supported image format, such as GIF, JPEG, or (as of 1.3) PNG.
<code>ImageIcon(Icon icon, String description)</code>	Creates an <code>ImageIcon</code> from an array of bytes which were read from an image file containing a supported image format, such as GIF, JPEG, or (as of 1.3) PNG.
<code>ImageIcon(Image image)</code>	Creates an <code>ImageIcon</code> from an image object.
<code>ImageIcon(Image image, String description)</code>	Creates an <code>ImageIcon</code> from the image.
<code>ImageIcon(String location)</code>	Creates an <code>ImageIcon</code> from the specified file.
<code>ImageIcon(File file, String description)</code>	Creates an <code>ImageIcon</code> from the specified file.
<code>ImageIcon(URL location)</code>	Creates an <code>ImageIcon</code> from the specified URL.
<code>ImageIcon(URL location, String description)</code>	Creates an <code>ImageIcon</code> from the specified URL.

