

Script generated by TTT

Title: Lehmann: Uebung_Einf_HF (12.07.2013)
Date: Fri Jul 12 11:14:59 CEST 2013
Duration: 86:03 min
Pages: 89

4 Recursion

Deepening readings:

<http://en.wikipedia.org/wiki/Recursion>
<http://en.wikipedia.org/wiki/Factorial>
http://en.wikipedia.org/wiki/Tower_of_Hanoi

4 Recursion

- **Recursion:** Divide a given problem into subproblems of the same type
 - One or more base cases
 - Rules to reduce other cases towards base case
- Example:
 - Factorial $n! = \begin{cases} 1 & \text{if } n = 0, \\ (n - 1)! \cdot n & \text{if } n > 0. \end{cases}$
 - Fibonacci $f(n) = \begin{cases} 1 & \text{if } n = 1, \\ 1 & \text{if } n = 2, \\ f(n - 1) + f(n - 2) & \text{if } n > 2. \end{cases}$

4 Recursion

- Example:
 - People are standing in queue
 - Doorman wants to know how many people are waiting
 - What are the **base-** and **general cases?**



Introduction to Java Basics (page 102 of 168) Fr. 12.Jul 11:20

4 Recursion

- Example:
 - People are standing in queue
 - Doorman wants to know how many people are waiting
 - What are the **base- and general cases?**

The diagram shows a doorman standing at an open doorway. A green speech bubble above him contains the number '6'. To his right, a queue of four people is shown. Red arrows point from each person's head to the next person in the queue, except for the last person who has a red arrow pointing down to a thought bubble containing three exclamation marks ('!!!'). This visualizes how a recursive function calls itself with a smaller argument until it reaches a base case (the thought bubble).

Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace Fr. 12.Jul 11:20

Proj Pac Typ

Factorial.java

```

1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of `n`.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of `n`.
16     */
17    // public static long factorialRecursive(int n) {
18    // }
19
20
21
22    public static void main(String[] args) {
23        System.out.println( factorialIterative(5) );
24        // System.out.println( factorialRecursive(5) );
25    }
26
27 }

```

Eclipse

VK

Console

No consoles to display at this time.

Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace Fr. 12.Jul 11:21

Proj Pac Typ

Factorial.java

```

1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of `n`.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of `n`.
16     */
17    // public static long factorialRecursive(int n) {
18    // }
19
20
21
22    public static void main(String[] args) {
23        System.out.println( factorialIterative(5) );
24        // System.out.println( factorialRecursive(5) );
25    }
26
27 }

```

Problems Javadoc Declaration Console

No consoles to display at this time.

Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace Fr. 12.Jul 11:21

Proj Pac Typ

Factorial.java

```

1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of `n`.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of `n`.
16     */
17    public static long factorialRecursive(int n) {
18
19    }
20
21
22    public static void main(String[] args) {
23        System.out.println( factorialIterative(5) );
24        // System.out.println( factorialRecursive(5) );
25    }
26
27 }

```

Problems Javadoc Declaration Console

No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:22 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ
*Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of `n`.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of `n`.
16     */
17    public static long factorialRecursive(int n) {
18        if (n == 1) {
19            return 1;
20        }
21
22        public static void main(String[] args) {
23            System.out.println( factorialIterative(5) );
24            // System.out.println( factorialRecursive(5) );
25        }
26    }
27
28 }
```

Problems @ Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:23 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ
*Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of `n`.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of `n`.
16     */
17    public static long factorialRecursive(int n) {
18        if (n == 1) {
19            return 1;
20        } else {
21            factorialRecursive(n-1);
22        }
23
24    }
25
26    public static void main(String[] args) {
27        System.out.println( factorialIterative(5) );
28        // System.out.println( factorialRecursive(5) );
29    }
30 }
```

Problems @ Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:24 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ
*Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of `n`.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of `n`.
16     */
17    public static long factorialRecursive(int n) {
18        if (n == 1) {
19            return 1;
20        } else {
21            factorialRecursive(n-1);
22        }
23
24    }
25
26    public static void main(String[] args) {
27        System.out.println( factorialIterative(5) );
28        // System.out.println( factorialRecursive(5) );
29    }
30 }
```

Problems @ Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:25 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ
*Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of `n`.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of `n`.
16     */
17    public static long factorialRecursive(int n) {
18        if (n <= 1) {
19            return 1;
20        } else {
21            factorialRecursive(n-1);
22        }
23
24    }
25
26    public static void main(String[] args) {
27        System.out.println( factorialIterative(5) );
28        // System.out.println( factorialRecursive(5) );
29    }
30 }
```

Problems @ Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:25 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ
*Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of 'n'.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of 'n'.
16     */
17    public static long factorialRecursive(int n) {
18        if (n <= 1) {
19            return 1;
20        } else {
21            factorialRecursive(n-1);
22        }
23    }
24
25
26    public static void main(String[] args) {
27        System.out.println( factorialIterative(5) );
28        // System.out.println( factorialRecursive(5) );
29    }
}
```

Problems @ Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:26 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ
*Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of 'n'.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of 'n'.
16     */
17    public static long factorialRecursive(int n) {
18        if (n < 0) {
19            System.out.println("Kaputti!");
20        } else if (n <= 1) {
21            return 1;
22        } else {
23            factorialRecursive(n-1);
24        }
25
26
27    public static void main(String[] args) {
28        System.out.println( factorialIterative(5) );
29    }
}
```

Problems @ Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:26 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ
*Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of 'n'.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of 'n'.
16     */
17    public static long factorialRecursive(int n) {
18        if (n < 0) {
19            System.out.println("Kaputti!");
20        } else if (n <= 1) {
21            return 1;
22        } else {
23            factorialRecursive(n-1);
24        }
25
26
27    public static void main(String[] args) {
28        System.out.println( factorialIterative(5) );
29    }
}
```

Problems @ Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:27 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ
*Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of 'n'.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of 'n'.
16     */
17    public static long factorialRecursive(int n) {
18        if (n < 0) {
19            System.out.println("Kaputti!");
20        } else if (n <= 1) {
21            return 1;
22        } else {
23            long a = factorialRecursive(n-1);
24            result = result * a;
25        }
26
27
28    public static void main(String[] args) {
29    }
}
```

Problems @ Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:27 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of 'n'.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of 'n'.
16     */
17    This method must return a result of type long|rsive(int,n) {
18        if (n < 0) {
19            System.err.println("Kaputti!");
20        } else if (n <= 1) {
21            return 1;
22        } else {
23            long a = factorialRecursive(n-1);
24            return n * a;
25        }
26    }
27
28
29    public static void main(String[] args) {
30
31        // Your code here
32    }
33}
```

Problems Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:28 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of 'n'.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of 'n'.
16     */
17    public static long factorialRecursive(int n) {
18        if (n < 0) {
19            System.err.println("Kaputti!");
20        } else if (n <= 1) {
21            return 1;
22        } else {
23            long a = factorialRecursive(n-1);
24            return n * a;
25        }
26    }
27
28
29    public static void main(String[] args) {
30
31        // Your code here
32    }
33}
```

Problems Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:28 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ Factorial.java
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of 'n'.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14    /**
15     * The recursive way of computing the factorial of 'n'.
16     */
17    public static long factorialRecursive(int n) {
18        if (n < 0) {
19            System.err.println("Kaputti!");
20            return -1;
21        } else if (n <= 1) {
22            return 1;
23        } else {
24            long a = factorialRecursive(n-1);
25            return n * a;
26        }
27    }
28
29    public static void main(String[] args) {
30
31        // Your code here
32    }
33}
```

Problems Javadoc Declaration Console No consoles to display at this time.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:31 Java - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```
Proj Pac Typ Factorial.java
1 public static long factorialIterative(int n) {
2     long result = 1;
3     for (int i = n; i > 1; i--) {
4         result = result * i;
5     }
6     return result;
7 }
8
9 /**
10  * The recursive way of computing the factorial of 'n'.
11 */
12 public static long factorialRecursive(int n) {
13     if (n < 0) {
14         System.err.println("Kaputti!");
15         return -1;
16     } else if (n <= 1) {
17         return 1;
18     } else {
19         long a = factorialRecursive(n-1);
20         return n * a;
21     }
22 }
23
24 public static void main(String[] args) {
25     System.out.println( factorialIterative(5) );
26     System.out.println( factorialRecursive(5) );
27 }
```

Problems Javadoc Declaration Console <terminated> Factorial [Java Application] / Parameters: n Press 'F2' for focus

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:31 Debug - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Quick Access

Debug

Factorial [Java Application]

Factorial at localhost:49326

Thread [main] (Suspended (breakpoint at line 18 in Factorial))

Factorial.factorialRecursive(int) line: 18

Factorial.main(String[]) line: 32

/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12)

Variables

| Name | Value |
|------|-------|
| n | 5 |

Breakpoints

Factorial.java

```
14*    * The recursive way of computing the factorial of `n`.
15
16    */
17
18 public static long factorialRecursive(int n) {
19     if (n < 0) {
20         System.err.println("Kaputt!");
21         return -1;
22     } else if (n <= 1) {
23         return 1;
24     } else {
25         long a = factorialRecursive(n-1);
26         long b = n * a;
27         return b;
28     }
29 }
```

Outline

Factorial

- factorialIterative(int) : long
- factorialRecursive(int) : long
- main(String[]) : void

Console

Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)

120

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:32 Debug - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Quick Access

Debug

Factorial [Java Application]

Factorial at localhost:49326

Thread [main] (Suspended (breakpoint at line 18 in Factorial))

Factorial.factorialRecursive(int) line: 18

Factorial.main(String[]) line: 32

/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12)

Variables

| Name | Value |
|------|-------|
| n | 5 |

Breakpoints

Factorial.java

```
14*    * The recursive way of computing the factorial of `n`.
15
16    */
17
18 public static long factorialRecursive(int n) {
19     if (n < 0) {
20         System.err.println("Kaputt!");
21         return -1;
22     } else if (n <= 1) {
23         return 1;
24     } else {
25         long a = factorialRecursive(n-1);
26         long b = n * a;
27         return b;
28     }
29 }
```

Outline

Factorial

- factorialIterative(int) : long
- factorialRecursive(int) : long
- main(String[]) : void

Console

Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)

120

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:33 Debug - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Quick Access

Debug

Factorial [Java Application]

Factorial at localhost:49326

Thread [main] (Suspended (breakpoint at line 18 in Factorial))

Factorial.factorialRecursive(int) line: 18

Factorial.main(String[]) line: 24

/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12)

Variables

| Name | Value |
|------|-------|
| n | 5 |

Breakpoints

Factorial.java

```
15
16    * The recursive way of computing the factorial of `n`.
17
18 public static long factorialRecursive(int n) {
19     if (n < 0) {
20         System.err.println("Kaputt!");
21         return -1;
22     } else if (n <= 1) {
23         return 1;
24     } else {
25         long a = factorialRecursive(n-1);
26         long b = n * a;
27         return b;
28     }
29 }
```

Outline

Factorial

- factorialIterative(int) : long
- factorialRecursive(int) : long
- main(String[]) : void

Console

Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)

120

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:34 Debug - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Quick Access

Debug

Factorial [Java Application]

Factorial at localhost:49326

Thread [main] (Suspended (breakpoint at line 18 in Factorial))

Factorial.factorialRecursive(int) line: 18

Factorial.main(String[]) line: 32

/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12)

Variables

| Name | Value |
|------|-------|
| n | 4 |

Breakpoints

Factorial.java

```
15
16    * The recursive way of computing the factorial of `n`.
17
18 public static long factorialRecursive(int n) {
19     if (n < 0) {
20         System.err.println("Kaputt!");
21         return -1;
22     } else if (n <= 1) {
23         return 1;
24     } else {
25         long a = factorialRecursive(n-1);
26         long b = n * a;
27         return b;
28     }
29 }
```

Outline

Factorial

- factorialIterative(int) : long
- factorialRecursive(int) : long
- main(String[]) : void

Console

Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)

120

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:34 Debug - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Quick Access

Debug

Factorial [Java Application]

Factorial at localhost:49326

Thread [main] (Suspended (breakpoint at line 18 in Factorial))

Factorial.factorialRecursive(int) line: 18

Factorial.factorialRecursive(int) line: 24

Factorial.factorialRecursive(int) line: 24

Factorial.main(String[]) line: 32

/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12)

Variables

| Name | Value |
|------|-------|
| n | 3 |

Breakpoints

Factorial.java

```
15 * The recursive way of computing the factorial of `n`.
16 */
17 public static long factorialRecursive(int n) {
18     if (n < 0) {
19         System.out.println("Kaputt!");
20         return -1;
21     } else if (n <= 1) {
22         return 1;
23     } else {
24         long a = factorialRecursive(n-1);
25         return n * a;
26     }
27 }
```

Outline

Factorial

- factorialIterative(int) : long
- factorialRecursive(int) : long
- main(String[]) : void

Console

Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)

120

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:35 Debug - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Quick Access

Debug

Factorial [Java Application]

Factorial at localhost:49326

Thread [main] (Suspended (breakpoint at line 18 in Factorial))

Factorial.factorialRecursive(int) line: 18

Factorial.factorialRecursive(int) line: 24

Factorial.factorialRecursive(int) line: 24

Factorial.factorialRecursive(int) line: 24

Factorial.factorialRecursive(int) line: 24

Factorial.main(String[]) line: 32

/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12)

Variables

| Name | Value |
|------|-------|
| n | 1 |

Breakpoints

Factorial.java

```
15 * The recursive way of computing the factorial of `n`.
16 */
17 public static long factorialRecursive(int n) {
18     if (n < 0) {
19         System.out.println("Kaputt!");
20         return -1;
21     } else if (n <= 1) {
22         return 1;
23     } else {
24         long a = factorialRecursive(n-1);
25         return n * a;
26     }
27 }
```

Outline

Factorial

- factorialIterative(int) : long
- factorialRecursive(int) : long
- main(String[]) : void

Console

Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)

120

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:35 Debug - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Quick Access

Debug

Factorial [Java Application]

Factorial at localhost:49326

Thread [main] (Suspended (breakpoint at line 24 in Factorial))

Factorial.factorialRecursive(int) line: 24

Factorial.factorialRecursive(int) line: 24

Factorial.factorialRecursive(int) line: 24

Factorial.factorialRecursive(int) line: 24

Factorial.main(String[]) line: 32

/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12)

Variables

| Name | Value |
|------|-------|
| n | 2 |

Breakpoints

Factorial.java

```
15 * The recursive way of computing the factorial of `n`.
16 */
17 public static long factorialRecursive(int n) {
18     if (n < 0) {
19         System.out.println("Kaputt!");
20         return -1;
21     } else if (n <= 1) {
22         return 1;
23     } else {
24         long a = factorialRecursive(n-1);
25         return n * a;
26     }
27 }
```

Outline

Factorial

- factorialIterative(int) : long
- factorialRecursive(int) : long
- main(String[]) : void

Console

Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)

120

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:35 Debug - SimpleRecursion/src/Factorial.java - Eclipse - /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Quick Access

Debug

Factorial [Java Application]

Factorial at localhost:49326

Thread [main] (Suspended (breakpoint at line 25 in Factorial))

Factorial.factorialRecursive(int) line: 25

Factorial.factorialRecursive(int) line: 24

Factorial.factorialRecursive(int) line: 24

Factorial.factorialRecursive(int) line: 24

Factorial.main(String[]) line: 32

/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12)

Variables

| Name | Value |
|------|-------|
| n | 2 |
| a | 1 |

Breakpoints

Factorial.java

```
22     return 1;
23 } else {
24     long a = factorialRecursive(n-1);
25     return n * a;
26 }
27 }

public static void main(String[] args) {
    System.out.println( factorialIterative(5) );
    System.out.println( factorialRecursive(5) );
}
```

Outline

Factorial

- factorialIterative(int) : long
- factorialRecursive(int) : long
- main(String[]) : void

Console

Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)

120

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:36 Debug – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Quick Access

Debug Factorial [Java Application] Factorial at localhost:49326 Thread (main) (Suspended) Factorial.factorialRecursive(int) line: 24 Factorial.factorialRecursive(int) line: 24 Factorial.factorialRecursive(int) line: 24 Factorial.main(String[]) line: 32 /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)

Variables

| Name | Value |
|------|-------|
| n | 3 |

Factorial.java

```
22     return 1;
23 } else {
24     long a = factorialRecursive(n-1);
25     return n * a;
26 }
27
28
29 public static void main(String[] args) {
30     System.out.println( factorialIterative(5) );
31     System.out.println( factorialRecursive(5) );
32 }
```

Console

```
Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)
120
```

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:36 Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java

```
8     for (int i = n; i > 1; i--) {
9         result = result * i;
10    }
11    return result;
12 }
13 /**
14 * The recursive way of computing the factorial of `n`.
15 */
16 public static long factorialRecursive(int n) {
17     if (n < 0) {
18         System.err.println("Kaputt!");
19         return -1;
20     } else if (n <= 1) {
21         return 1;
22     } else {
23         long a = factorialRecursive(n-1);
24         return n * a;
25     }
26 }
27
28
29 public static void main(String[] args) {
30     System.out.println( factorialIterative(5) );
31     System.out.println( factorialRecursive(5) );
32 }
```

Problems Javadoc Declaration Console

```
<terminated> Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 12:0
120
120
```

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:37 Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java

```
8     for (int i = n; i > 1; i--) {
9         result = result * i;
10    }
11    return result;
12 }
13 /**
14 * The recursive way of computing the factorial of `n`.
15 */
16 public static long factorialRecursive(int n) {
17     if (n < 0) {
18         System.err.println("Kaputt!");
19         return -1;
20     } else if (n <= 1) {
21         return 1;
22     } else {
23         long a = factorialRecursive(n-1);
24         return n * a;
25     }
26 }
27
28
29 public static void main(String[] args) {
30     System.out.println( factorialIterative(5) );
31     System.out.println( factorialRecursive(5) );
32 }
```

Problems Javadoc Declaration Console

```
<terminated> Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 12:0
120
120
```

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:37 Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java

```
1 public class Factorial {
2
3     /**
4      * The iterative way of computing the factorial of `n`.
5      */
6     public static long factorialIterative(int n) {
7         long result = 1;
8         for (int i = n; i > 1; i--) {
9             result = result * i;
10        }
11        return result;
12    }
13
14 /**
15 * The recursive way of computing the factorial of `n`.
16 */
17 public static long factorialRecursive(int n) {
18     if (n < 0) {
19         System.err.println("Kaputt!");
20         return -1;
21     } else if (n <= 1) {
22         return 1;
23     } else {
24         long a = factorialRecursive(n-1);
25         return n * a;
26     }
27 }
28
29
```

Problems Javadoc Declaration Console

```
<terminated> Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 12:0
120
120
```

Eclipse IDE interface showing the Factorial.java file. The code contains two implementations of the factorial function: an iterative version and a recursive version. The terminal window shows the application has terminated successfully.

```

public class Factorial {
    /**
     * The iterative way of computing the factorial of 'n'.
     */
    public static long factorialIterative(int n) {
        long result = 1;
        for (int i = n; i > 1; i--) {
            result = result * i;
        }
        return result;
    }

    /**
     * The recursive way of computing the factorial of 'n'.
     */
    public static long factorialRecursive(int n) {
        if (n < 0) {
            System.err.println("Kaputt!");
            return -1;
        } else if (n <= 1) {
            return 1;
        } else {
            long a = factorialRecursive(n-1);
            return n * a;
        }
    }
}

```

Console output:

```

<terminated> Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013)
120
120

```

4 Recursion

Recursive method calls & Stack

- Local variables and parameters stored on stack
- For each function call, a corresponding stack frame is created

```

long factorial(int n) {
    long temp;
    if (n == 1) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}

public static void main(String[] args) {
    long result;
    result = factorial(4);
    System.out.println(result);
}

```

4 Recursion

Recursive method calls & Stack

- Local variables and parameters stored on stack
- For each function call, a corresponding stack frame is created

```

long factorial(int n) {
    long temp;
    if (n == 1) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}

public static void main(String[] args) {
    long result;
    result = factorial(4);
    System.out.println(result);
}

```

A vertical stack diagram on the right shows the stack frames for the recursive calls. It starts with a base frame at the bottom, followed by three recursive frames for n=3, n=2, and n=1, and finally the result frame at the top.

```

0
4
...
0
...

```

Eclipse IDE interface showing the Factorial.java file. The code is identical to the one in the other windows. The terminal window shows the application has terminated successfully.

```

public class Factorial {
    /**
     * The iterative way of computing the factorial of 'n'.
     */
    public static long factorialIterative(int n) {
        long result = 1;
        for (int i = n; i > 1; i--) {
            result = result * i;
        }
        return result;
    }

    /**
     * The recursive way of computing the factorial of 'n'.
     */
    public static long factorialRecursive(int n) {
        if (n < 0) {
            System.err.println("Kaputt!");
            return -1;
        } else if (n <= 1) {
            return 1;
        } else {
            long a = factorialRecursive(n-1);
            return n * a;
        }
    }
}

```

Console output:

```

<terminated> Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013)
120
120

```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:50 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
13    }
14    }
15  }
16 }
17 }
18 /**
19 * Prints the elements of the given array of integers to the console.
20 */
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27 /**
28 * Searches the given sorted array between left and right for the given
29 * element. Returns true if the element was found, else false.
30 */
31 // public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
32 // }
33 /**
34 * Prints the elements of the given array of integers to the console.
35 */
36 public static void main(String[] args) {
37     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
38 }
39
40 }
```

Problems Javadoc Declaration Console

```
<terminated> Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013)
120
120
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:51 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
13    }
14    }
15  }
16 }
17 }
18 /**
19 * Prints the elements of the given array of integers to the console.
20 */
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27 /**
28 * Searches the given sorted array between left and right for the given
29 * element. Returns true if the element was found, else false.
30 */
31 // public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
32 // }
33 /**
34 * Prints the elements of the given array of integers to the console.
35 */
36 public static void main(String[] args) {
37     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
38 }
39
40 }
```

Problems Javadoc Declaration Console

```
<terminated> Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013)
120
120
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:51 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
13    }
14    }
15  }
16 }
17 }
18 /**
19 * Prints the elements of the given array of integers to the console.
20 */
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27 /**
28 * Searches the given sorted array between left and right for the given
29 * element. Returns true if the element was found, else false.
30 */
31 // public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
32 // }
33 /**
34 * Prints the elements of the given array of integers to the console.
35 */
36 public static void main(String[] args) {
37     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
38     bub;
39 }
40 }
```

Problems Javadoc Declaration Console

```
<terminated> Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013)
120
120
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:52 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
13    }
14    }
15  }
16 }
17 }
18 /**
19 * Prints the elements of the given array of integers to the console.
20 */
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27 /**
28 * Searches the given sorted array between left and right for the given
29 * element. Returns true if the element was found, else false.
30 */
31 // public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
32 // }
33 /**
34 * Prints the elements of the given array of integers to the console.
35 */
36 public static void main(String[] args) {
37     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
38     bubbleSort(someNumbers);
39     printArray(someNumbers);
40 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013)
1 2 5 7 14 21 84
34 : 30
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:53 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
13     numbers[j+1] = tmp;
14 }
15 }
16 }
17 }
18 /**
19 * Prints the elements of the given array of integers to the console.
20 */
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27 /**
28 * Searches the given sorted array between left and right for the given
29 * element. Returns true if the element was found, else false.
30 */
31 public static boolean binarySearch(int[] numbers, int left, int right) {
32     int pivotIndex = (left + right) / 2;
33 }
34
35 public static void main(String[] args) {
36     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
37
38     bubbleSort(someNumbers);
39     printArray(someNumbers);
40
41     binarySearch(2, someNumbers, 0, someNumbers.length-1);
42
43 }
44
45 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:53 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19 /**
20 * Prints the elements of the given array of integers to the console.
21 */
22 public static void printArray(int[] numbers) {
23     for (int i = 0; i < numbers.length; i++) {
24         System.out.print(numbers[i] + "\t");
25     }
26     System.out.println();
27 }
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32 public static boolean binarySearch(int[] numbers, int left, int right) {
33     int pivotIndex = (left + right) / 2;
34 }
35
36 public static void main(String[] args) {
37     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
38
39     bubbleSort(someNumbers);
40     printArray(someNumbers);
41
42     binarySearch();
43 }
44
45 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:54 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19 /**
20 * Prints the elements of the given array of integers to the console.
21 */
22 public static void printArray(int[] numbers) {
23     for (int i = 0; i < numbers.length; i++) {
24         System.out.print(numbers[i] + "\t");
25     }
26     System.out.println();
27 }
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32 public static boolean binarySearch(int[] numbers, int left, int right) {
33     int pivotIndex = (left + right) / 2;
34 }
35
36 public static void main(String[] args) {
37     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
38
39     bubbleSort(someNumbers);
40     printArray(someNumbers);
41
42     binarySearch(2, someNumbers, 0, someNumbers.length-1);
43
44 }
45
46 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:54 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19 /**
20 * Prints the elements of the given array of integers to the console.
21 */
22 public static void printArray(int[] numbers) {
23     for (int i = 0; i < numbers.length; i++) {
24         System.out.print(numbers[i] + "\t");
25     }
26     System.out.println();
27 }
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32 public static boolean binarySearch(int[] numbers, int left, int right) {
33     int pivotIndex = (left + right) / 2;
34 }
35
36 public static void main(String[] args) {
37     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
38
39     bubbleSort(someNumbers);
40     printArray(someNumbers);
41
42     binarySearch(2, someNumbers, 0, someNumbers.length-1);
43 }
44
45 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:56 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19 /**
 * Prints the elements of the given array of integers to the console.
 */
20 public static void printArray(int[] numbers) {
21     for (int i = 0; i < numbers.length; i++) {
22         System.out.print(numbers[i] + "\t");
23     }
24     System.out.println();
25 }
26
27 /**
 * Searches the given sorted array between left and right for the given
 * element. Returns true if the element was found, else false.
 */
28 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
29     int pivotIndex = (left + right) / 2;
30     This method must return a result of type boolean
31     2 quick fixes available:
32     Add return statement
33     Change return type to 'void'
34 }
35
36 public static void main(String[] args) {
37     int[] someNumbers = { 21, 14, 84, 7 };
38     bubbleSort(someNumbers);
39     printArray(someNumbers);
40
41     binarySearch(2, someNumbers, 0, someNumbers.length-1);
42 }
43
44
45
46
47 }
```

Problems @ Javadoc Declaration Console <terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)

1 2 5 7 14 21 84

This method must return a result of type boolean Writable Smart Insert 33 : 45

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:57 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19 /**
 * Prints the elements of the given array of integers to the console.
 */
20 public static void printArray(int[] numbers) {
21     for (int i = 0; i < numbers.length; i++) {
22         System.out.print(numbers[i] + "\t");
23     }
24     System.out.println();
25 }
26
27 /**
 * Searches the given sorted array between left and right for the given
 * element. Returns true if the element was found, else false.
 */
28 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
29     int pivotIndex = (left + right) / 2;
30     if () {
31
32
33
34
35
36
37
38
39
39 public static void main(String[] args) {
40     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
41     bubbleSort(someNumbers);
42     printArray(someNumbers);
43
44     binarySearch(2, someNumbers, 0, someNumbers.length-1);
45 }
46
47 }
```

Problems @ Javadoc Declaration Console <terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)

1 2 5 7 14 21 84

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:57 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19 /**
 * Prints the elements of the given array of integers to the console.
 */
20 public static void printArray(int[] numbers) {
21     for (int i = 0; i < numbers.length; i++) {
22         System.out.print(numbers[i] + "\t");
23     }
24     System.out.println();
25 }
26
27 /**
 * Searches the given sorted array between left and right for the given
 * element. Returns true if the element was found, else false.
 */
28 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
29     int pivotIndex = (left + right) / 2;
30
31
32
33
34
35
36
37
38
39
39 public static void main(String[] args) {
40     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
41     bubbleSort(someNumbers);
42     printArray(someNumbers);
43
44     binarySearch(2, someNumbers, 0, someNumbers.length-1);
45 }
46
47 }
```

Problems @ Javadoc Declaration Console <terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)

1 2 5 7 14 21 84

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:58 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19 /**
 * Prints the elements of the given array of integers to the console.
 */
20 public static void printArray(int[] numbers) {
21     for (int i = 0; i < numbers.length; i++) {
22         System.out.print(numbers[i] + "\t");
23     }
24     System.out.println();
25 }
26
27 /**
 * Searches the given sorted array between left and right for the given
 * element. Returns true if the element was found, else false.
 */
28 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
29     int pivotIndex = (left + right) / 2;
30     if (elt == )
31
32
33
34
35
36
37
38
39
39 public static void main(String[] args) {
40     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
41     bubbleSort(someNumbers);
42     printArray(someNumbers);
43
44     binarySearch(2, someNumbers, 0, someNumbers.length-1);
45 }
46
47 }
```

Syntax error on token ")", Statement expected after this token Writable Smart Insert 36 : 20

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 11:58 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19 /**
 * Prints the elements of the given array of integers to the console.
 */
20 public static void printArray(int[] numbers) {
21     for (int i = 0; i < numbers.length; i++) {
22         System.out.print(numbers[i] + "\t");
23     }
24     System.out.println();
25 }
26
27 /**
 * Searches the given sorted array between left and right for the given
 * element. Returns true if the element was found, else false.
 */
28 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
29     int pivotIndex = (left + right) / 2;
30     if (elt == )
31
32
33
34
35
36
37
38
39
39 public static void main(String[] args) {
40     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
41     bubbleSort(someNumbers);
42     printArray(someNumbers);
43
44     binarySearch(2, someNumbers, 0, someNumbers.length-1);
45 }
46
47 }
```

Syntax error on token ")", Statement expected after this token Writable Smart Insert 36 : 20

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:59 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19* * Prints the elements of the given array of integers to the console.
20*
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
33     int pivotIndex = (left + right) / 2;
34
35     if (elt == someNumbers[])
36         return true;
37
38     public static void main(String[] args) {
39         int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
40
41         bubbleSort(someNumbers);
42         printArray(someNumbers);
43
44         binarySearch2(someNumbers, 0, someNumbers.length-1);
45     }
46 }
47
```

Problems @ Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:59 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19* * Prints the elements of the given array of integers to the console.
20*
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
33     int pivotIndex = (left + right) / 2;
34
35     if (elt == numbers[pivotIndex])
36         return true;
37
38     public static void main(String[] args) {
39         int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
40
41         bubbleSort(someNumbers);
42         printArray(someNumbers);
43
44         binarySearch2(someNumbers, 0, someNumbers.length-1);
45     }
46 }
47
```

Problems @ Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 11:59 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19* * Prints the elements of the given array of integers to the console.
20*
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
33     int pivotIndex = (left + right) / 2;
34
35     if (elt == numbers[pivotIndex])
36         return true;
37     else if (elt < numbers[pivotIndex])
38         return false;
39
40     public static void main(String[] args) {
41         int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
42
43         bubbleSort(someNumbers);
44         printArray(someNumbers);
45
46         binarySearch2(someNumbers, 0, someNumbers.length-1);
47     }
48 }
```

Problems @ Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 12:00 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19* * Prints the elements of the given array of integers to the console.
20*
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
33     int pivotIndex = (left + right) / 2;
34
35     if (elt == numbers[pivotIndex])
36         return true;
37     else if (elt < numbers[pivotIndex])
38         return false;
39
40     public static void main(String[] args) {
41         int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
42
43         bubbleSort(someNumbers);
44         printArray(someNumbers);
45
46         binarySearch2(someNumbers, 0, someNumbers.length-1);
47     }
48 }
```

Problems @ Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:01 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19* * Prints the elements of the given array of integers to the console.
20*
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32
33 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
34     int pivotIndex = (left + right) / 2;
35
36     if (elt == numbers[pivotIndex]) {
37         return true;
38     } else if (elt < numbers[pivotIndex]) {
39         binarySearch();
40     } else {
41
42     }
43 }
44
45 public static void main(String[] args) {
46     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
47 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:01 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19* * Prints the elements of the given array of integers to the console.
20*
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32
33 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
34     int pivotIndex = (left + right) / 2;
35
36     if (elt == numbers[pivotIndex]) {
37         return true;
38     } else if (elt < numbers[pivotIndex]) {
39         binarySearch();
40     } else {
41
42     }
43 }
44
45 public static void main(String[] args) {
46     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
47 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:01 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19* * Prints the elements of the given array of integers to the console.
20*
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32
33 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
34     int pivotIndex = (left + right) / 2;
35
36     if (elt == numbers[pivotIndex]) {
37         return true;
38     } else if (elt < numbers[pivotIndex]) {
39         boolean binarySearch(elt, numbers, left, pivotIndex-1);
40     } else {
41
42     }
43 }
44
45 public static void main(String[] args) {
46     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
47 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:02 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19* * Prints the elements of the given array of integers to the console.
20*
21 public static void printArray(int[] numbers) {
22     for (int i = 0; i < numbers.length; i++) {
23         System.out.print(numbers[i] + "\t");
24     }
25     System.out.println();
26 }
27
28 /**
29 * Searches the given sorted array between left and right for the given
30 * element. Returns true if the element was found, else false.
31 */
32
33 public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
34     int pivotIndex = (left + right) / 2;
35
36     if (elt == numbers[pivotIndex]) {
37         return true;
38     } else if (elt < numbers[pivotIndex]) {
39         return binarySearch(elt, numbers, left, pivotIndex-1);
40     } else {
41
42     }
43 }
44
45 public static void main(String[] args) {
46     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
47 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:03 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19 /**
 * Prints the elements of the given array of integers to the console.
 */
public static void printArray(int[] numbers) {
    for (int i = 0; i < numbers.length; i++) {
        System.out.print(numbers[i] + "\t");
    }
    System.out.println();
}

/**
 * Searches the given sorted array between left and right for the given
 * element. Returns true if the element was found, else false.
 */
public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
    int pivotIndex = (left + right) / 2;

    if (elt == numbers[pivotIndex]) {
        return true;
    } else if (elt < numbers[pivotIndex]) {
        return binarySearch(elt, numbers, left, pivotIndex-1);
    } else {
        return binarySearch(elt, numbers, pivotIndex+1, right);
    }
}

public static void main(String[] args) {
    int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
}
```

Problems Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:03 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
19 /**
 * Prints the elements of the given array of integers to the console.
 */
public static void printArray(int[] numbers) {
    for (int i = 0; i < numbers.length; i++) {
        System.out.print(numbers[i] + "\t");
    }
    System.out.println();
}

/**
 * Searches the given sorted array between left and right for the given
 * element. Returns true if the element was found, else false.
 */
public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
    if (left > right) {
        return false;
    }

    int pivotIndex = (left + right) / 2;

    if (elt == numbers[pivotIndex]) {
        return true;
    } else if (elt < numbers[pivotIndex]) {
        return binarySearch(elt, numbers, left, pivotIndex-1);
    } else {
        return binarySearch(elt, numbers, pivotIndex+1, right);
    }
}

public static void main(String[] args) {
```

Problems Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:04 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
31 /**
 * element. Returns true if the element was found, else false.
 */
public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
    if (left > right) {
        return false;
    }

    int pivotIndex = (left + right) / 2;

    if (elt == numbers[pivotIndex]) {
        return true;
    } else if (elt < numbers[pivotIndex]) {
        return binarySearch(elt, numbers, left, pivotIndex-1);
    } else {
        return binarySearch(elt, numbers, pivotIndex+1, right);
    }
}

public static void main(String[] args) {
    int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };

    bubbleSort(someNumbers);
    printArray(someNumbers);

    binarySearch(2, someNumbers, 0, someNumbers.length-1);
}
```

Problems Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1 2 5 7 14 21 84
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:06 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Factorial.java BinarySearch.java

```
31 /**
 * element. Returns true if the element was found, else false.
 */
public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
    if (left > right) {
        return false;
    }

    int pivotIndex = (left + right) / 2;

    if (elt == numbers[pivotIndex]) {
        return true;
    } else if (elt < numbers[pivotIndex]) {
        return binarySearch(elt, numbers, left, pivotIndex-1);
    } else {
        return binarySearch(elt, numbers, pivotIndex+1, right);
    }
}

public static void main(String[] args) {
    int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };

    bubbleSort(someNumbers);
    printArray(someNumbers);

    System.out.println( binarySearch(2, someNumbers, 0, someNumbers.length-1) );
}
```

Problems Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
true
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:07 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Proj Factorial.java BinarySearch.java

```
37     int pivotIndex = (left + right) / 2;
38
39     if (elt == numbers[pivotIndex]) {
40         return true;
41     } else if (elt < numbers[pivotIndex]) {
42         return binarySearch(elt, numbers, left, pivotIndex-1);
43     } else {
44         return binarySearch(elt, numbers, pivotIndex+1, right);
45     }
46
47 }
48
49 static {
50
51     public static void main(String[] args) {
52         int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
53
54         bubbleSort(someNumbers);
55         printArray(someNumbers);
56
57         System.out.println( binarySearch(2, someNumbers, 0, someNumbers.length-1) );
58
59         int i = 1;
60         while (i <= 10) {
61             System.out.println(i);
62             i = i + 1;
63         }
64     }
65 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
4
5
6
7
8
9
10
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:07 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Proj Factorial.java BinarySearch.java

```
37     int pivotIndex = (left + right) / 2;
38
39     if (elt == numbers[pivotIndex]) {
40         return true;
41     } else if (elt < numbers[pivotIndex]) {
42         return binarySearch(elt, numbers, left, pivotIndex-1);
43     } else {
44         return binarySearch(elt, numbers, pivotIndex+1, right);
45     }
46
47 }
48
49 static void geb1Aus() {
50
51     public static void main(String[] args) {
52         int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
53
54         bubbleSort(someNumbers);
55         printArray(someNumbers);
56
57         System.out.println( binarySearch(2, someNumbers, 0, someNumbers.length-1) );
58
59         int i = 1;
60         while (i <= 10) {
61             System.out.println(i);
62             i = i + 1;
63         }
64     }
65 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
4
5
6
7
8
9
10
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:08 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Proj Factorial.java BinarySearch.java

```
48
49 static void geb1Aus() {
50     System.out.println("1");
51 }
52
53 static void geb2Aus() {
54     geb1Aus();
55     System.out.println("2");
56 }
57
58 static void geb3Aus() {
59     geb2Aus();
60     System.out.println("3");
61 }
62
63 public static void main(String[] args) {
64     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
65
66     bubbleSort(someNumbers);
67     printArray(someNumbers);
68
69     System.out.println( binarySearch(2, someNumbers, 0, someNumbers.length-1) );
70
71     int i = 1;
72     while (i <= 10) {
73         System.out.println(i);
74         i = i + 1;
75     }
76 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
7
8
9
10
1
2
3
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12.Jul 12:09 Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Proj Factorial.java BinarySearch.java

```
41     return true;
42 } else if (elt < numbers[pivotIndex]) {
43     return binarySearch(elt, numbers, left, pivotIndex-1);
44 } else {
45     return binarySearch(elt, numbers, pivotIndex+1, right);
46 }
47
48 }
49
50 static void gebNAus(int n) {
51     if (n > 0) {
52         gebNAus(n-1);
53     }
54
55     public static void main(String[] args) {
56         int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
57
58         bubbleSort(someNumbers);
59         printArray(someNumbers);
60
61         System.out.println( binarySearch(2, someNumbers, 0, someNumbers.length-1) );
62
63         int i = 1;
64         while (i <= 10) {
65             System.out.println(i);
66             i = i + 1;
67         }
68     }
69 }
```

Problems Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
7
8
9
10
1
2
3
```

Eclipse Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```

41     return true;
42 } else if (elt < numbers[pivotIndex]) {
43     return binarySearch(elt, numbers, left, pivotIndex-1);
44 } else {
45     return binarySearch(elt, numbers, pivotIndex+1, right);
46 }
47
48
49 static void gebAus(int n) {
50     if (n > 0) {
51         gebAus();
52         System.out.println(n);
53     }
54 }
55
56 public static void main(String[] args) {
57     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
58
59     bubbleSort(someNumbers);
60     printArray(someNumbers);
61
62     System.out.println( binarySearch(2, someNumbers, 0, someNumbers.length-1) );
63
64     int i = 1;
65     while (i <= 10) {
66         System.out.println(i);
67         i = i + 1;
68     }
69 }
70

```

Problems Declaration Console

```

<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
7
8
9
10
1
2
3

```

Eclipse Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```

43     return binarySearch(elt, numbers, left, pivotIndex-1);
44 } else {
45     return binarySearch(elt, numbers, pivotIndex+1, right);
46 }
47
48
49 static void gebAus(int n) {
50     if (n > 0) {
51         gebAus(n-1);
52         System.out.println(n);
53     }
54 }
55
56 public static void main(String[] args) {
57     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
58
59     bubbleSort(someNumbers);
60     printArray(someNumbers);
61
62     System.out.println( binarySearch(2, someNumbers, 0, someNumbers.length-1) );
63
64     int i = 1;
65     while (i <= 10) {
66         System.out.println(i);
67         i = i + 1;
68     }
69
70     gebAus(3);
71

```

Problems Declaration Console

```

<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
7
8
9
10
1
2
3

```

Eclipse Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

```

43     return binarySearch(elt, numbers, left, pivotIndex-1);
44 } else {
45     return binarySearch(elt, numbers, pivotIndex+1, right);
46 }
47
48
49 static void gebAus(int n) {
50     if (n > 0) {
51         gebAus(n-1);
52         System.out.println(n);
53     }
54 }
55
56 public static void main(String[] args) {
57     int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
58
59     bubbleSort(someNumbers);
60     printArray(someNumbers);
61
62     System.out.println( binarySearch(2, someNumbers, 0, someNumbers.length-1) );
63
64     int i = 1;
65     while (i <= 10) {
66         System.out.println(i);
67         i = i + 1;
68     }
69
70     gebAus(3);
71

```

Problems Declaration Console

```

<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
7
8
9
10
1
2
3

```

Preview Java - SimpleRecursion/src/BinarySearch.java - Eclipse ~ /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

5 Error 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.

Introduction to Java Basics (page 131 of 168)

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:16 Introduction to Java Basics (page 132 of 168)

5 Error 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.

The diagram shows three methods stacked vertically:

- method 3**: Shows an **Error occurs** box with a yellow arrow pointing to a **Cannot handle exception of type ExceptionXYZ** message. A blue arrow labeled **calls** points from method 2 to method 3.
- method 2**: Shows a **Cannot handle exception of type ExceptionXYZ** message. A blue arrow labeled **calls** points from method 1 to method 2. A yellow arrow labeled **throws exception** points from method 2 to method 3.
- method 1**: Shows a **Can handle exception of type ExceptionXYZ** message. A yellow arrow labeled **fowards exception** points from method 2 to method 1. A yellow arrow labeled **catches exception (handles it)** points from method 1 back to method 2.

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:17 Introduction to Java Basics (page 133 of 168)

5 Error 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();  
}
```

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:20 Introduction to Java Basics (page 134 of 168)

5 Error Handling & Exceptions

Exceptions

- "finally" block is always executed, i.e. even if an exception occurs

```
FileWriter fileWriter = null;  
try {  
    fileWriter = new FileWriter("someFileName.txt");  
    fileWriter.write('a');  
    // ...  
} catch (Exception e) {  
    // Exception handling code goes here  
} finally {  
    // Make sure that the file is closed, no matter whether  
    // an exception occurred or not.  
    if (fileWriter != null) {  
        fileWriter.close();  
    }  
}
```

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:21 Introduction to Java Basics (page 134 of 168)

5 Error Handling & Exceptions

Exceptions

- "finally" block is always executed, i.e. even if an exception occurs

```
FileWriter fileWriter = null;  
try {  
    fileWriter = new FileWriter("someFileName.txt");  
    fileWriter.write('a');  
    // ...  
} catch (Exception e) {  
    // Exception handling code goes here  
} finally {  
    // Make sure that the file is closed, no matter whether  
    // an exception occurred or not.  
    if (fileWriter != null) {  
        fileWriter.close();  
    }  
}
```

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:21 Introduction to Java Basics (page 133 of 168)

5 Error 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();  
}
```

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:21 Introduction to Java Basics (page 135 of 168)

5 Error 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 {  
        // ...  
    }  
}
```

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:23 Introduction to Java Basics (page 136 of 168)

5 Error Handling & Exceptions

Exceptions

- **Checked Exceptions**
 - Subclasses of `java.lang.Exception`
 - *Must* be caught or forwarded where they possibly occur (try/catch or throws)
 - Example: Opening a file (is likely to go wrong)
- **Unchecked ("Runtime") Exceptions**
 - Subclasses of `java.lang.RuntimeException` or `java.lang.Error`
 - *Can* be caught, i.e. no try/catch or throws necessary
 - Example:

```
static long factorial(int n) {  
    if (n < 0) {  
        throw new RuntimeException("Check your math!");  
    } else if (n == 1) {  
        return 1;  
    }  
    return n * factorial(n-1);  
}
```

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:24 Introduction to Java Basics (page 137 of 168)

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>

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 12:26 Java – Uebung06/src/ExceptionsDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Proj Factorial.java BinarySearch.java *ExceptionsDemo.java

```
1 import java.io.FileReader;
2
3 public class ExceptionsDemo {
4
5     public static void main(String[] args) {
6         try {
7             FileReader fr = new FileReader("MichGibtEsNicht.txt");
8
9         }
10    }
11
12 }
13
```

FileReader cannot be resolved to a type

Problems @ Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
7
8
9
10
1
2
3
```

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 12:27 Java – Uebung06/src/ExceptionsDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Proj Factorial.java BinarySearch.java *ExceptionsDemo.java

```
1 import java.io.FileReader;
2
3 public class ExceptionsDemo {
4
5     public static void main(String[] args) {
6         try {
7             FileReader fr = new FileReader("MichGibtEsNicht.txt");
8
9         }
10    }
11
12 }
13
```

Problems @ Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
7
8
9
10
1
2
3
```

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 12:27 Java – Uebung06/src/ExceptionsDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Proj Factorial.java BinarySearch.java *ExceptionsDemo.java

```
1 import java.io.FileReader;
2
3 public class ExceptionsDemo {
4
5     public static void main(String[] args) {
6         FileReader fr = new FileReader("MichGibtEsNicht.txt");
7     }
8
9
10 }
11
```

Unhandled exception type FileNotFoundException

Problems @ Javadoc Declaration Console

```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
7
8
9
10
1
2
3
```

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 12:29 Java – Uebung06/src/ExceptionsDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ Proj Factorial.java BinarySearch.java ExceptionsDemo.java

```
1 import java.io.FileNotFoundException;
2 import java.io.FileReader;
3
4 public class ExceptionsDemo {
5
6     public static void main(String[] args) {
7         try {
8             FileReader fr = new FileReader("MichGibtEsNicht.txt");
9         } catch (FileNotFoundException e) {
10             System.out.println("Irgendwas");
11         }
12
13         int[] numbers = { 1, 2, 3, 4, 5 };
14         for (int i = 0; i < numbers.length; i++) {
15             System.out.println(numbers[i]);
16         }
17     }
18
19 }
20
```

Problems @ Javadoc Declaration Console

```
<terminated> ExceptionsDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20)
1
2
3
4
5
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
at ExceptionsDemo.main(ExceptionsDemo.java:15)
```

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 12:29

Java - Uebung06/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss_2013/teil_java/workspace

Proj Pac Typ

Factorial.java BinarySearch.java *ExceptionsDemo.java

```

1 import java.io.FileNotFoundException;
2 import java.io.FileReader;
3
4 public class ExceptionsDemo {
5
6     public static void main(String[] args) {
7         try {
8             FileReader fr = new FileReader("MichGibtEsNicht.txt");
9         } catch (FileNotFoundException e) {
10             System.out.println("Irgendwas");
11         }
12
13         int[] numbers = { 1, 2, 3, 4, 5 };
14         for (int i = 0; i < numbers.length; i++) {
15             System.out.println(numbers[i]);
16         }
17     }
18 }
19
20

```

BankAccount BeesAndFlowers BicycleDemo ControlFlowDemo DritteUebung Erathostenes Exceptions Fakultaet Floodfill Histogram ImageDemo InterfaceDemo OverloadAndOverride Polymorphism QuickSort SimpleRecursion src (default package) BinarySearch.java BinarySearch Factorial.java Fibonacci.java JRE System Library [Java SE 7 (MstatementsAndOperators Uebung06 src (default package) ExceptionsDemo.java JRE System Library [JavaSE-1.7]

Problems @ Javadoc Declaration Console

<terminated> ExceptionsDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.0)

1
2
3
4
5
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
at ExceptionsDemo.main(ExceptionsDemo.java:15)

Writable Smart Insert 14 : 28

Eclipse File Edit Source Refactor Navigate Project Run Window Help Fr. 12.Jul 12:29

Java - Uebung06/src/ExceptionsDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss_2013/teil_java/workspace

Proj Pac Typ

Factorial.java BinarySearch.java ExceptionsDemo.java

```

1 import java.io.FileNotFoundException;
2 import java.io.FileReader;
3
4 public class ExceptionsDemo {
5
6     public static void main(String[] args) {
7         try {
8             FileReader fr = new FileReader("MichGibtEsNicht.txt");
9         } catch (FileNotFoundException e) {
10             System.out.println("Irgendwas");
11         }
12
13         int[] numbers = { 1, 2, 3, 4, 5 };
14         for (int i = 0; i < numbers.length; i++) {
15             System.out.println(numbers[i]);
16         }
17     }
18 }
19
20

```

BankAccount BeesAndFlowers BicycleDemo ControlFlowDemo DritteUebung Erathostenes Exceptions Fakultaet Floodfill Histogram ImageDemo InterfaceDemo OverloadAndOverride Polymorphism QuickSort SimpleRecursion src (default package) BinarySearch.java BinarySearch Factorial.java Fibonacci.java JRE System Library [Java SE 7 (MstatementsAndOperators Uebung06 src (default package) ExceptionsDemo.java JRE System Library [JavaSE-1.7]

Problems @ Javadoc Declaration Console

<terminated> ExceptionsDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.0)

Irgendwas
1
2
3
4
5

Writable Smart Insert 14 : 28

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:30

Introduction to Java Basics (page 137 of 168)

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>

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:30

Introduction to Java Basics (page 140 of 168)

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!



Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:32

Introduction to Java Basics (page 146 of 168)

7 Using the Java Class Library

The screenshot shows the Java API documentation for Java Platform SE 6. The main title is "Java™ Platform Standard Ed. 6 API Specification". Under the "Packages" section, the "java.awt" package is selected. The table lists various classes and their descriptions:

| Package | Description |
|---|--|
| java.awt | Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context. |
| java.awt.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.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.image.renderable | Provides classes and interfaces for producing rendering-independent |

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:33

Introduction to Java Basics (page 147 of 168)

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
```

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:34

Introduction to Java Basics (page 148 of 168)

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);
```

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:38

Introduction to Java Basics (page 148 of 168)

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);
```

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:39 Introduction to Java Basics (page 149 of 168)

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;
```

Preview File Edit View Go Tools Bookmarks Window Help Fr. 12.Jul 12:40 Introduction to Java Basics (page 150 of 168)

8 Solving a Problem

Deepening readings (optional):
[The Internet](#)

Main reference:

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