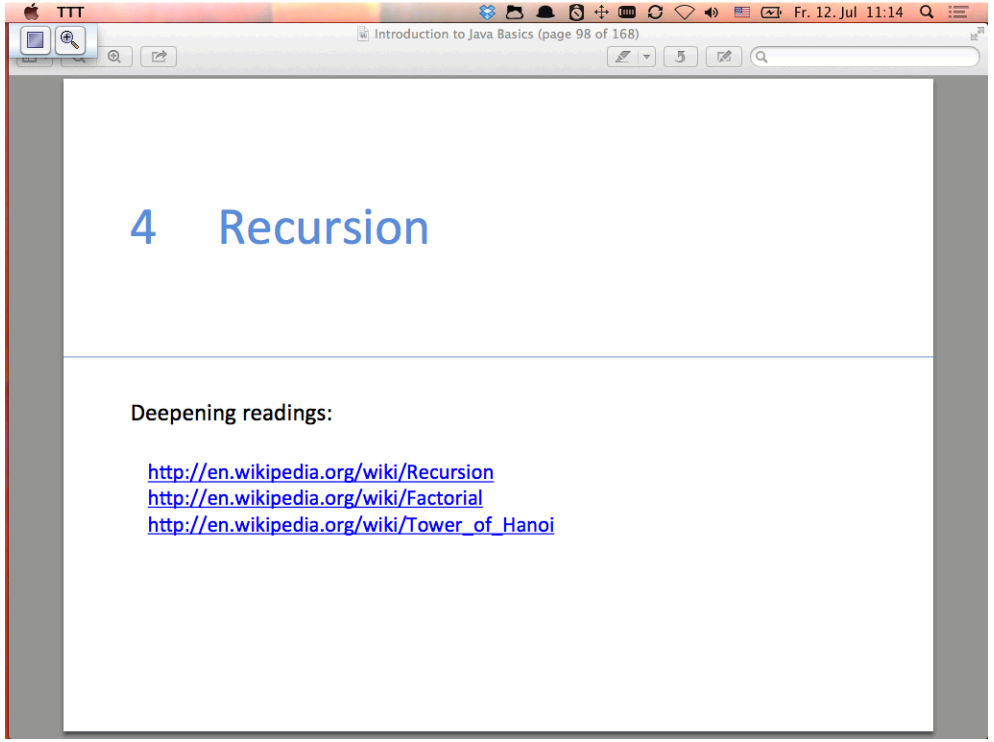
Script generated by TTT

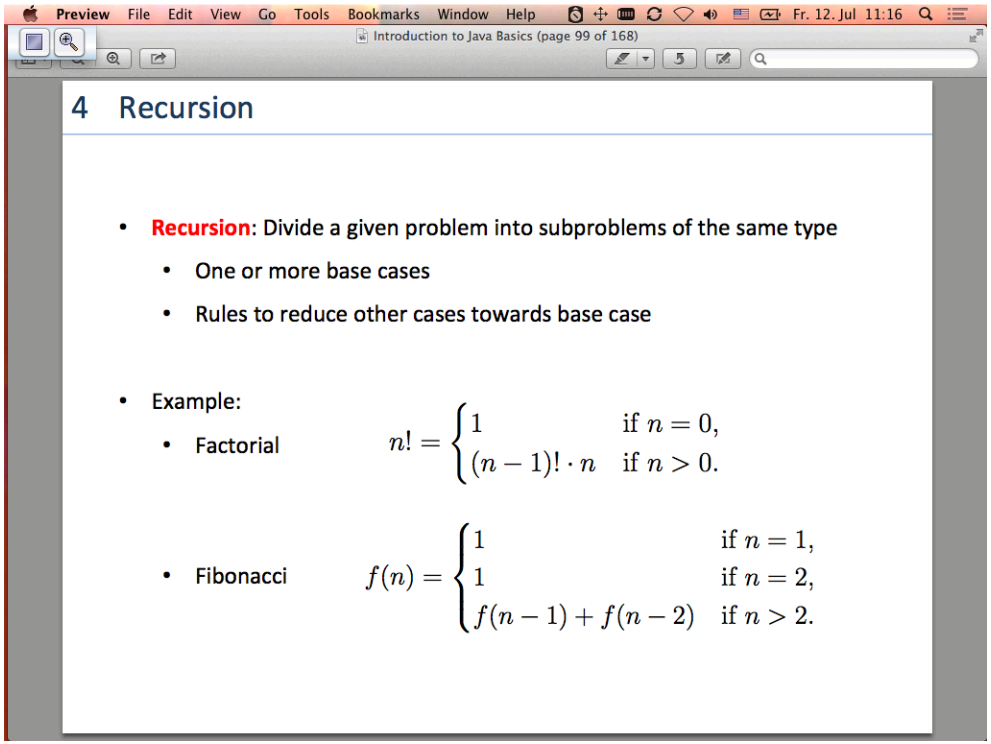
Title: Lehmann: Uebung_Einf_HF (12.07.2013)

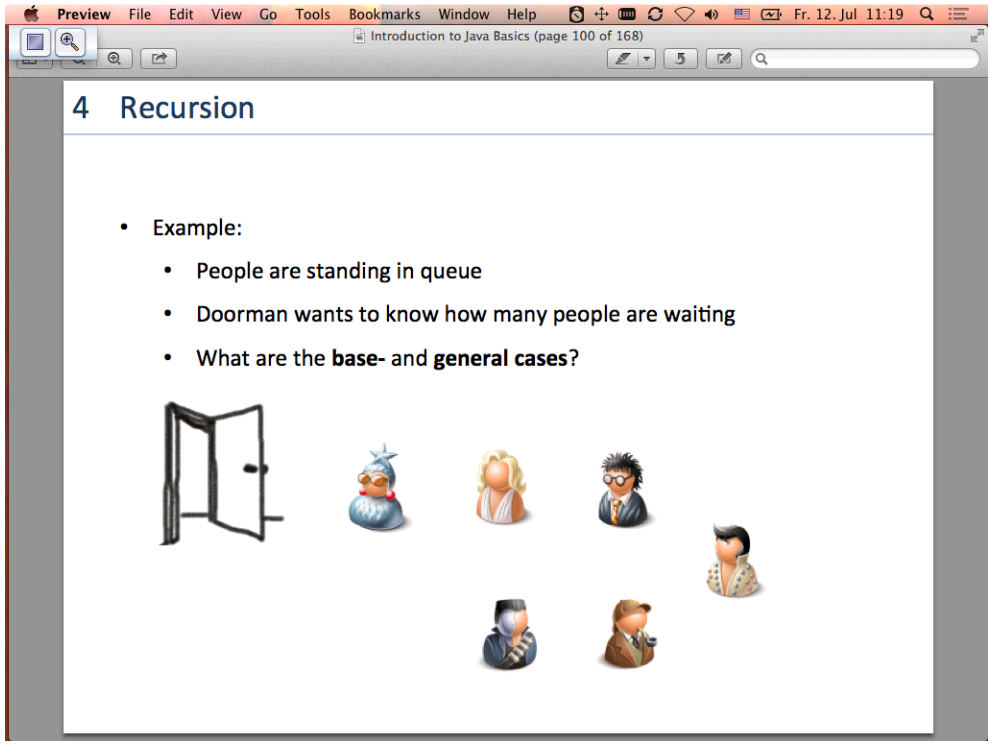
Date: Fri Jul 12 11:14:59 CEST 2013

Duration: 86:03 min

Pages: 89





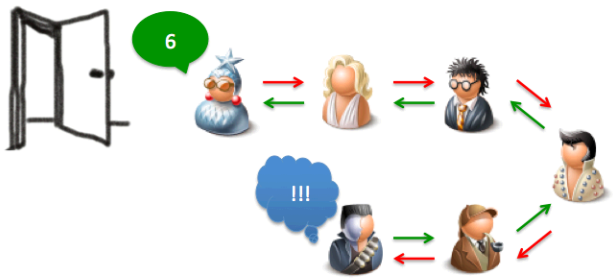


Preview File Edit View Go Tools Bookmarks Window Help Fr. 12. Jul 11:20

Introduction to Java Basics (page 102 of 168)

4 Recursion

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



Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:20

Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ

BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators

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 }
28
```

Problems Javadoc Declaration Console

No consoles to display at this time.

Writable Smart Insert 18 : 3

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:21

Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ

BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators

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 }
28
```

Problems Javadoc Declaration Console

No consoles to display at this time.

Writable Smart Insert 24 : 55

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:21

Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

Proj Pac Typ

BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators

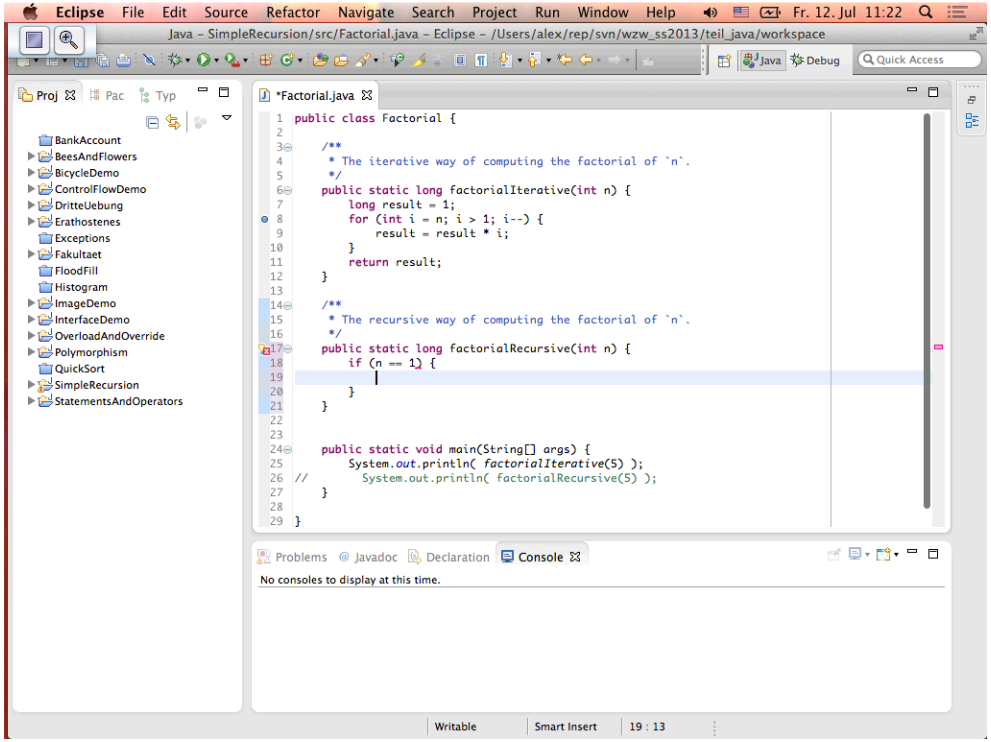
*Factorial.java

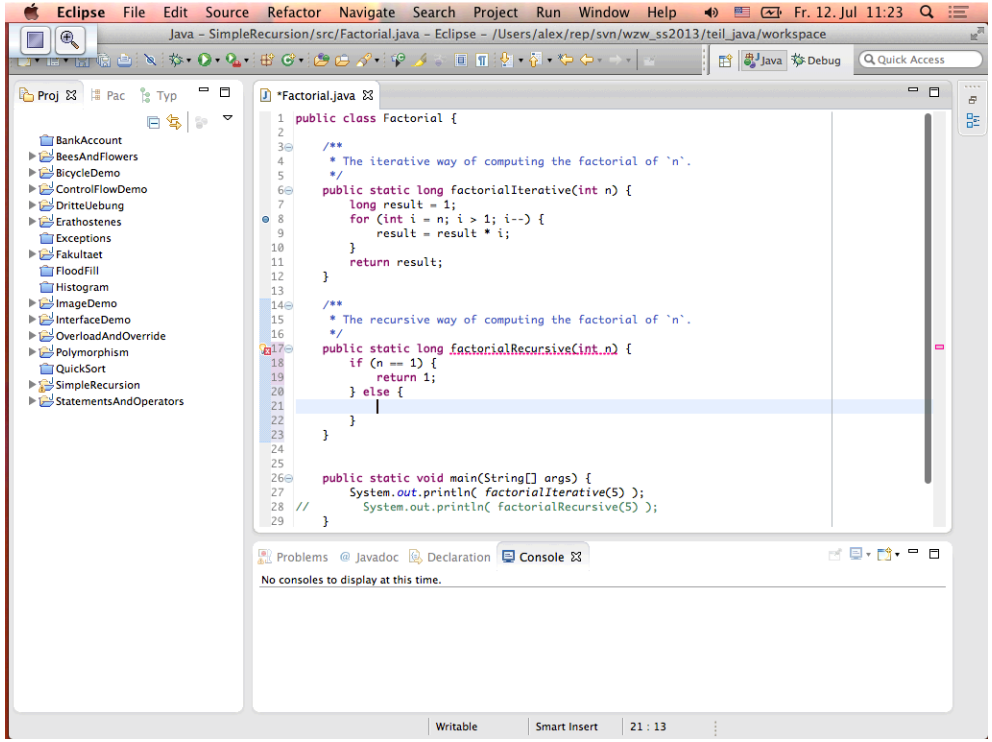
```
1  public class Factorial {
2
3      /**
4       * The iterative way of computing the factorial of `n`.
5       */
6      public static long factorialIterative(int n) {
7          long result = 1;
8          for (int i = n; i > 1; i--) {
9              result = result * i;
10         }
11         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 }
28
```

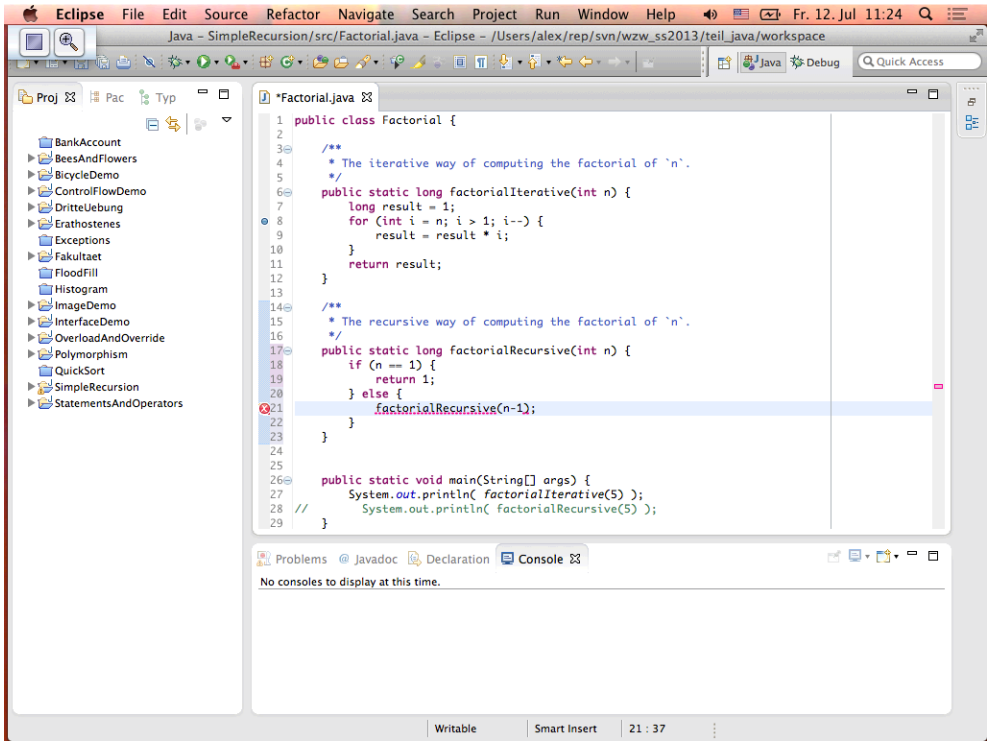
Problems Javadoc Declaration Console

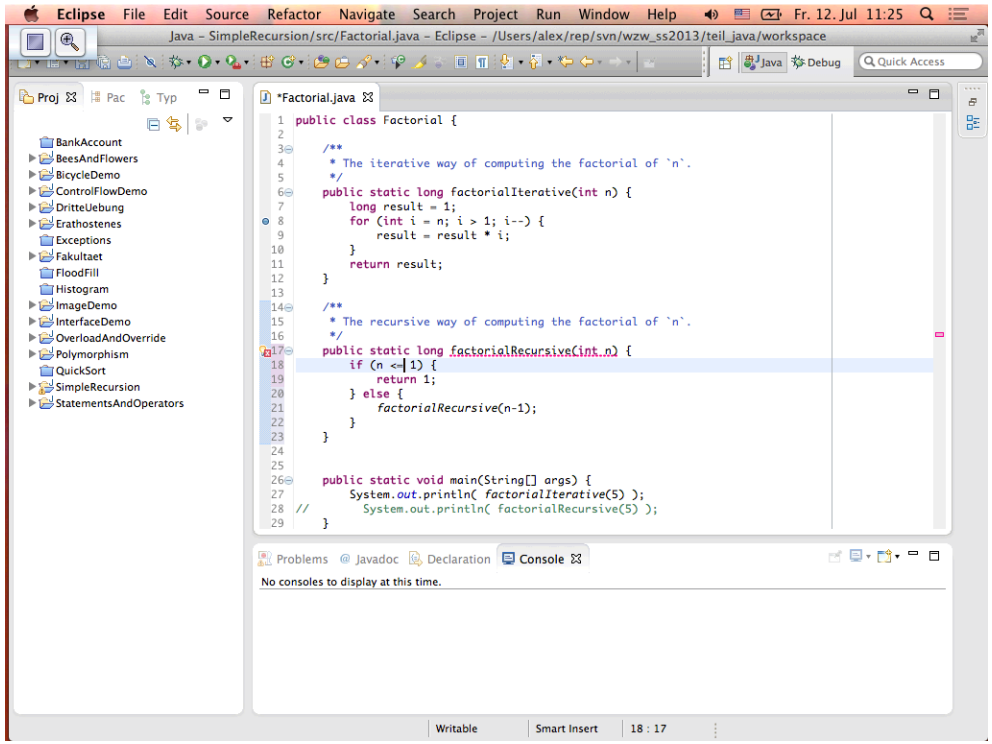
No consoles to display at this time.

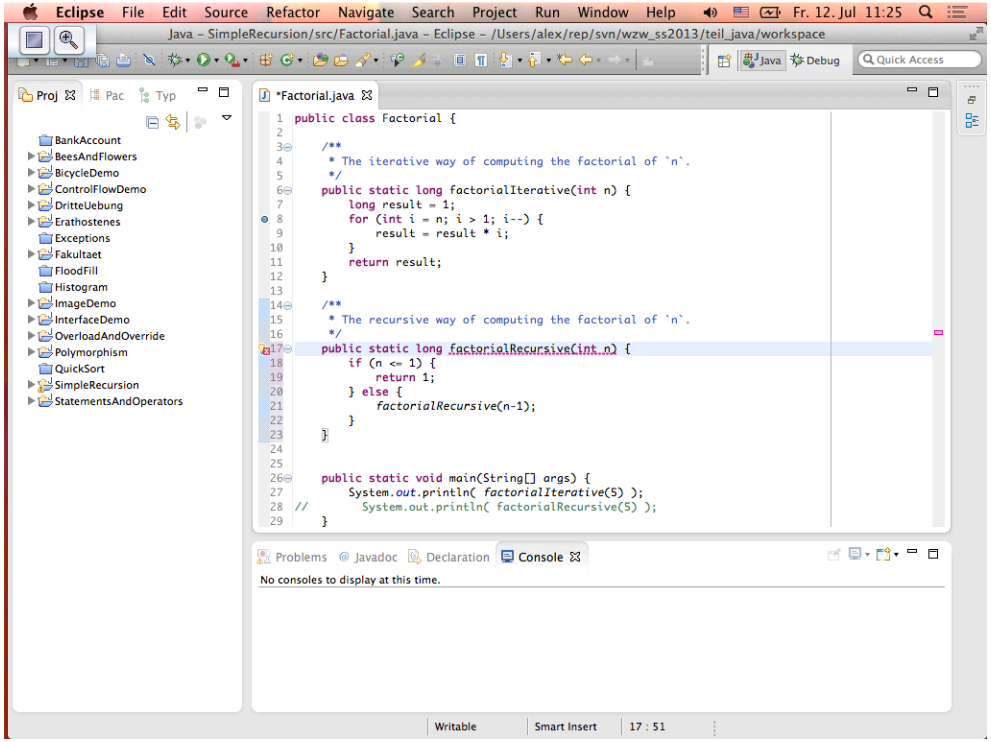
Writable Smart Insert 18 : 9

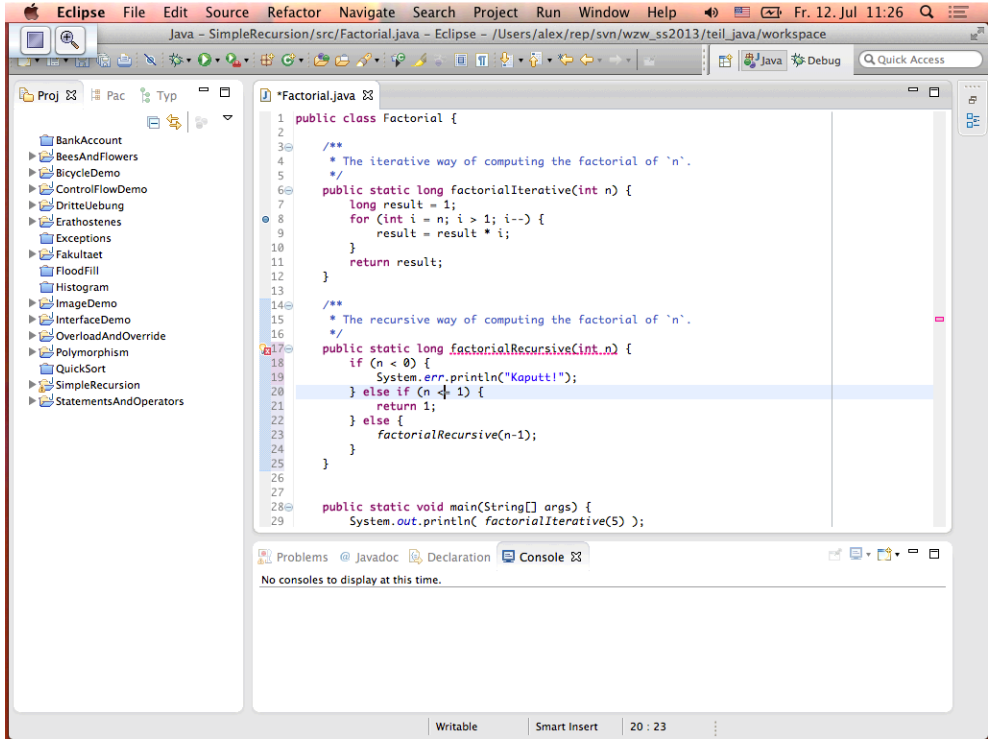

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:22
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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 == 1) {
}
}
public static void main(String[] args) {
System.out.println(factorialIterative(5));
// System.out.println(factorialRecursive(5));
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 19 : 13

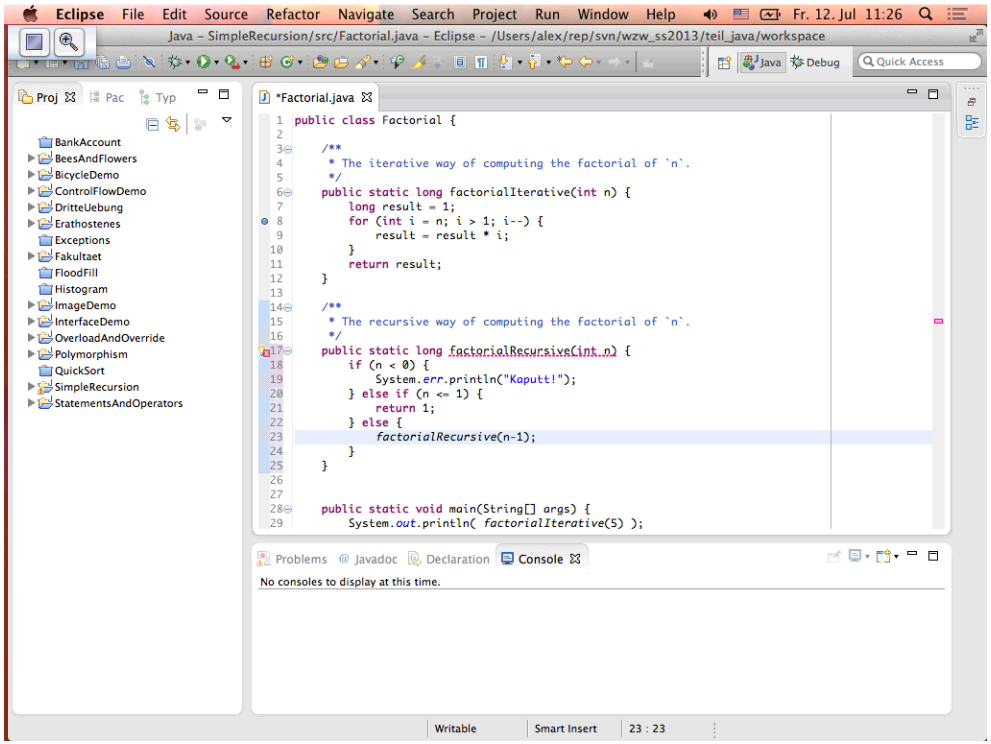

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:23
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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 == 1) {
return 1;
} else {
}
}
public static void main(String[] args) {
System.out.println(factorialIterative(5));
// System.out.println(factorialRecursive(5));
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 21 : 13

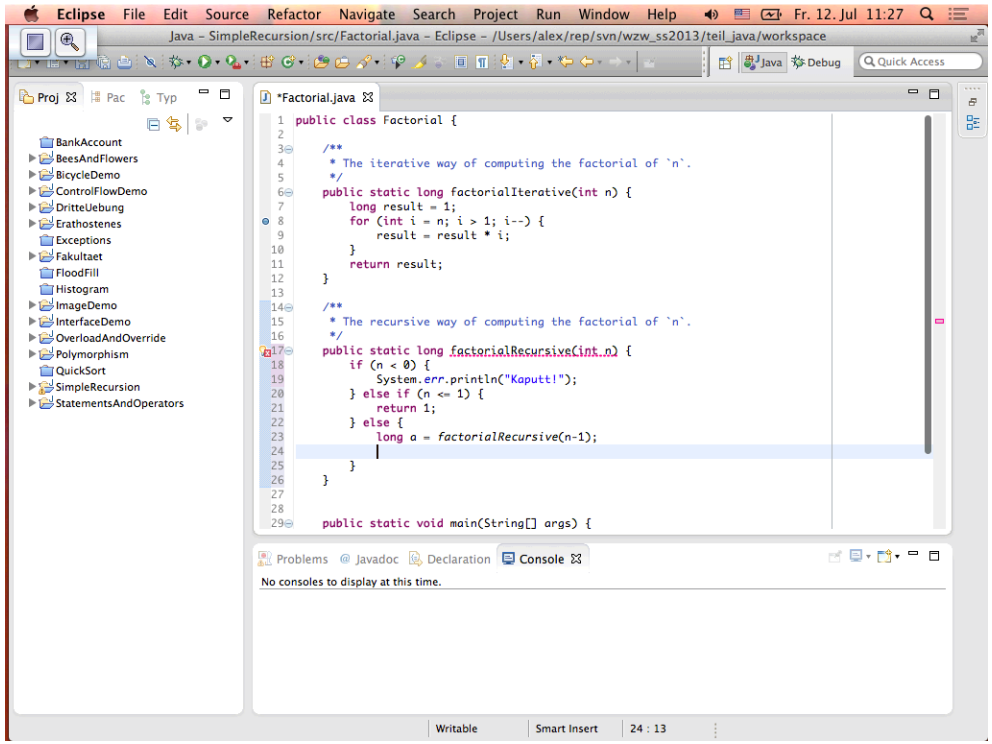

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:24
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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 == 1) {
return 1;
} else {
factorialRecursive(n-1);
}
}
public static void main(String[] args) {
System.out.println(factorialIterative(5));
// System.out.println(factorialRecursive(5));
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 21 : 37

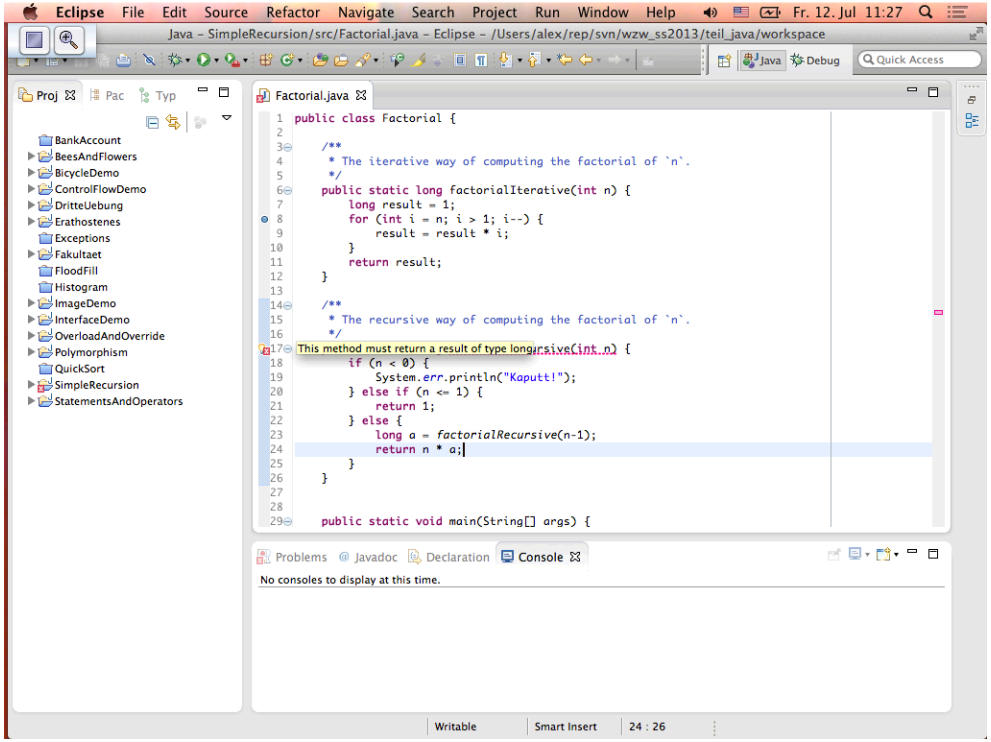

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:25
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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 <= 1) {
return 1;
} else {
factorialRecursive(n-1);
}
}
public static void main(String[] args) {
System.out.println(factorialIterative(5));
// System.out.println(factorialRecursive(5));
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 18 : 17

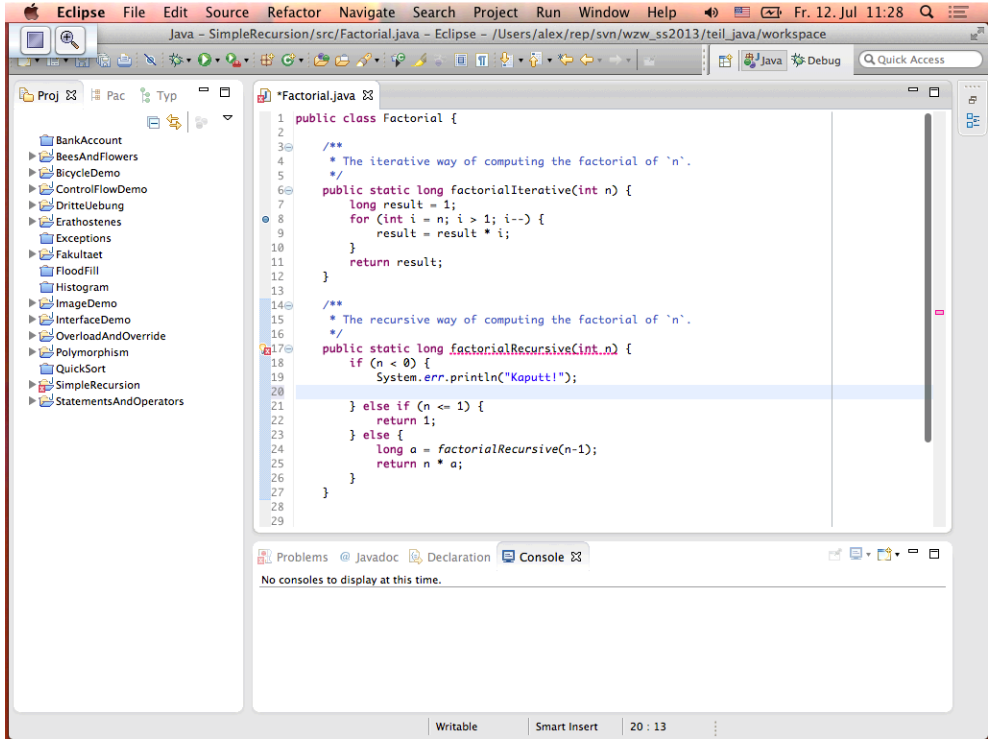

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:25
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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 <= 1) {
return 1;
} else {
factorialRecursive(n-1);
}
}
public static void main(String[] args) {
System.out.println(factorialIterative(5));
// System.out.println(factorialRecursive(5));
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 17 : 51

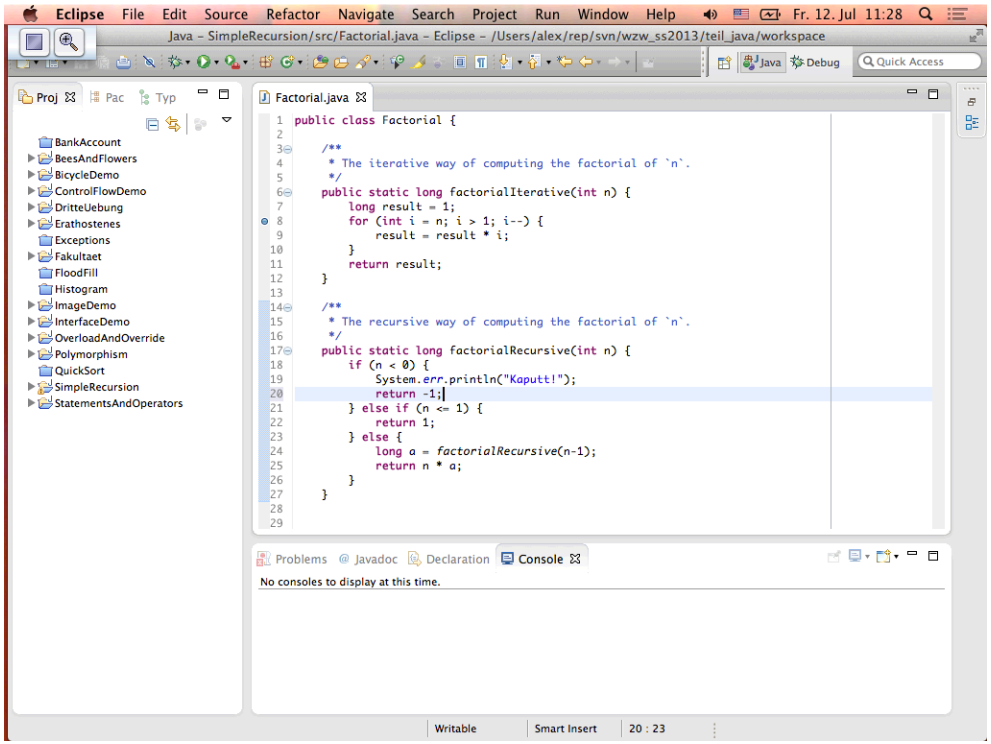

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:26
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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!");
} else if (n <= 1) {
return 1;
} else {
factorialRecursive(n-1);
}
}
public static void main(String[] args) {
System.out.println(factorialIterative(5));
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 20 : 23

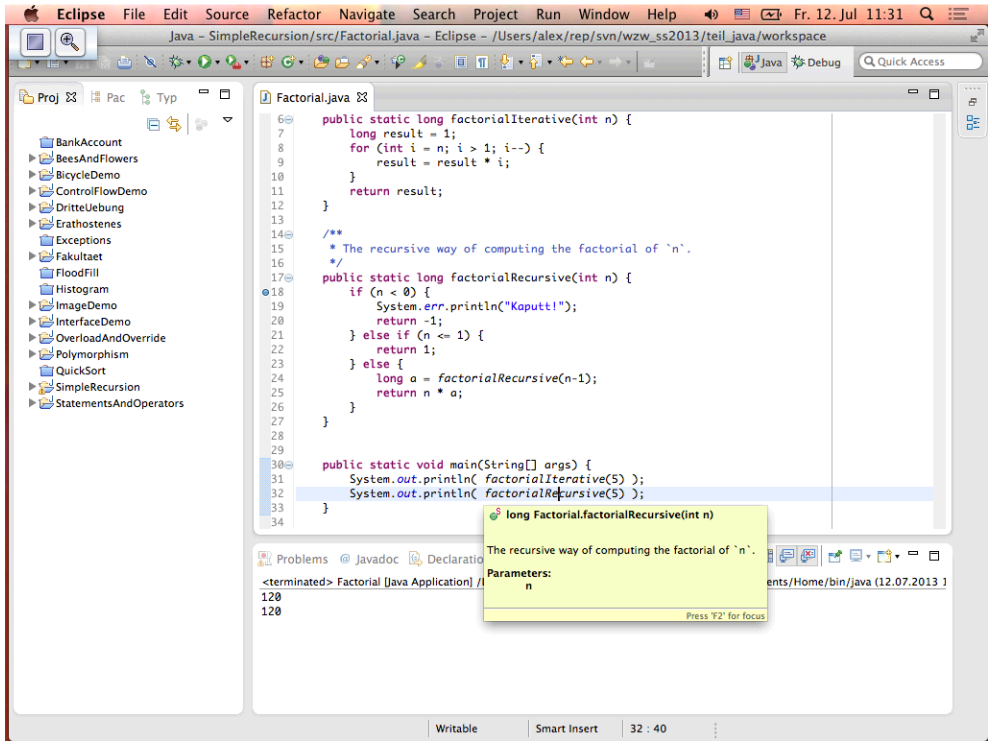

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:26
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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!");
} else if (n <= 1) {
return 1;
} else {
factorialRecursive(n-1);
}
}
public static void main(String[] args) {
System.out.println(factorialIterative(5));
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 23 : 23

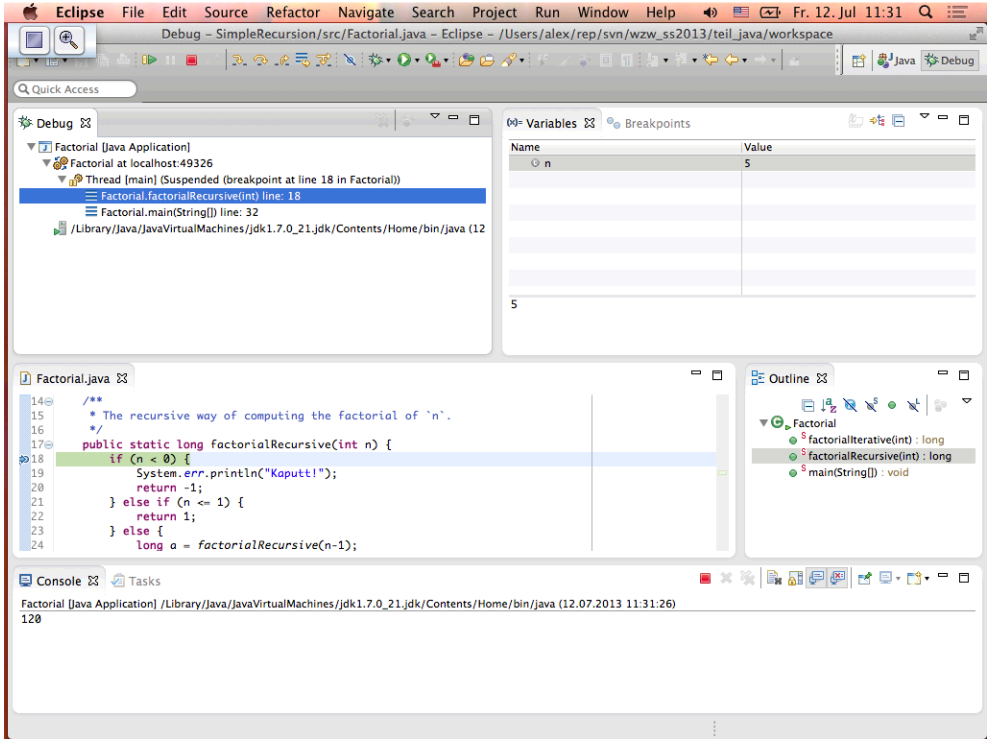

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:27
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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!");
} else if (n <= 1) {
return 1;
} else {
long a = factorialRecursive(n-1);
}
}
public static void main(String[] args) {
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 24 : 13

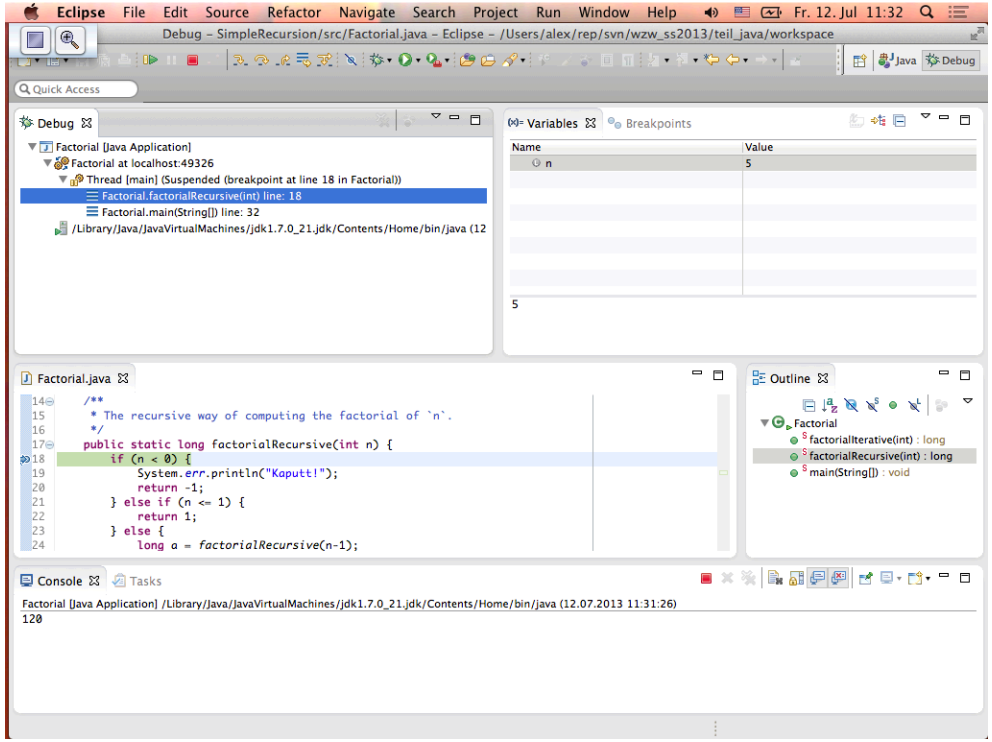

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:27
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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`.
*/
This method must return a result of type long
if (n < 0) {
System.err.println("Kaputt!");
} else if (n <= 1) {
return 1;
} else {
long a = factorialRecursive(n-1);
return n * a;
}
}
public static void main(String[] args) {
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 24 : 26

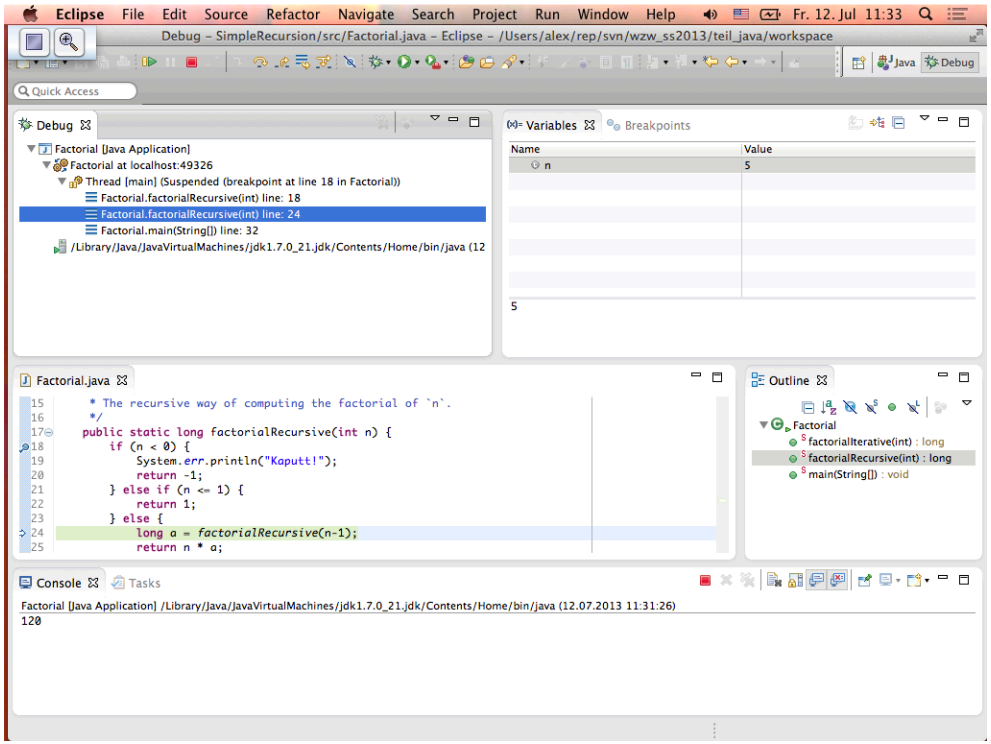

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:28
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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!");
} else if (n <= 1) {
return 1;
} else {
long a = factorialRecursive(n-1);
return n * a;
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 20 : 13

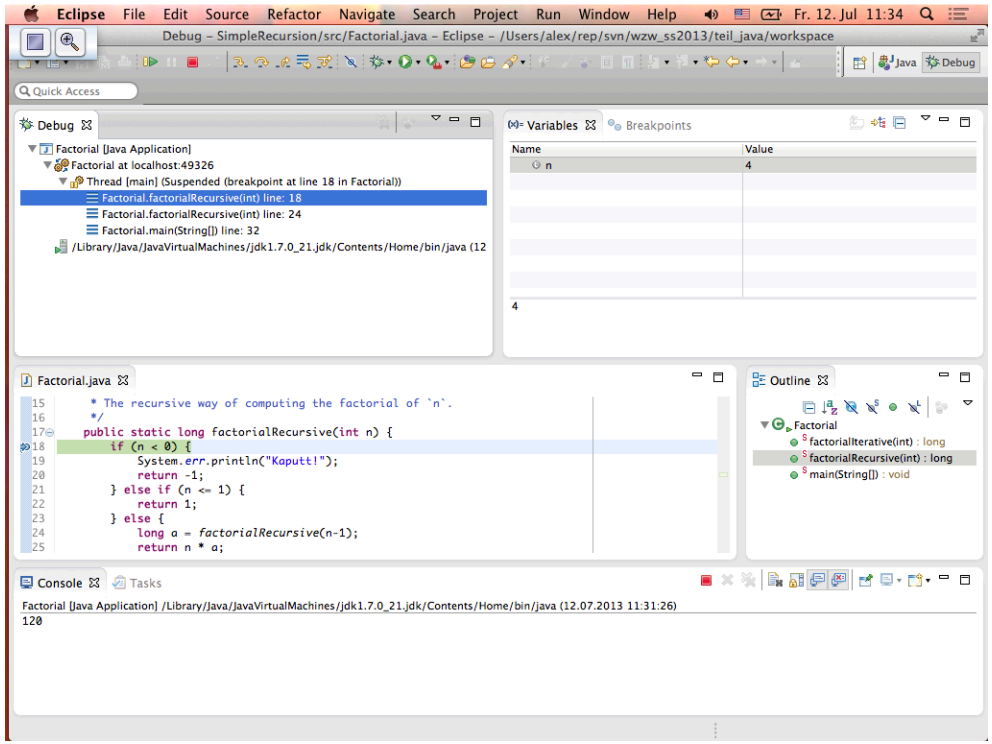

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:28
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
public class Factorial {
/**
* The iterative way of computing the factorial of `n`.
*/
public 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;
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 20 : 23

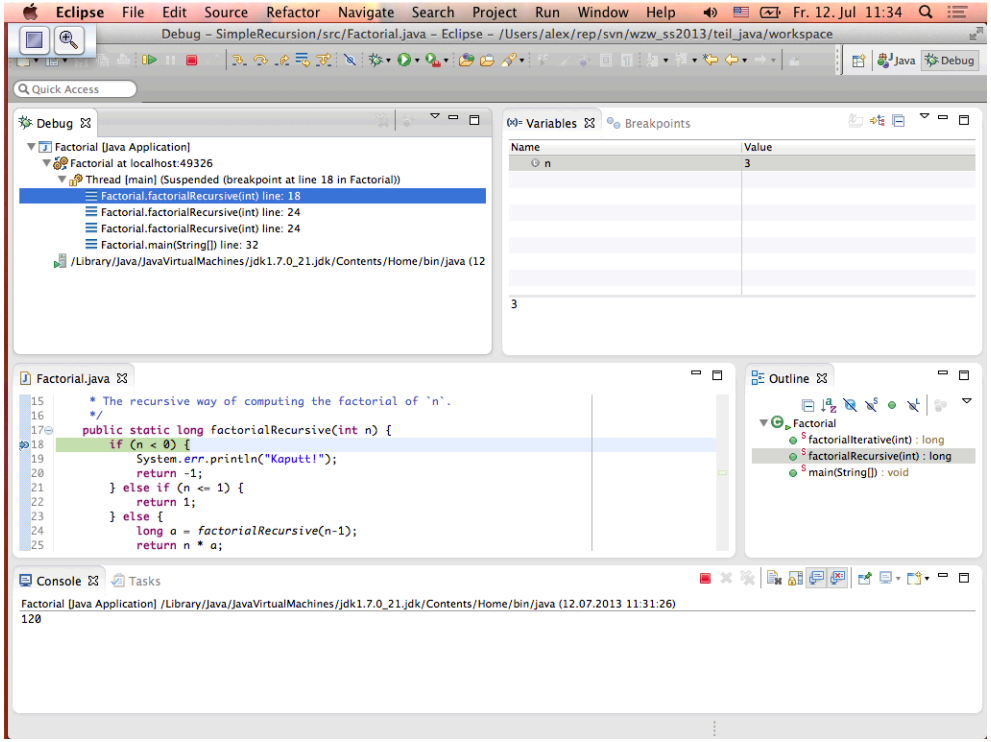

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:31
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
public static long factorialIterative(int n) {
long result = 1;
for (int i = n; i > 1; i--) {
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;
}
}
public static void main(String[] args) {
System.out.println(factorialIterative(5));
System.out.println(factorialRecursive(5));
}
long Factorial.factorialRecursive(int n)
The recursive way of computing the factorial of `n`.
Parameters:
n
Press 'F2' for focus
Problems Javadoc Declaration
<terminated> Factorial [Java Application]
ents/Home/bin/java (12.07.2013
120
120
Writable Smart Insert 32 : 40

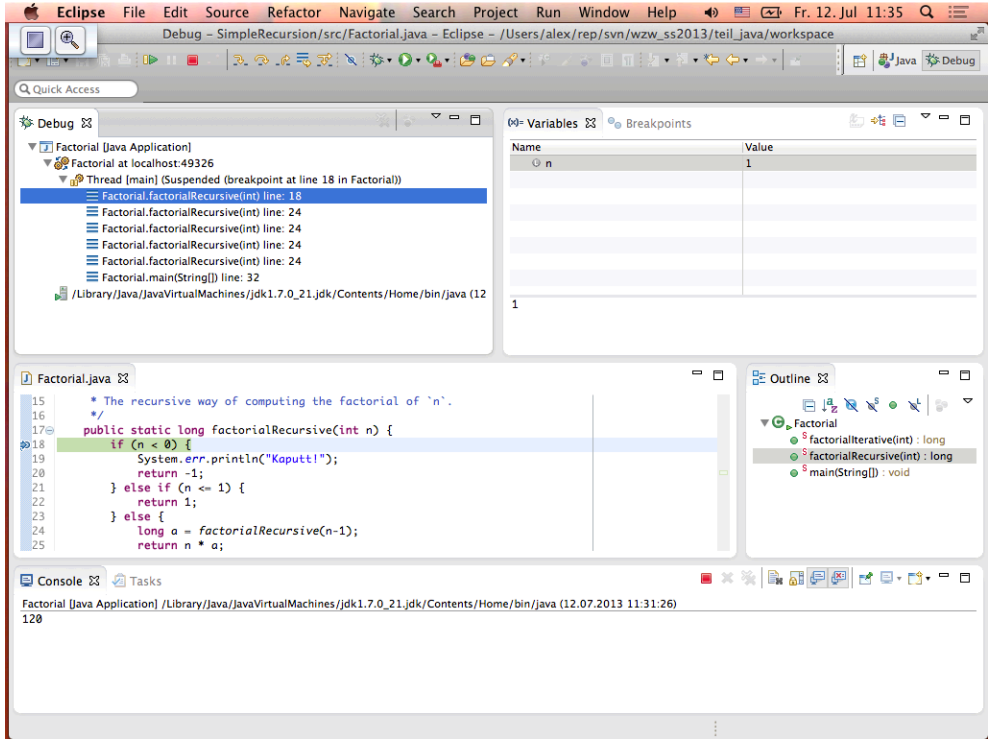

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:31
Debug – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
Factorial [Java Application]
Factorial at localhost:49326
Thread [main] (Suspended (breakpoint at 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
Breakpoints
Name
Value
n
5
Factorial.java
/**
* 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);
Outline
Factorial
factorialIterative(int) : long
factorialRecursive(int) : long
main(String[]) : void
Console
Tasks
Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)
120

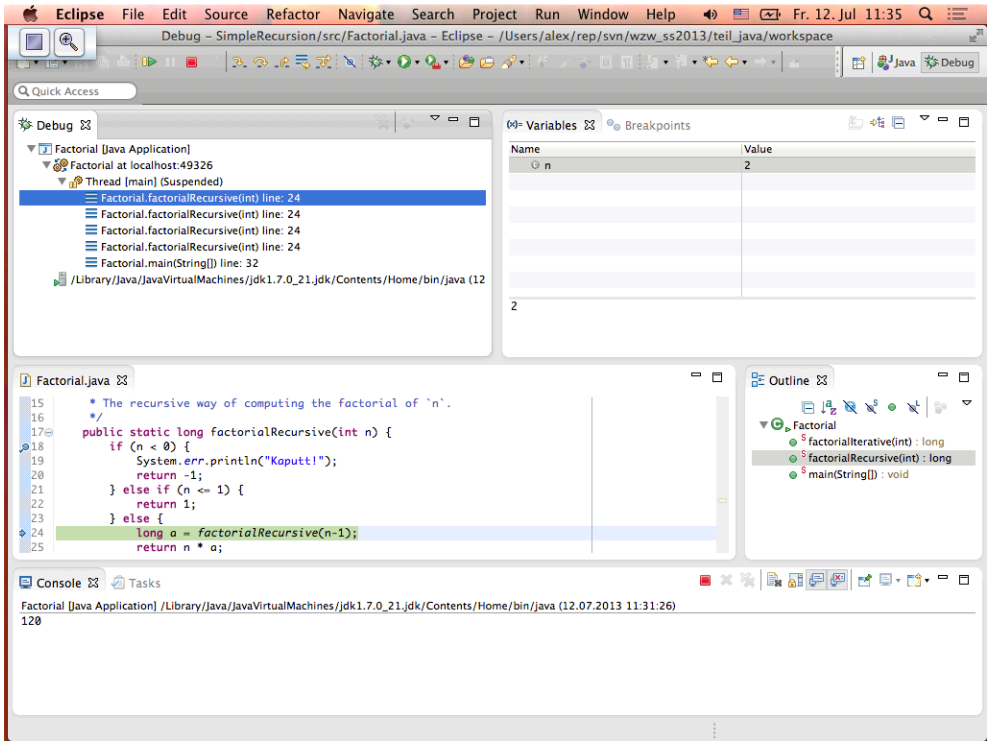

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:32
Debug – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
Factorial [Java Application]
Factorial at localhost:49326
Thread [main] (Suspended (breakpoint at 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
Breakpoints
Name
Value
n
5
Factorial.java
/**
* 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);
Outline
Factorial
factorialIterative(int) : long
factorialRecursive(int) : long
main(String[]) : void
Console
Tasks
Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)
120

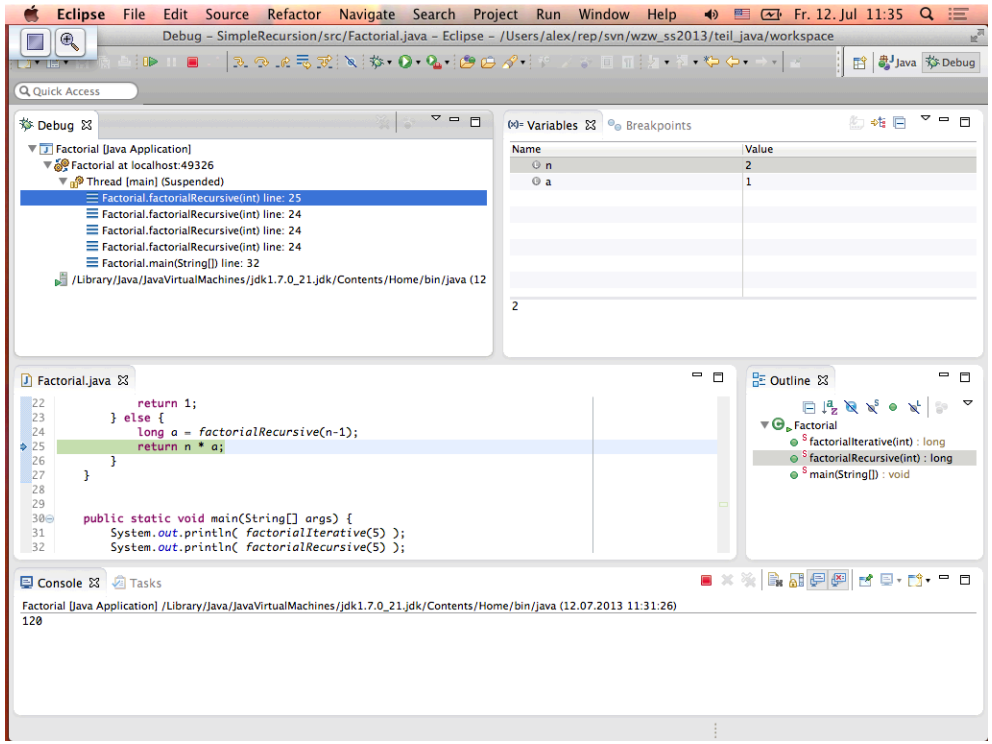

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:33
Debug – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
Factorial [Java Application]
Factorial at localhost:49326
Thread [main] (Suspended (breakpoint at line 18 in Factorial))
Factorial.factorialRecursive(int) line: 18
Factorial.factorialRecursive(int) line: 24
Factorial.main(String[]) line: 32
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12
Variables
Breakpoints
Name
Value
n
5
Factorial.java
* 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;
Outline
Factorial
factorialIterative(int) : long
factorialRecursive(int) : long
main(String[]) : void
Console
Tasks
Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)
120

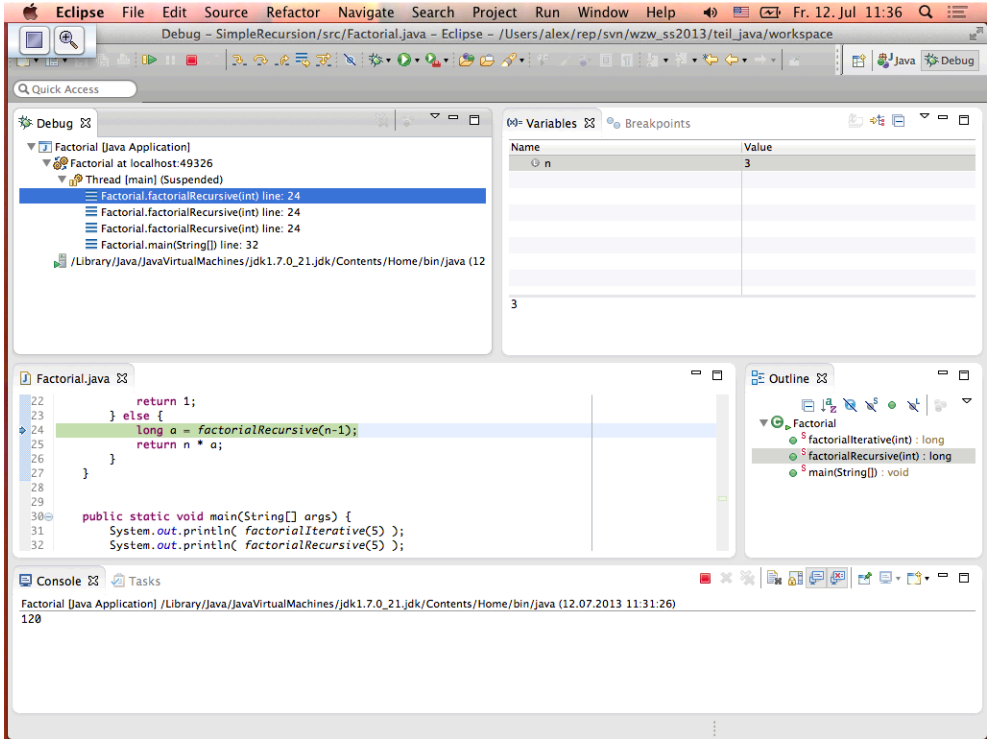

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:34
Debug – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
Factorial [Java Application]
Factorial at localhost:49326
Thread [main] (Suspended (breakpoint at line 18 in Factorial))
Factorial.factorialRecursive(int) line: 18
Factorial.factorialRecursive(int) line: 24
Factorial.main(String[]) line: 32
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12
Variables
Breakpoints
Name
Value
n
4
Factorial.java
* 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;
Outline
Factorial
factorialIterative(int) : long
factorialRecursive(int) : long
main(String[]) : void
Console
Tasks
Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)
120

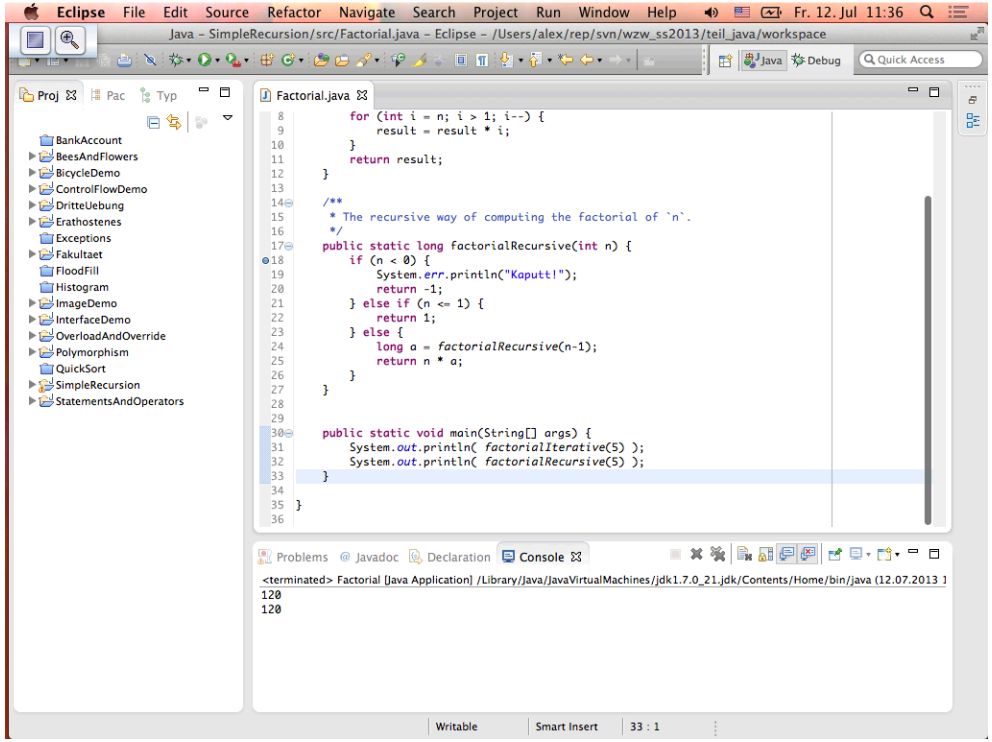

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:34
Debug – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
Factorial [Java Application]
Factorial at localhost:49326
Thread [main] (Suspended (breakpoint at 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
Breakpoints
Name
Value
n
3
3
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.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;
Outline
Factorial
factorialIterative(int) : long
factorialRecursive(int) : long
main(String[]) : void
Console
Tasks
Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)
120

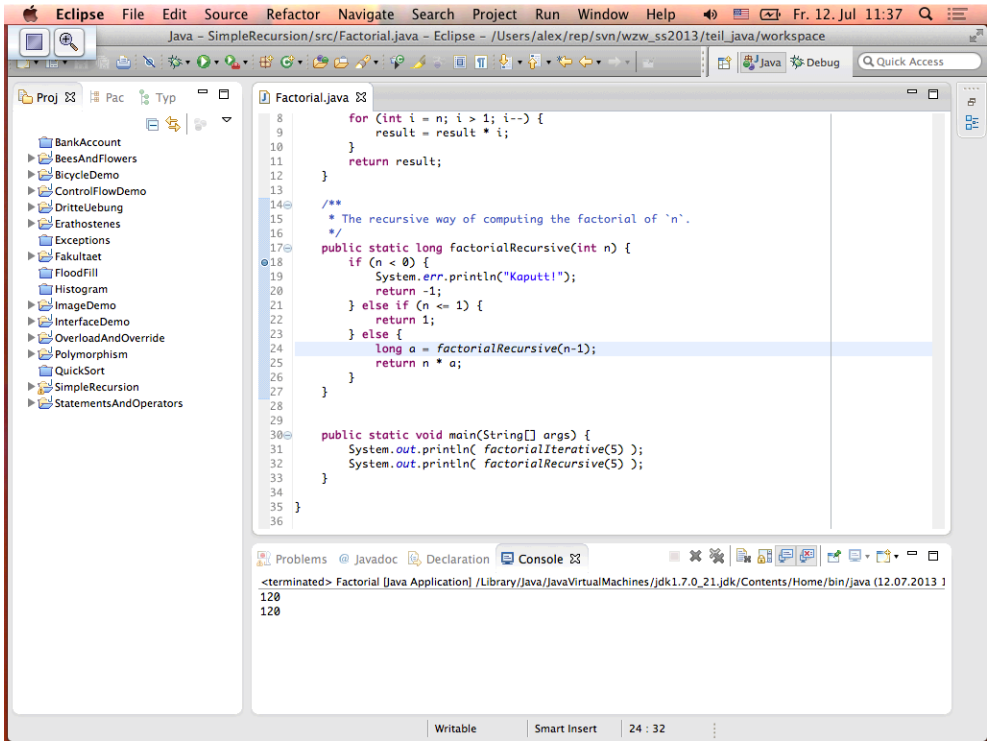

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:35
Debug – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
Factorial [Java Application]
Factorial at localhost:49326
Thread [main] (Suspended (breakpoint at 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
Breakpoints
Name
Value
n
1
1
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.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;
Outline
Factorial
factorialIterative(int) : long
factorialRecursive(int) : long
main(String[]) : void
Console
Tasks
Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)
120

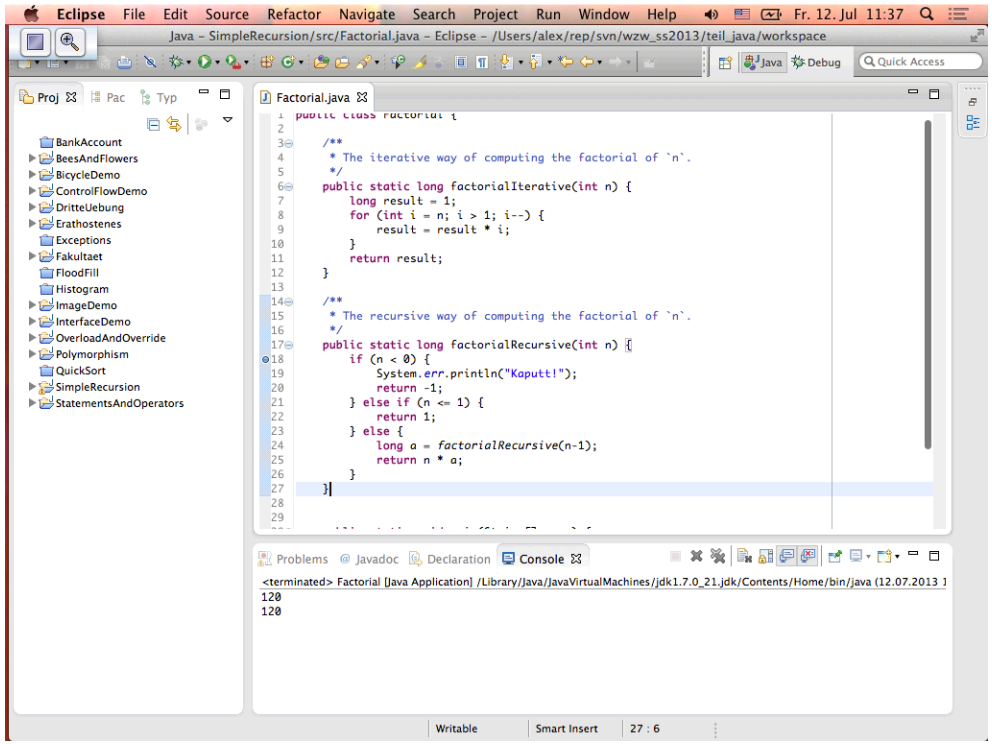

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:35
Debug – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
Factorial [Java Application]
Factorial at localhost:49326
Thread [main] (Suspended)
Factorial.factorialRecursive(int) line: 24
Factorial.factorialRecursive(int) line: 24
Factorial.factorialRecursive(int) line: 24
Factorial.factorialRecursive(int) line: 24
Factorial.main(String[]) line: 32
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12
Variables
Breakpoints
Name
Value
n
2
2
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.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;
Outline
Factorial
factorialIterative(int) : long
factorialRecursive(int) : long
main(String[]) : void
Console
Tasks
Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)
120

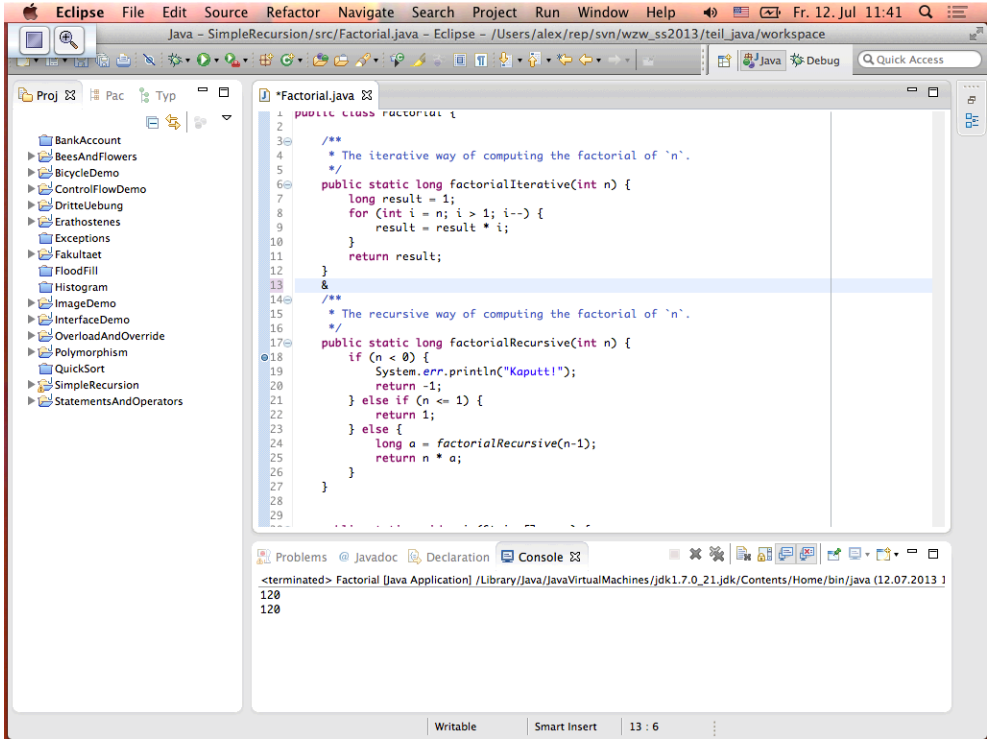

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:35
Debug – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
Factorial [Java Application]
Factorial at localhost:49326
Thread [main] (Suspended)
Factorial.factorialRecursive(int) 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
Breakpoints
Name
Value
n
2
a
1
2
Factorial.java
22 return 1;
23 } else {
24 long a = factorialRecursive(n-1);
25 return n * a;
26 }
27 }
28
29
30 public static void main(String[] args) {
31 System.out.println(factorialIterative(5));
32 System.out.println(factorialRecursive(5));
Outline
Factorial
factorialIterative(int) : long
factorialRecursive(int) : long
main(String[]) : void
Console
Tasks
Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)
120

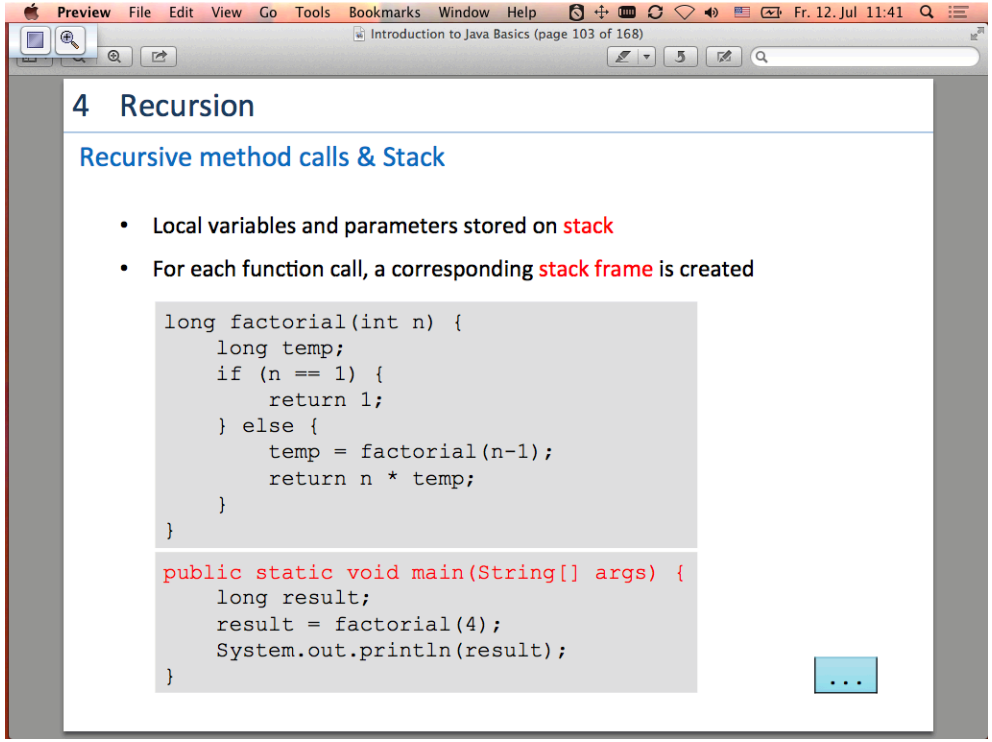

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:36
Debug – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
Factorial [Java Application]
Factorial at localhost:49326
Thread [main] (Suspended)
Factorial.factorialRecursive(int) 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
Variables
Breakpoints
Name
Value
n
3
Factorial.java
return 1;
} else {
long a = factorialRecursive(n-1);
return n * a;
}
}
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
Tasks
Factorial [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.2013 11:31:26)
120

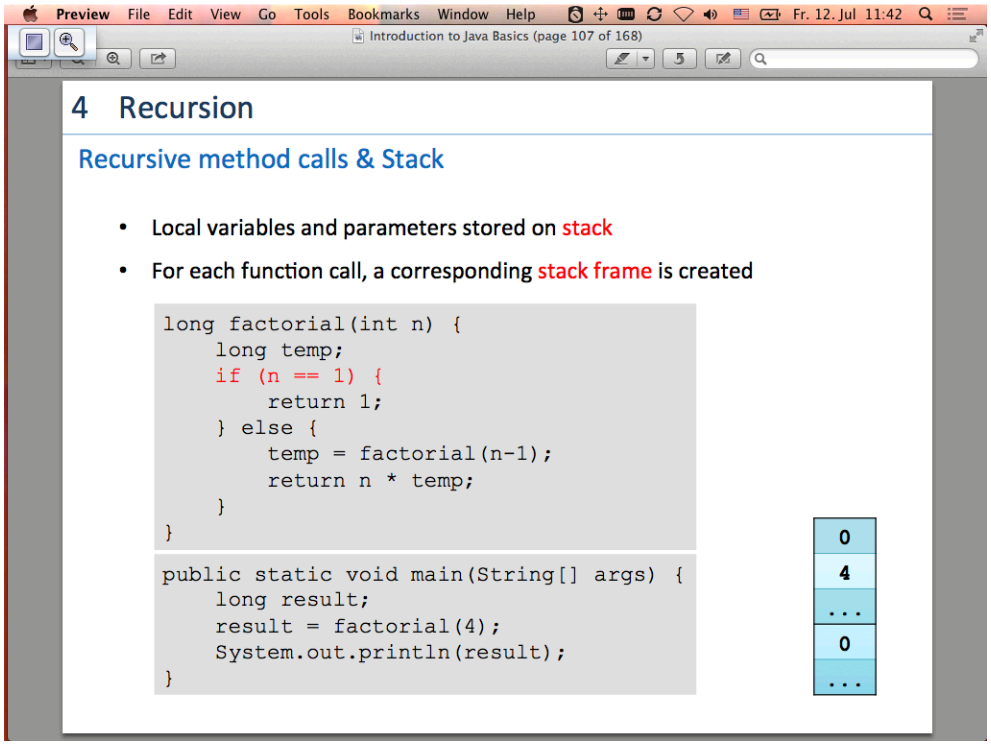

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:36
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Proj
Pac
Typ
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
Factorial.java
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;
}
}
public static void main(String[] args) {
System.out.println(factorialIterative(5));
System.out.println(factorialRecursive(5));
}
}
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
Writable
Smart Insert
33 : 1

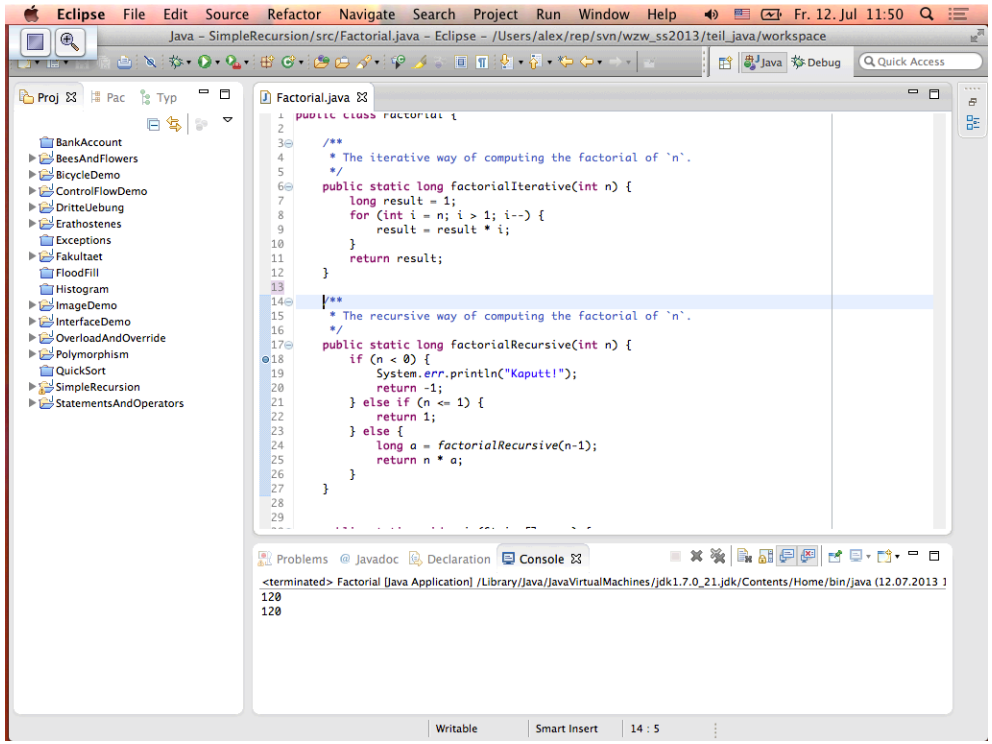

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:37
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Proj
Pac
Typ
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
Factorial.java
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;
}
}
public static void main(String[] args) {
System.out.println(factorialIterative(5));
System.out.println(factorialRecursive(5));
}
}
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
Writable
Smart Insert
24 : 32

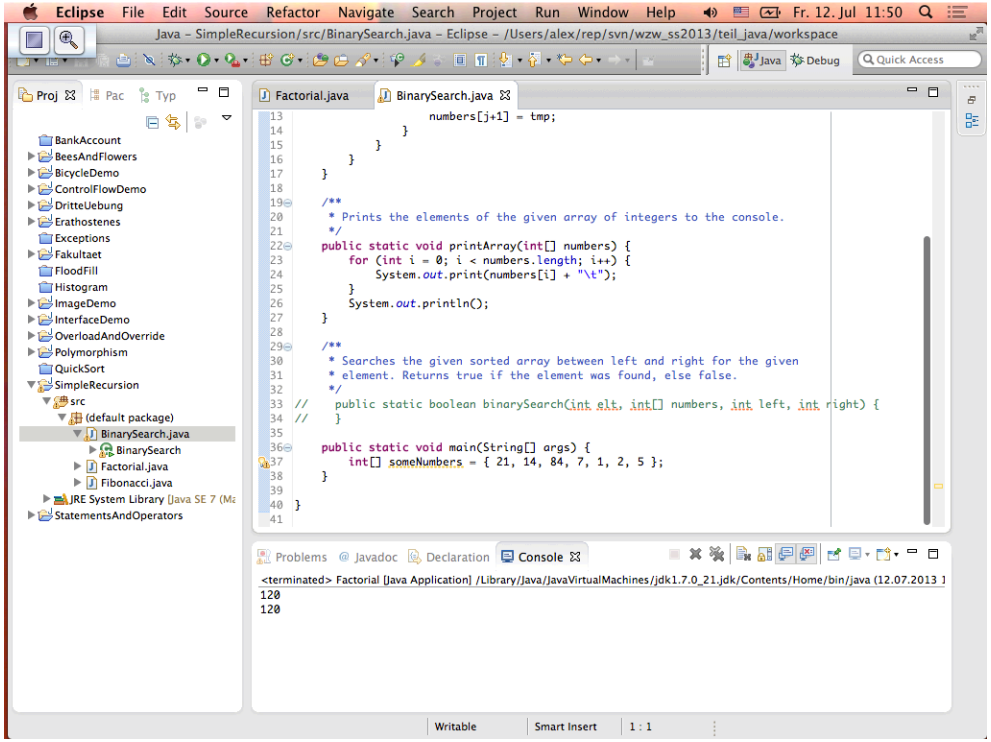

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:37
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Proj
Pac
Typ
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
Factorial.java
/**
* 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;
}
}
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
Writable
Smart Insert
27 : 6

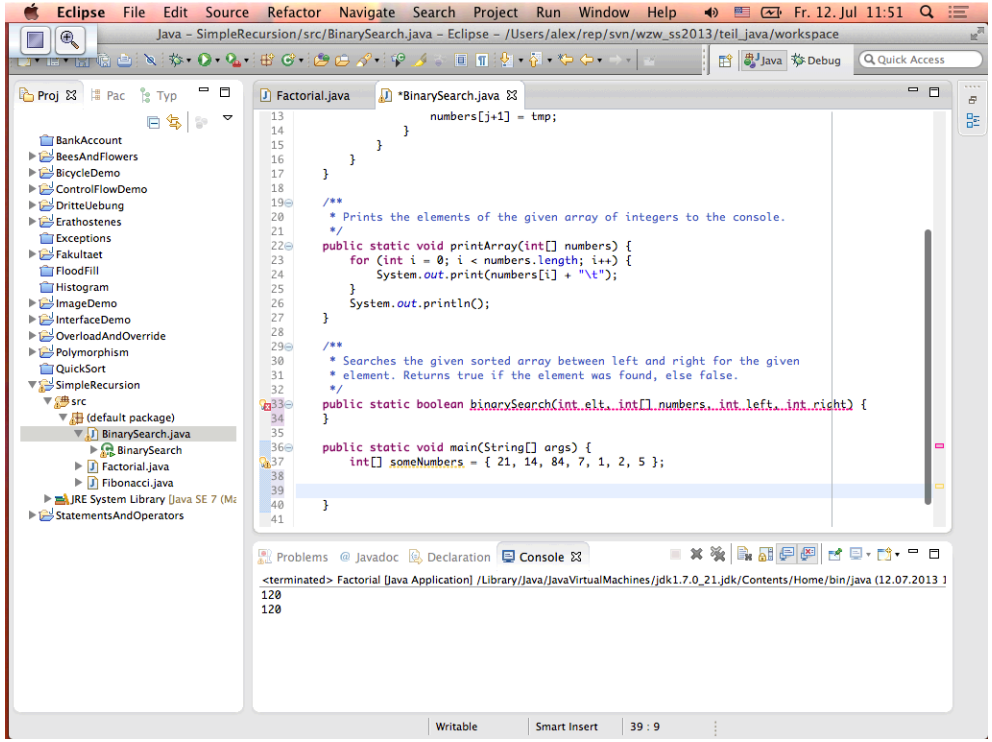

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 12. Jul 11:41
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Quick Access
Java
Debug
Proj
Pac
Typ
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
*Factorial.java
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;
}
}
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
Writable
Smart Insert
13 : 6

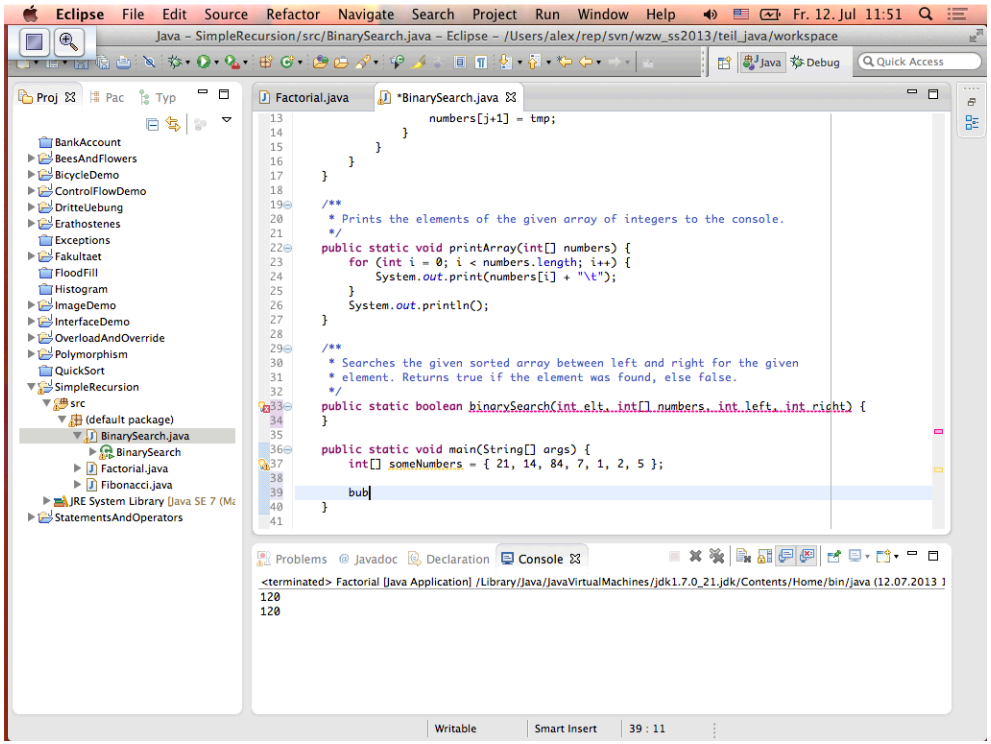

Preview File Edit View Go Tools Bookmarks Window Help
Fr. 12. Jul 11:41
Introduction to Java Basics (page 103 of 168)
4 Recursion
Recursive method calls & Stack
Local variables and parameters stored on stack
For each function call, a corresponding stack frame is created
long factorial(int n) {
long temp;
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);
}
...

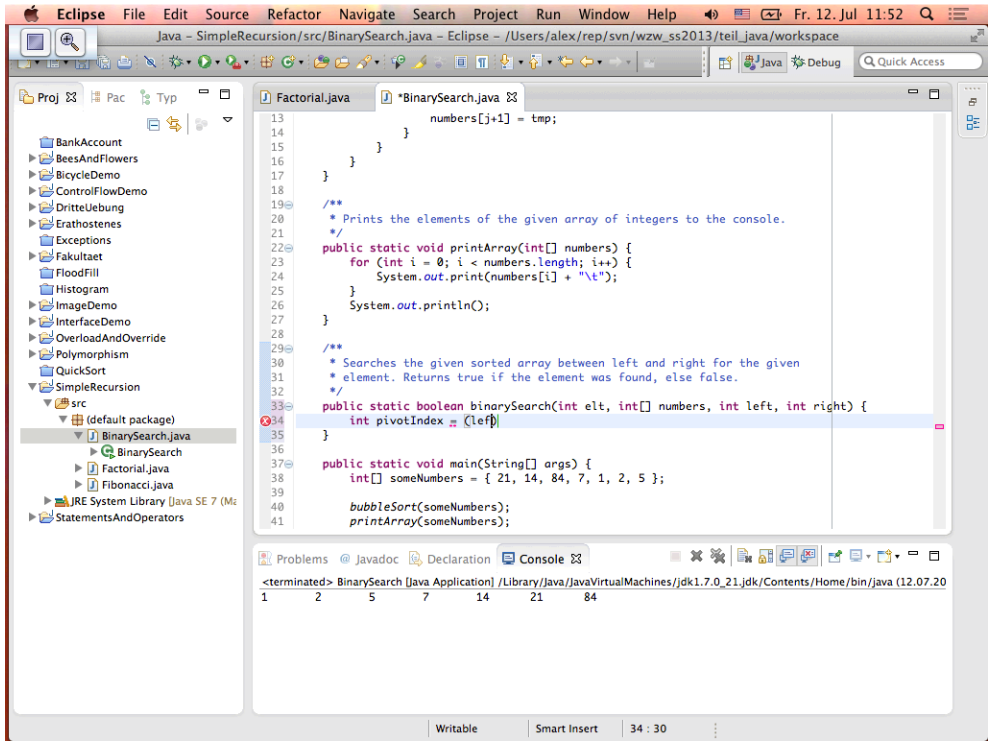

Preview File Edit View Go Tools Bookmarks Window Help
Fr. 12. Jul 11:42
Introduction to Java Basics (page 107 of 168)
4 Recursion
Recursive method calls & Stack
Local variables and parameters stored on stack
For each function call, a corresponding stack frame is created
long factorial(int n) {
long temp;
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);
}
0
4
...
0
...

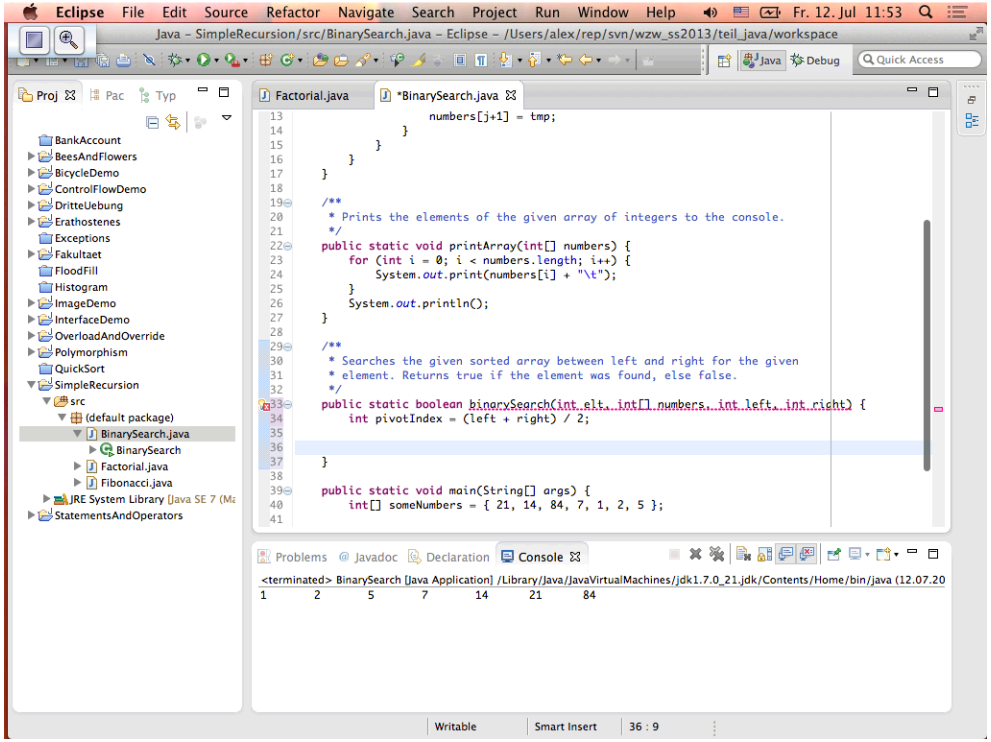

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 12. Jul 11:50
Java – SimpleRecursion/src/Factorial.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Quick Access
Java
Debug
Proj
Pac
Typ
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
Factorial.java
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;
}
}
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
Writable
Smart Insert
14 : 5

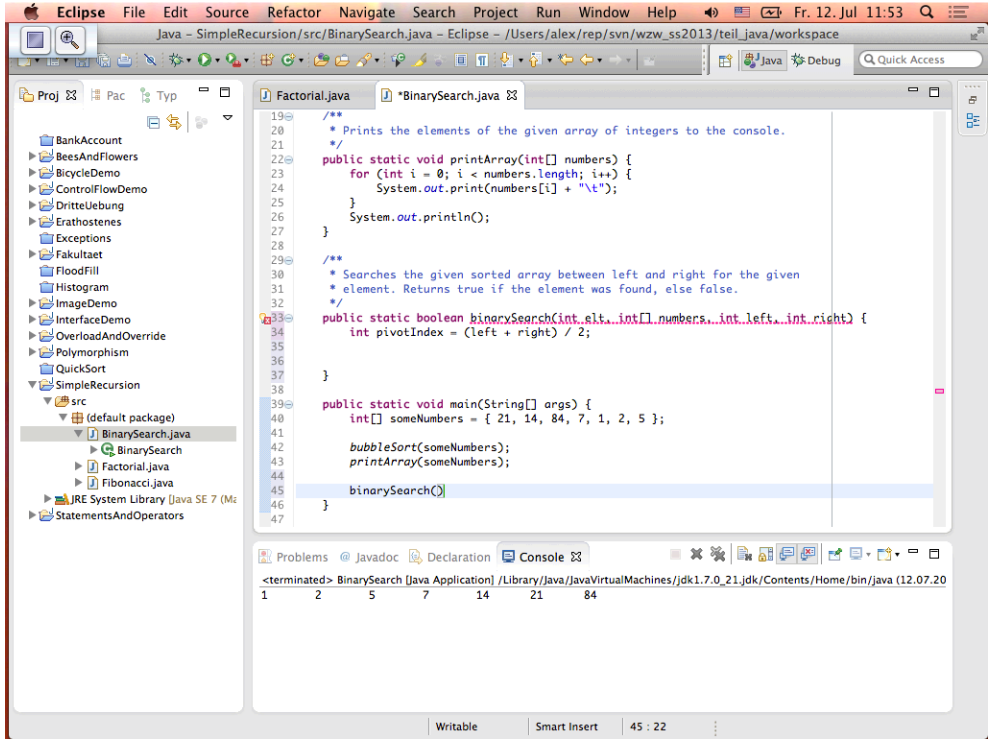

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:50
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
BinarySearch.java
numbers[j+1] = tmp;
}
}
}
}
/**
* Prints the elements of the given array 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) {
// }
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
}
}
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
Writable Smart Insert 1 : 1

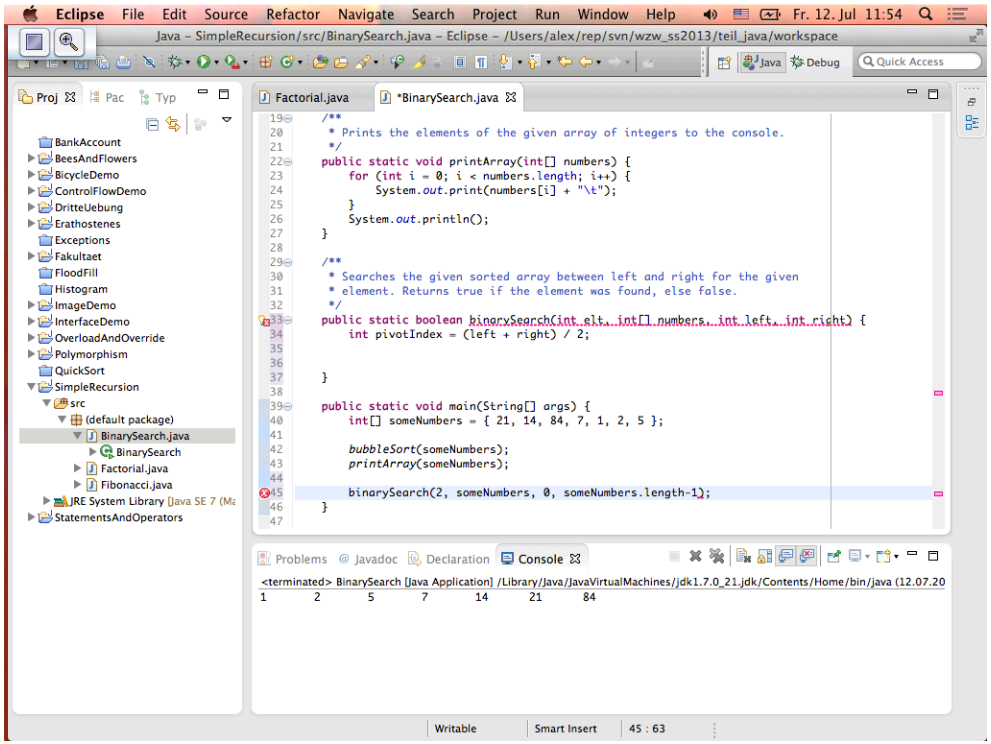

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:51
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
numbers[j+1] = tmp;
}
}
}
}
/**
* Prints the elements of the given 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) {
}
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
}
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
Writable Smart Insert 39 : 9

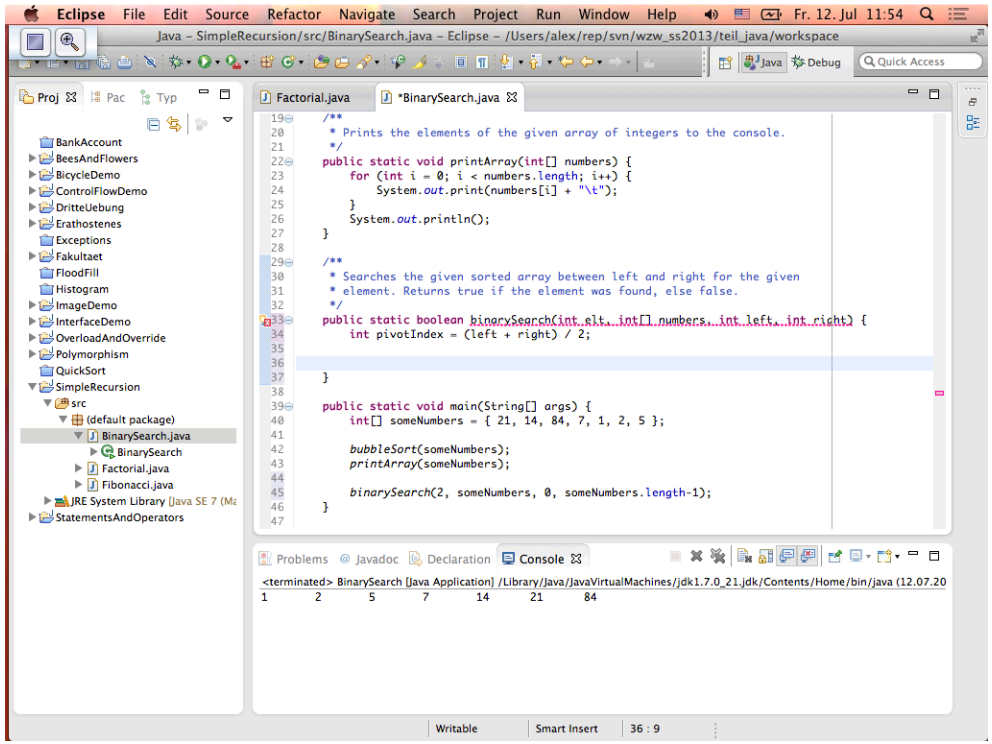

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:51
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
numbers[j+1] = tmp;
}
}
}
}
/**
* Prints the elements of the given 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) {
}
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
bub
}
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
Writable Smart Insert 39 : 11

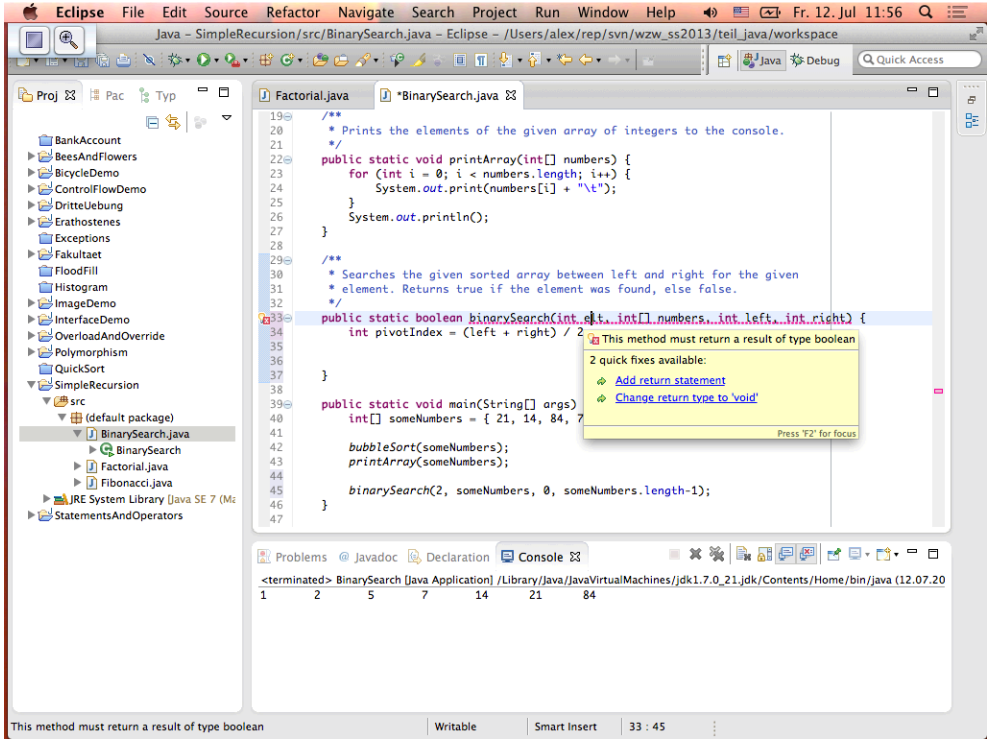

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:52
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
numbers[j+1] = tmp;
}
}
}
}
/**
* Prints the elements of the given 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)
}
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
bubbleSort(someNumbers);
printArray(someNumbers);
}
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
Writable Smart Insert 34 : 30

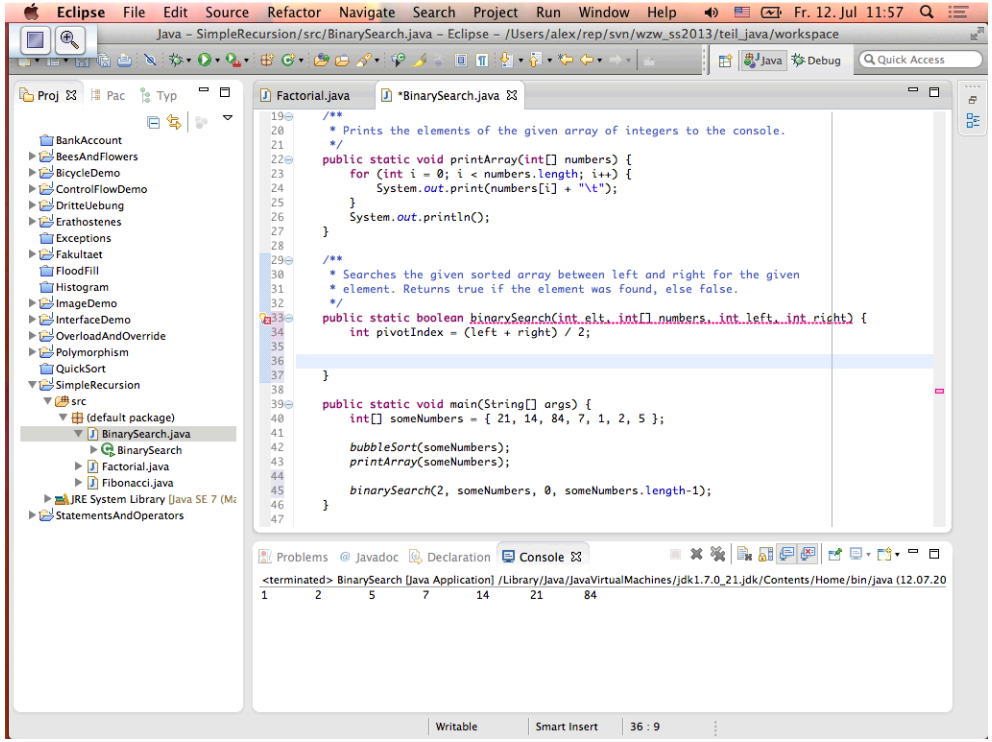

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:53
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
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 (Ma
StatementsAndOperators
numbers[j+1] = tmp;
}
}
}
}
/**
* 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;
}
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
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
Writable Smart Insert 36 : 9

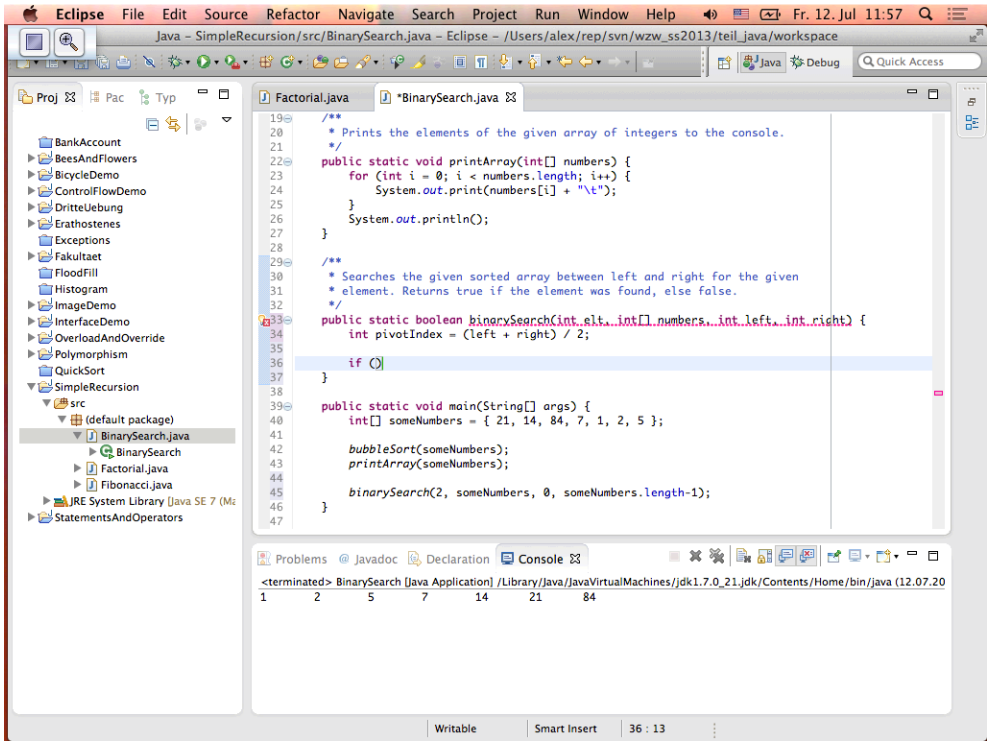

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:53
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
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 (Ma
StatementsAndOperators
/**
* 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;
}
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
bubbleSort(someNumbers);
printArray(someNumbers);
binarySearch()
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
Writable Smart Insert 45 : 22

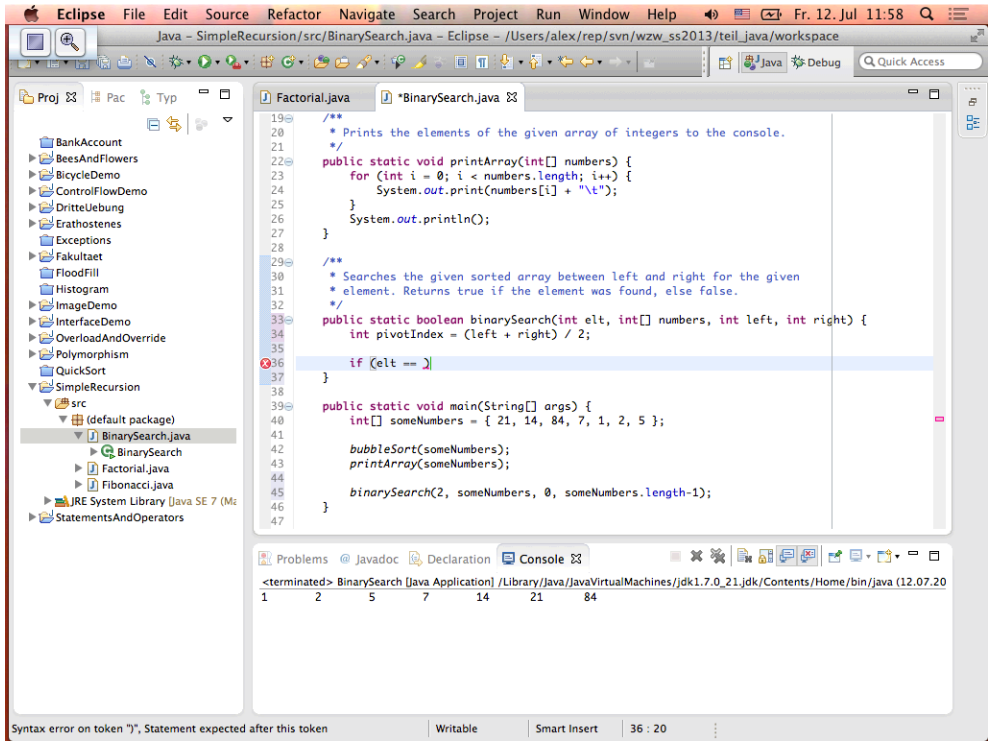

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:54
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
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 (Ma
StatementsAndOperators
/**
* 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;
}
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 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
Writable Smart Insert 45 : 63

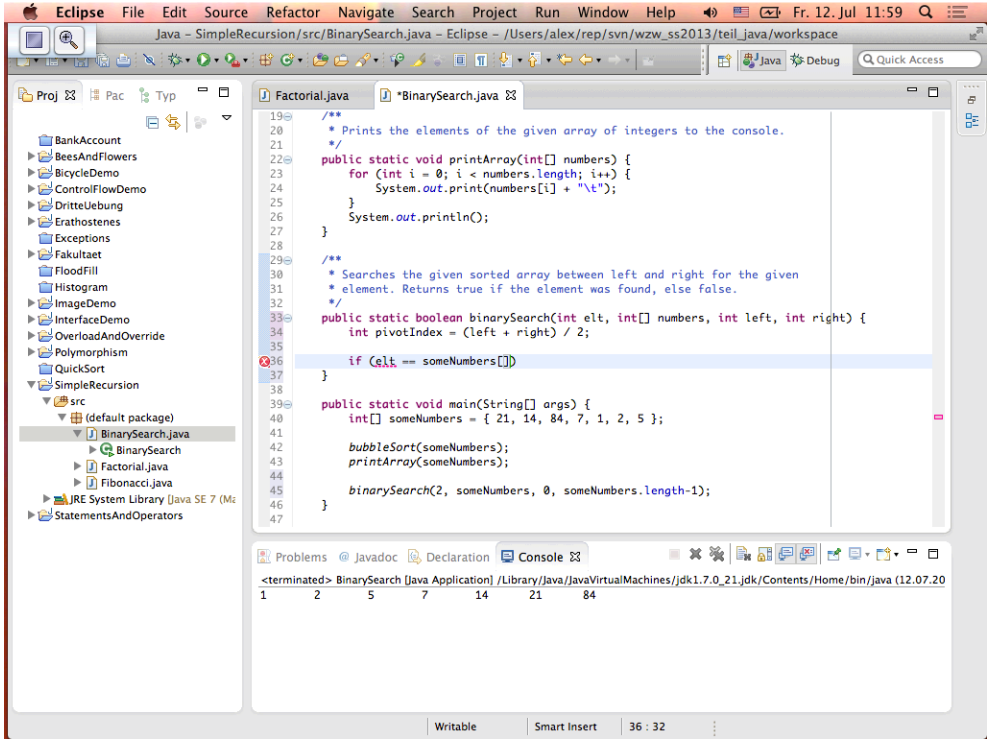

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:54
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
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 (Ma
StatementsAndOperators
/**
* 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;
}
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 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
Writable Smart Insert 36 : 9

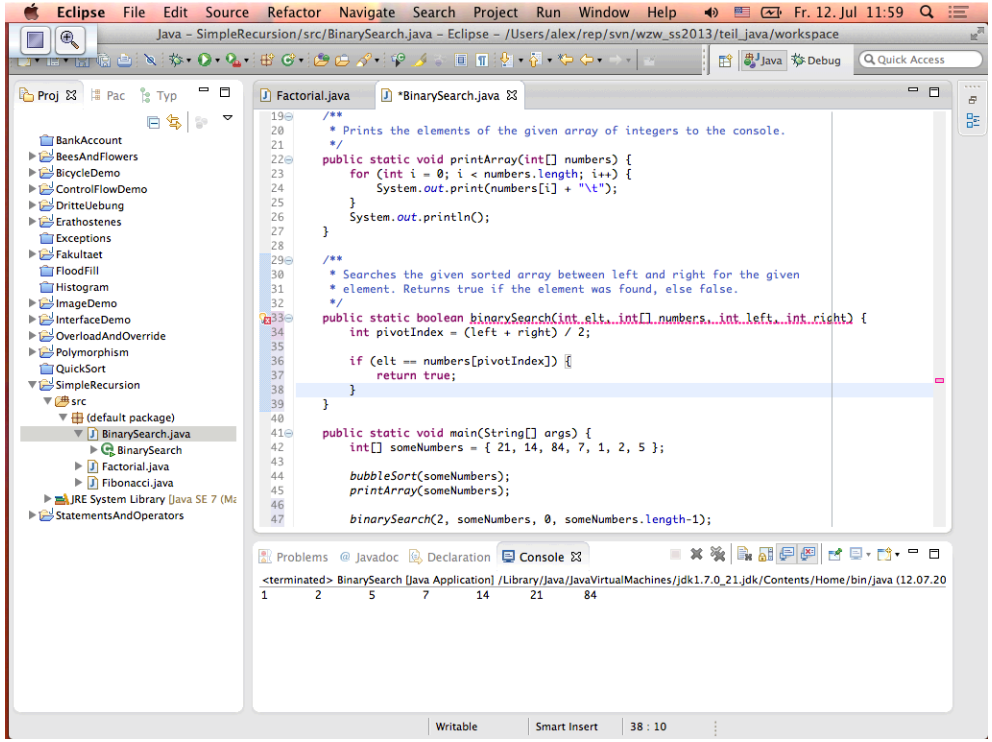

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:56
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
/**
* Prints the elements of the given array of integers to the console.
*/
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
}
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7
bubbleSort(someNumbers);
printArray(someNumbers);
binarySearch(2, someNumbers, 0, someNumbers.length-1);
}
This method must return a result of type boolean
2 quick fixes available:
Add return statement
Change return type to 'void'
Press 'F2' for focus
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

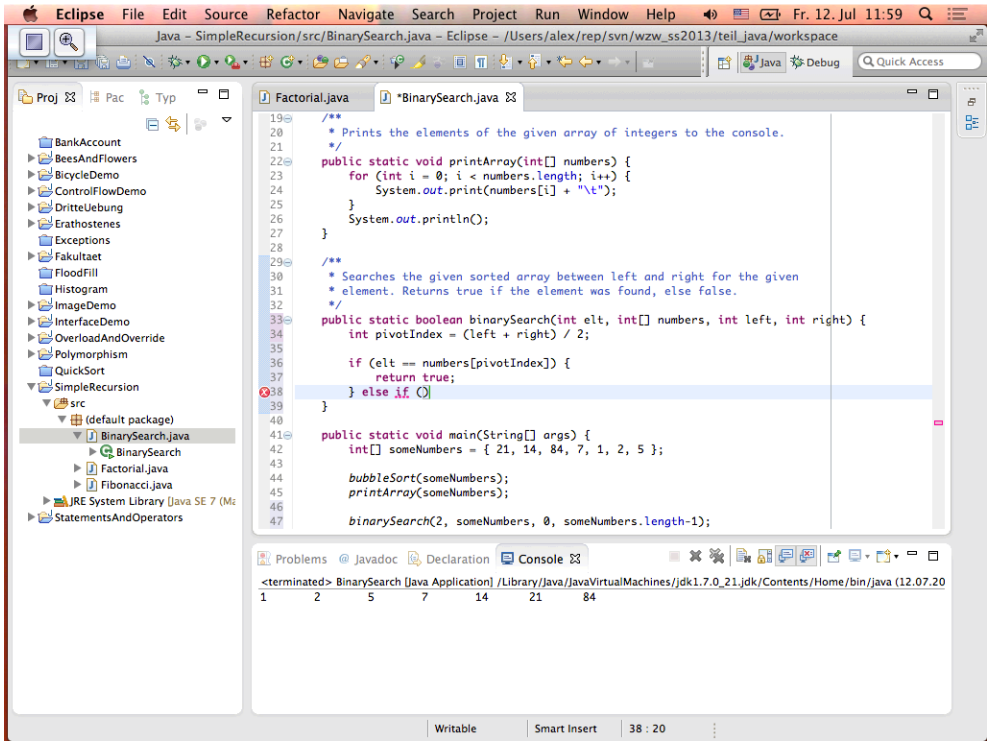

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:57
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
/**
* Prints the elements of the given array of integers to the console.
*/
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;
}
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 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
Writable Smart Insert 36 : 9

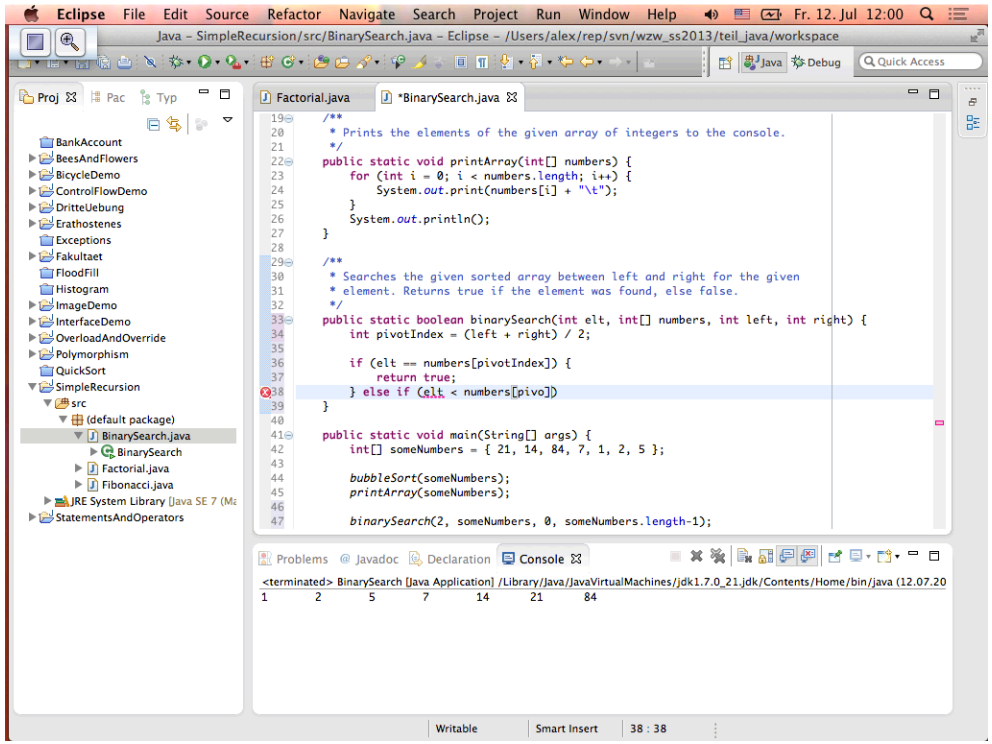

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:57
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
/**
* Prints the elements of the given array of integers to the console.
*/
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 ()
}
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 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
Writable Smart Insert 36 : 13

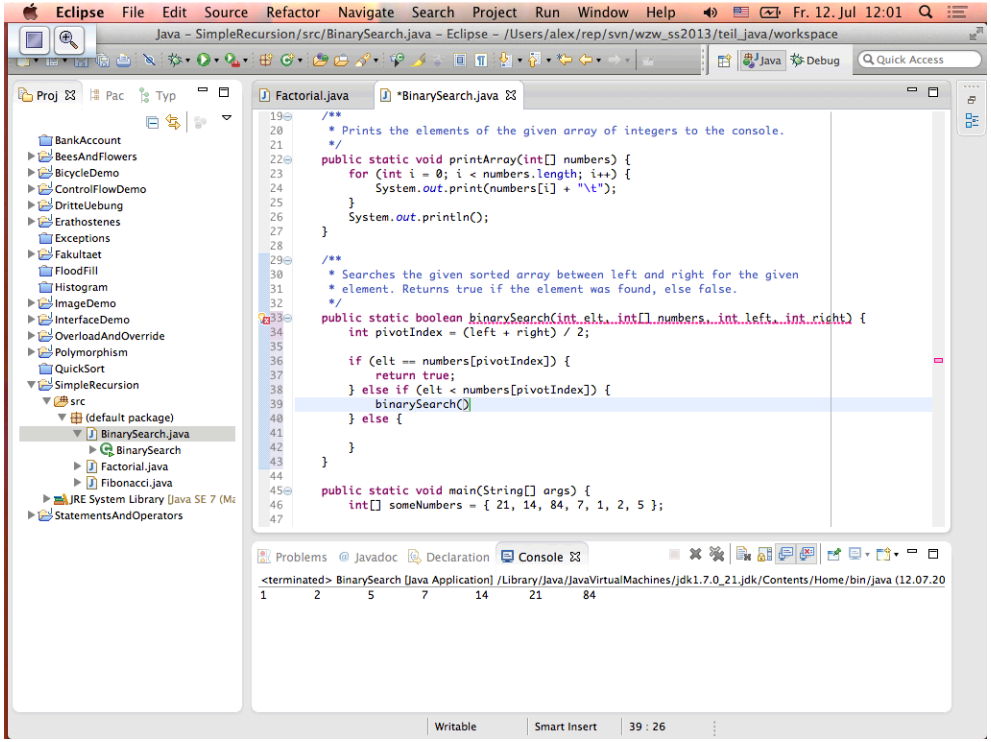

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:58
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
/**
* Prints the elements of the given array of integers to the console.
*/
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 ==)
}
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 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
Syntax error on token ")", Statement expected after this token
Writable Smart Insert 36 : 20

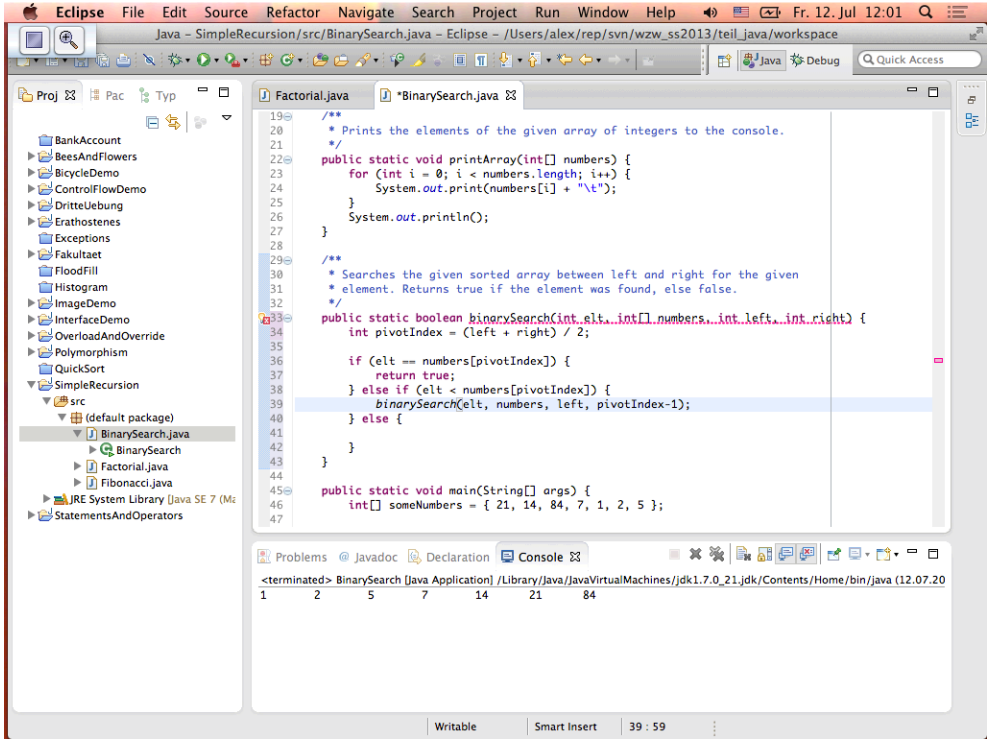

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:59
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
/**
* Prints the elements of the given array of integers to the console.
*/
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 == someNumbers[])
}
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 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
Writable Smart Insert 36 : 32

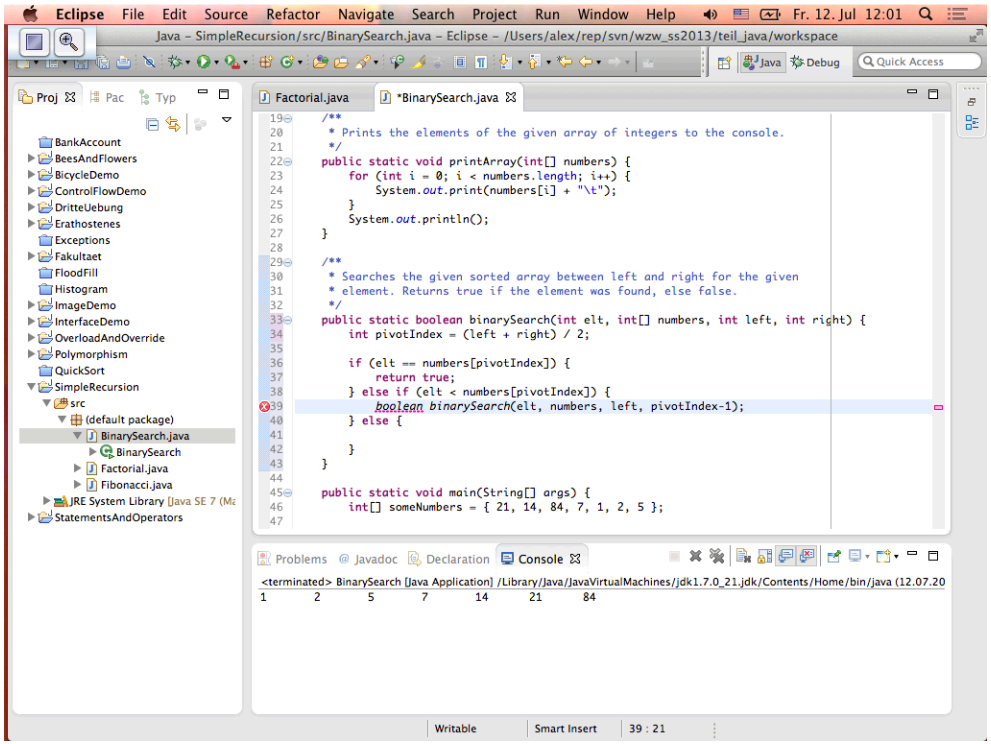

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:59
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
/**
* Prints the elements of the given array of integers to the console.
*/
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;
}
}
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 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
Writable Smart Insert 38 : 10

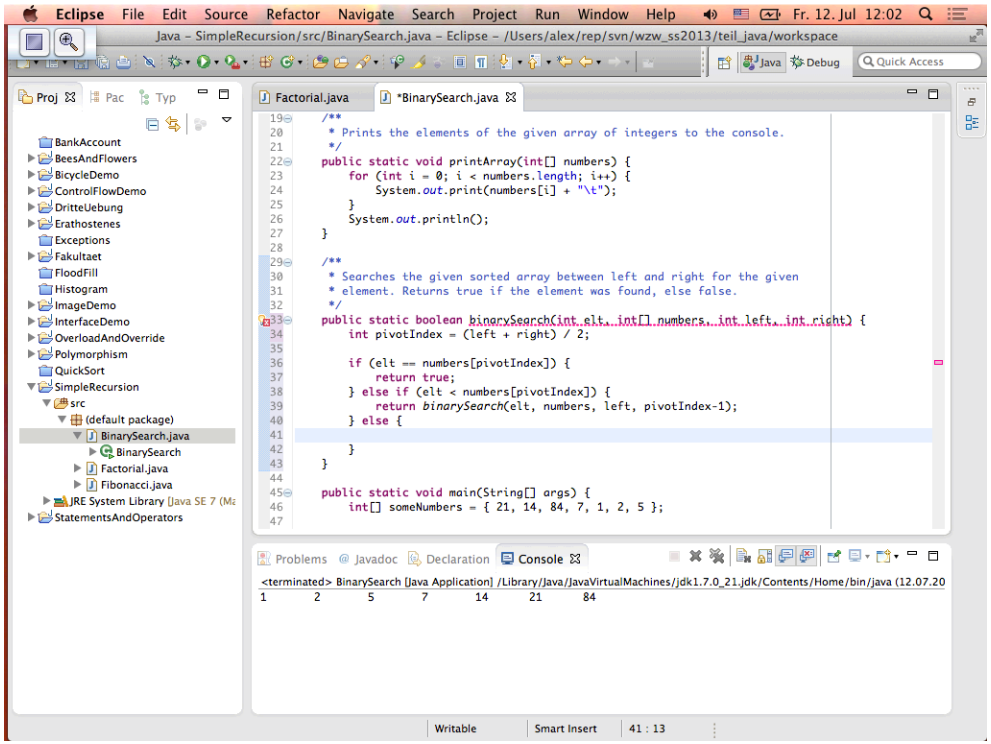

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 11:59
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
/**
* Prints the elements of the given array of integers to the console.
*/
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 ()
}
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 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
Writable Smart Insert 38 : 20

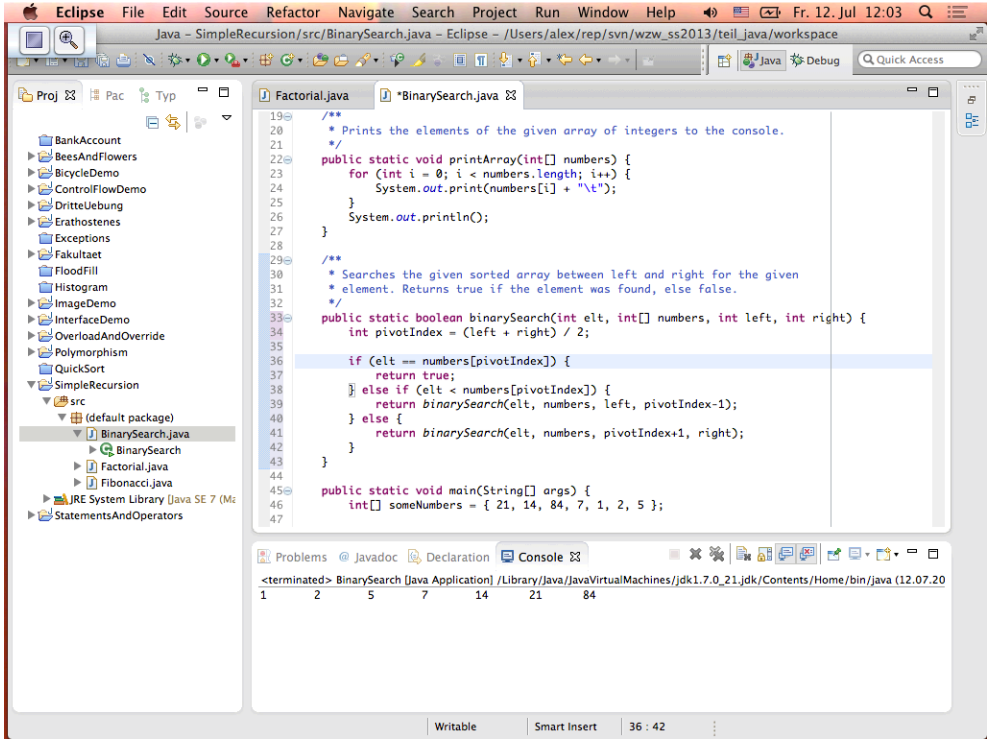

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 12:00
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
/**
* Prints the elements of the given array of integers to the console.
*/
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[pivo])
}
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 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
Writable Smart Insert 38 : 38


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
Factorial.java
*BinarySearch.java
/**
* 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]) {
binarySearch()
} else {
}
}
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
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
Writable Smart Insert 39 : 26


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
Factorial.java
*BinarySearch.java
/**
* 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]) {
binarySearch(elt, numbers, left, pivotIndex-1);
} else {
}
}
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
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
Writable Smart Insert 39 : 59


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
Factorial.java
*BinarySearch.java
/**
* 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]) {
boolean binarySearch(elt, numbers, left, pivotIndex-1);
} else {
}
}
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
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
Writable Smart Insert 39 : 21

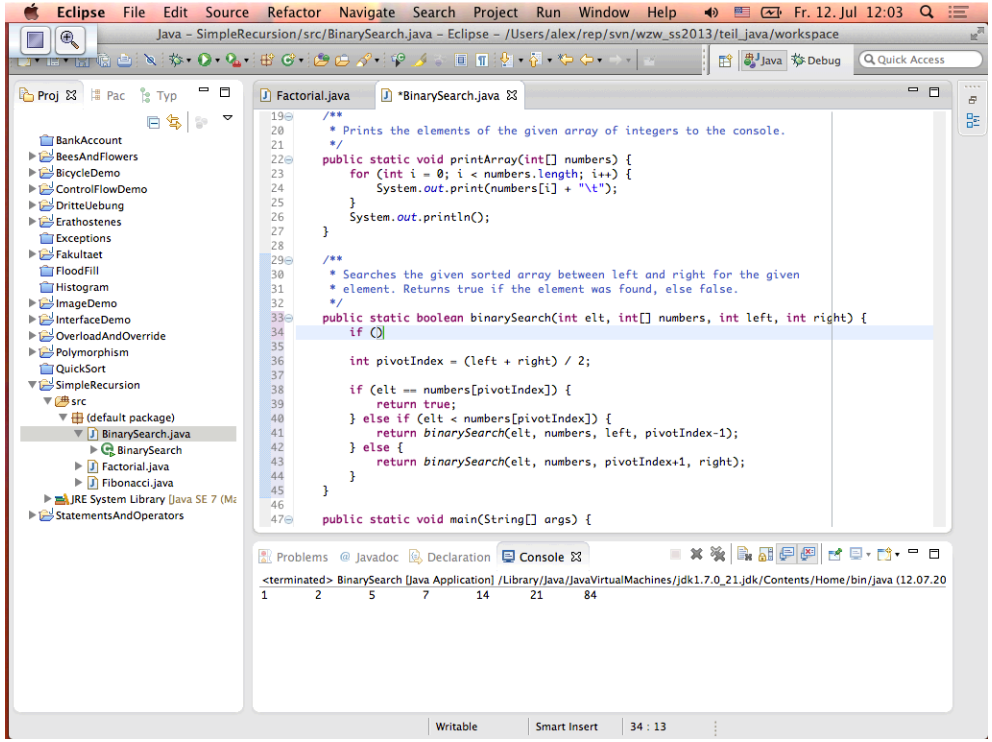

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 12:02
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
/**
* Prints the elements of the given array of integers to the console.
*/
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 {
}
}
public static void main(String[] args) {
int[] someNumbers = { 21, 14, 84, 7, 1, 2, 5 };
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
Writable Smart Insert 41 : 13



```
    /**
     * 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 };
```

```
<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
```

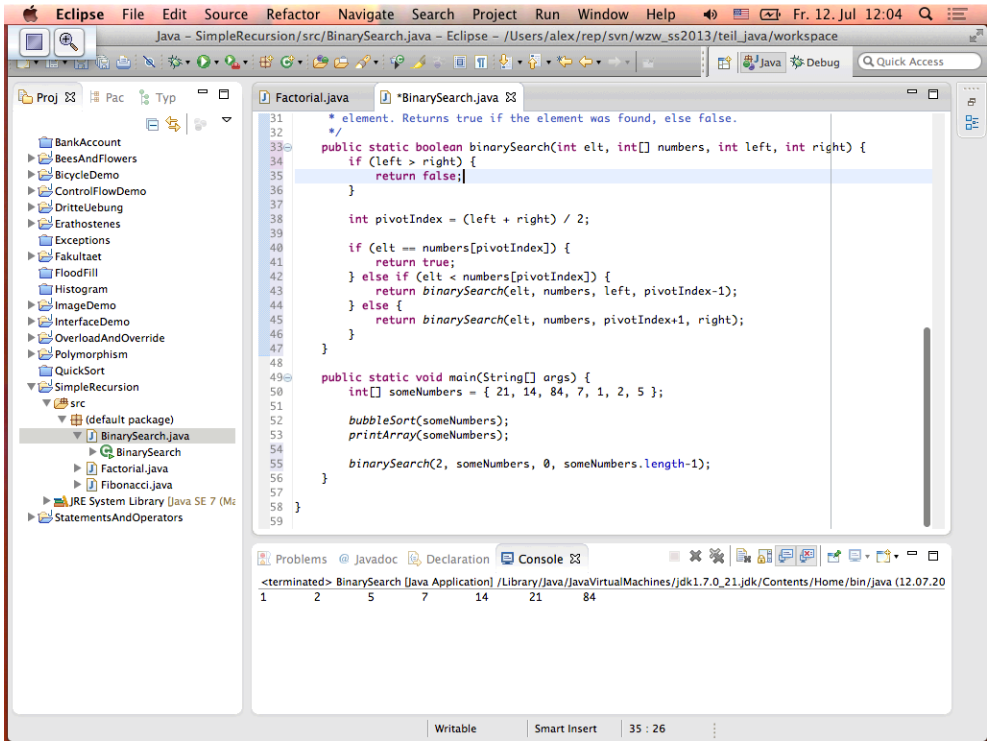
Writable | Smart Insert | 36 : 42



```
    /**
     * 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 ()

        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) {
```

```
<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
```

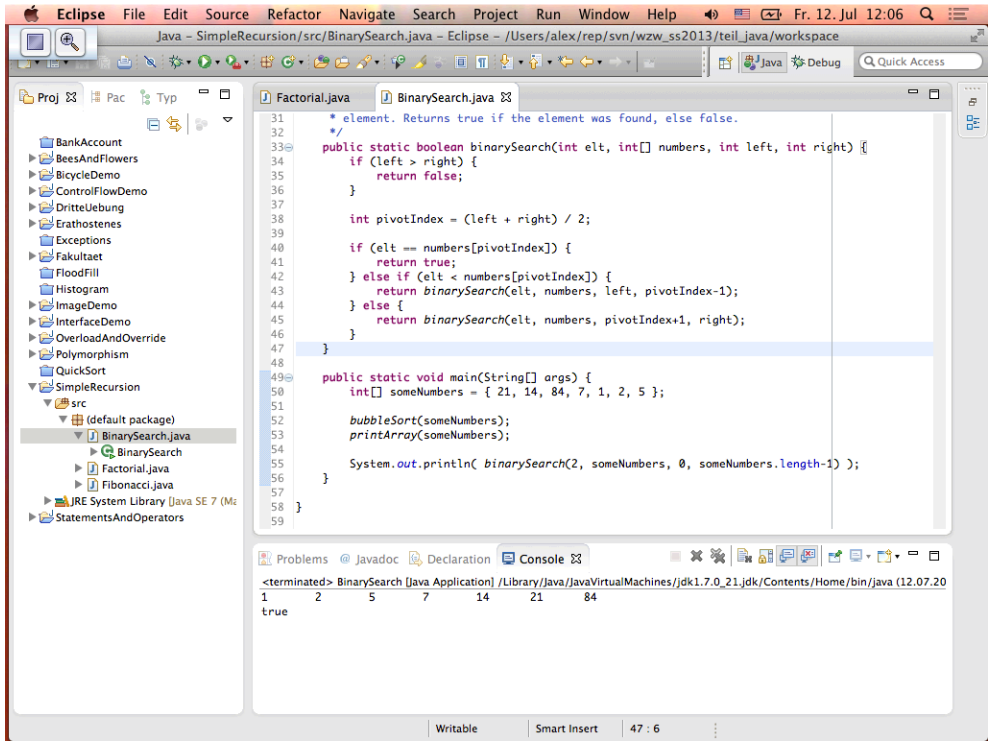
Writable | Smart Insert | 34 : 13



```
     * 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);
    }
}
```

```
<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
```

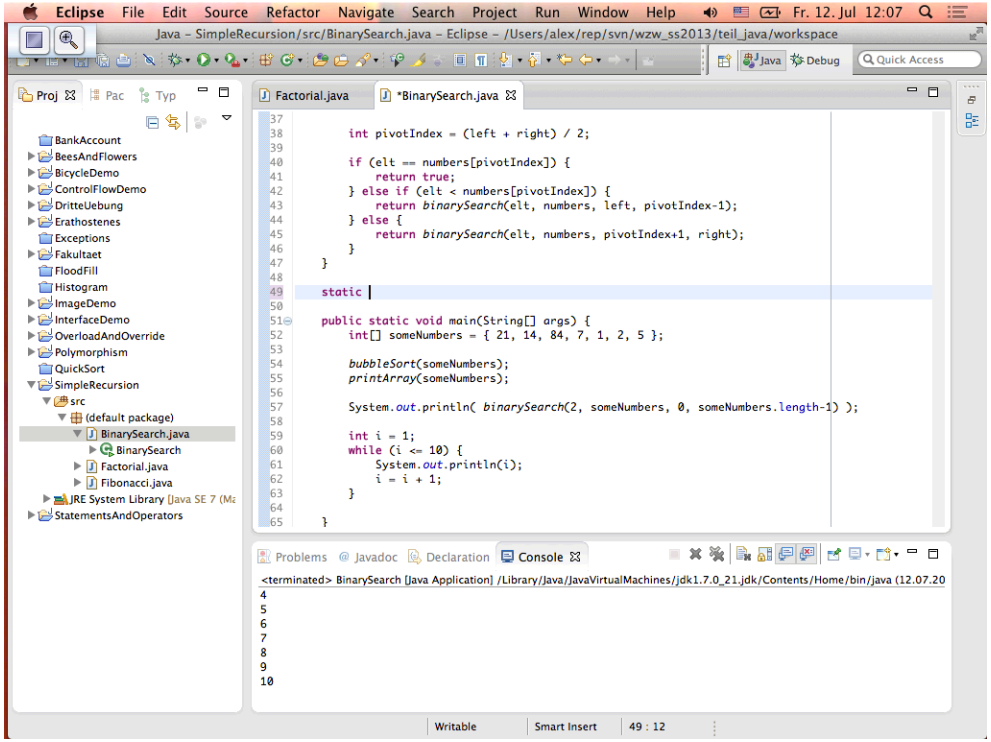
Writable | Smart Insert | 35 : 26

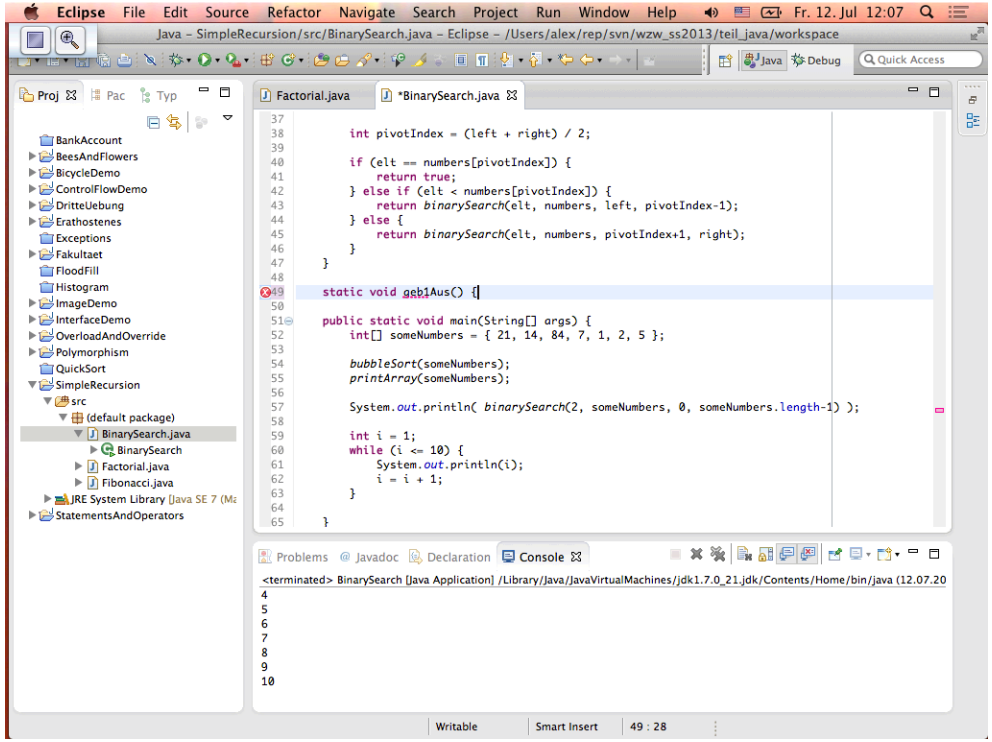


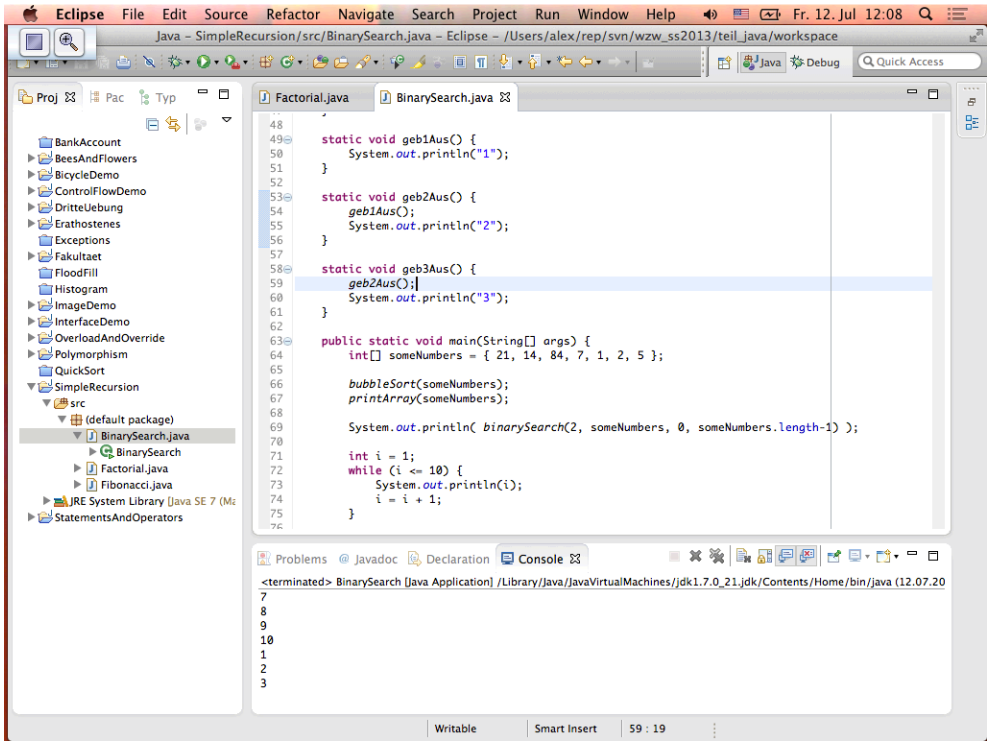
```
     * element. Returns true if the element was found, else false.
     */
    public static boolean binarySearch(int elt, int[] numbers, int left, int right) {
        if (left > right) {
            return false;
        }

        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) );
    }
}
```

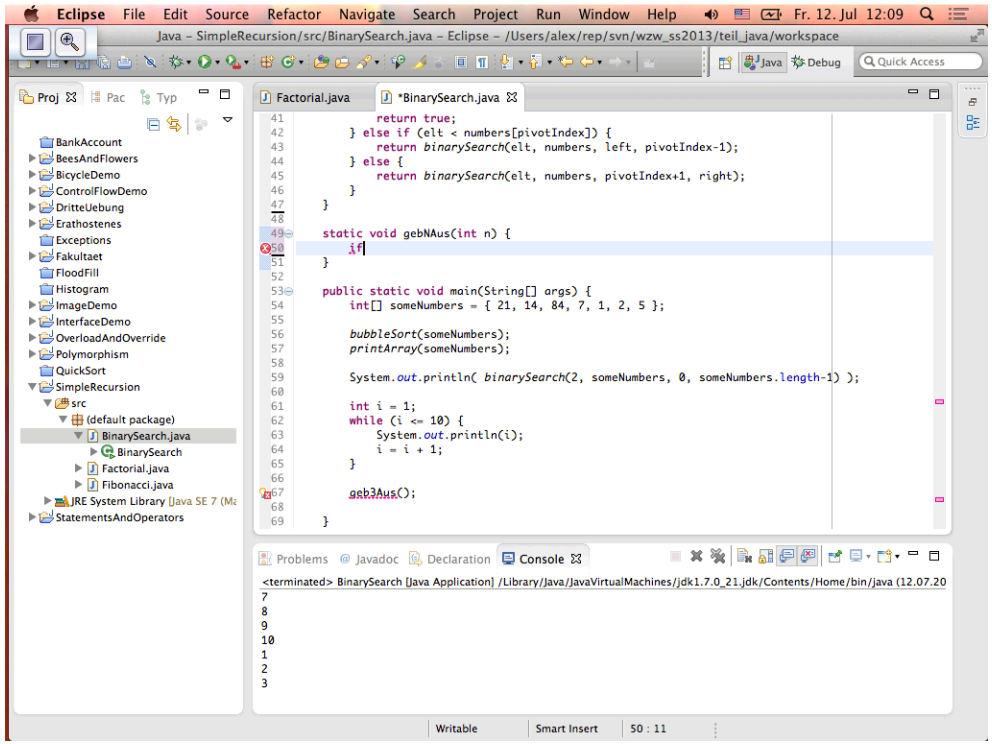
```
<terminated> BinarySearch [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (12.07.20
1       2       5       7       14      21      84
true
```

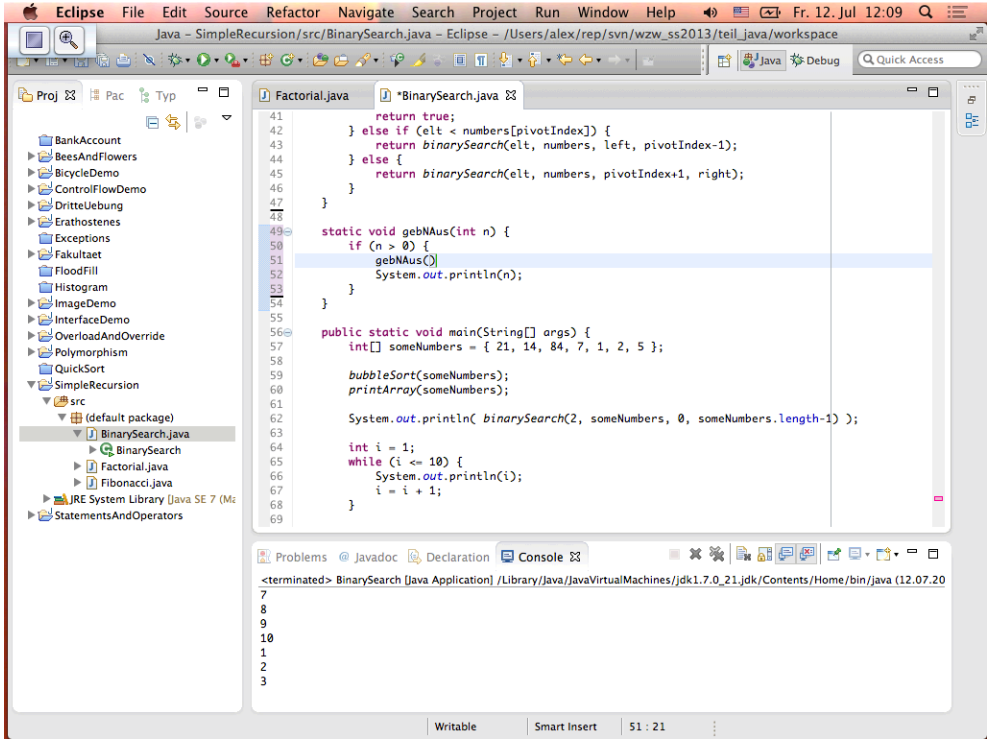
Writable | Smart Insert | 47 : 6

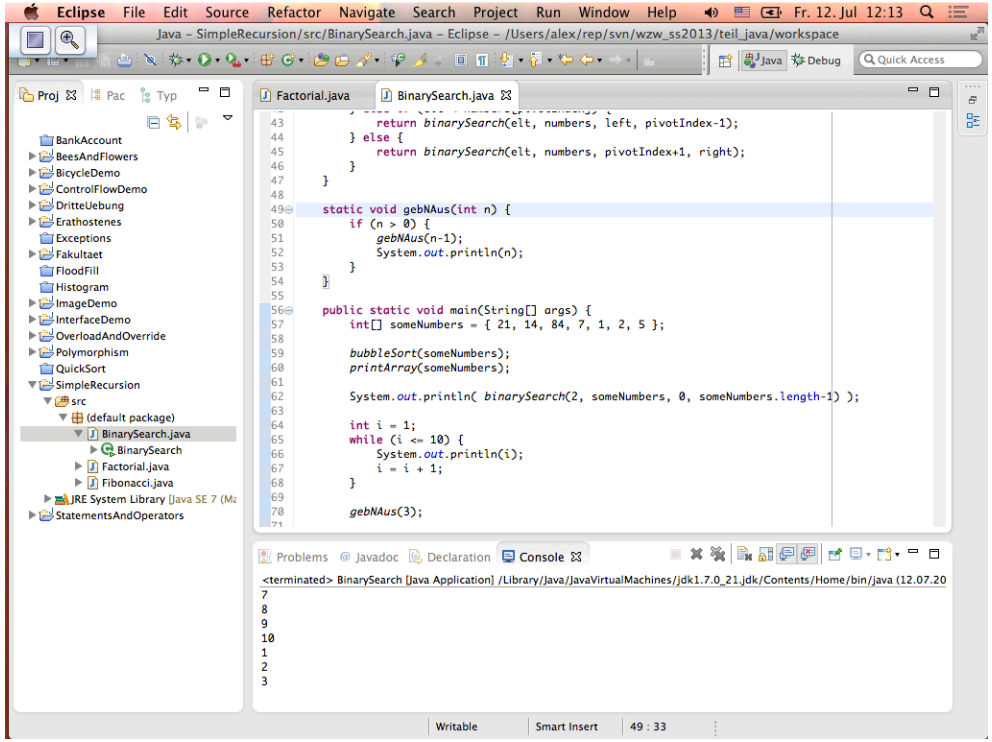

Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
int pivotIndex = (left + right) / 2;
if (elt == numbers[pivotIndex]) {
return true;
} else if (elt < numbers[pivotIndex]) {
return binarySearch(elt, numbers, left, pivotIndex-1);
} else {
return binarySearch(elt, numbers, pivotIndex+1, right);
}
}
static
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));
int i = 1;
while (i <= 10) {
System.out.println(i);
i = i + 1;
}
}
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
Writable
Smart Insert
49 : 12

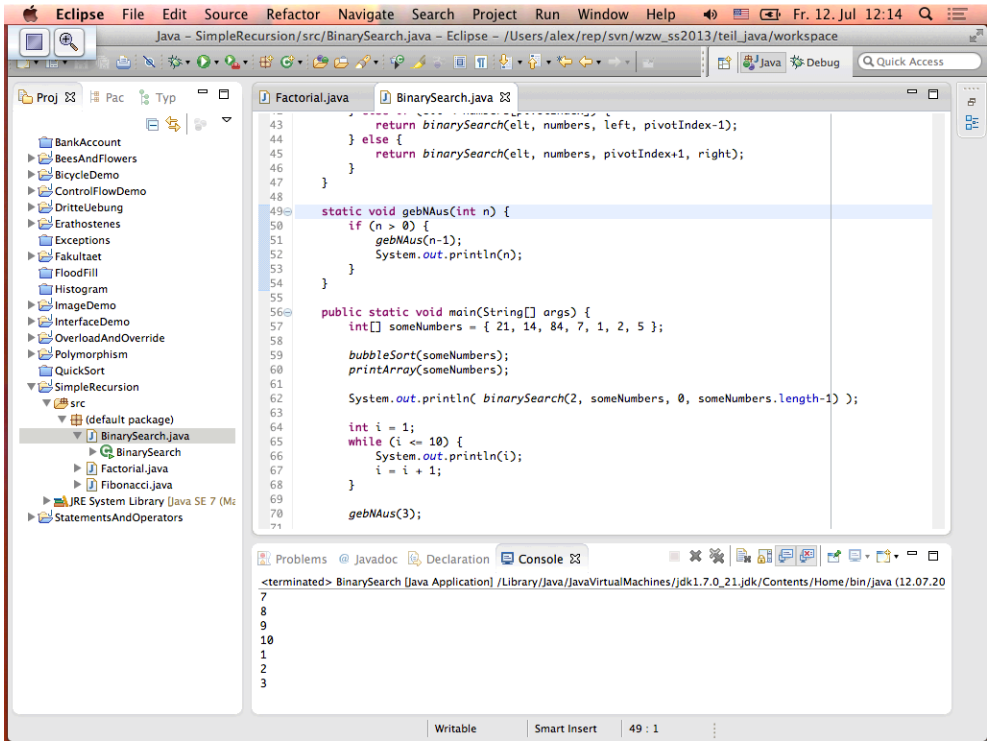

Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
int pivotIndex = (left + right) / 2;
if (elt == numbers[pivotIndex]) {
return true;
} else if (elt < numbers[pivotIndex]) {
return binarySearch(elt, numbers, left, pivotIndex-1);
} else {
return binarySearch(elt, numbers, pivotIndex+1, right);
}
}
static void geb1Aus() {
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));
int i = 1;
while (i <= 10) {
System.out.println(i);
i = i + 1;
}
}
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
Writable
Smart Insert
49 : 28

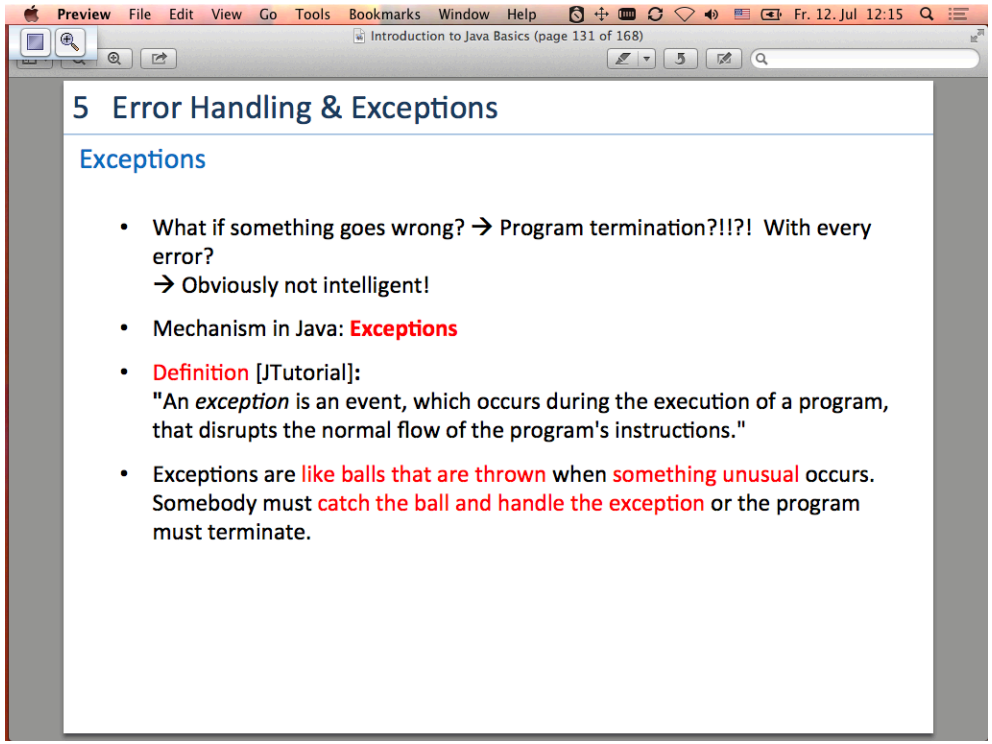

Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
BinarySearch.java
static void geb1Aus() {
System.out.println("1");
}
static void geb2Aus() {
geb1Aus();
System.out.println("2");
}
static void geb3Aus() {
geb2Aus();
System.out.println("3");
}
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));
int i = 1;
while (i <= 10) {
System.out.println(i);
i = i + 1;
}
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
Writable
Smart Insert
59 : 19


Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
*BinarySearch.java
return true;
} else if (elt < numbers[pivotIndex]) {
return binarySearch(elt, numbers, left, pivotIndex-1);
} else {
return binarySearch(elt, numbers, pivotIndex+1, right);
}
}
static void gebNAus(int n) {
if
}
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));
int i = 1;
while (i <= 10) {
System.out.println(i);
i = i + 1;
}
geb3Aus();
}
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
Writable
Smart Insert
50 : 11


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
Factorial.java
*BinarySearch.java
return true;
} else if (elt < numbers[pivotIndex]) {
return binarySearch(elt, numbers, left, pivotIndex-1);
} else {
return binarySearch(elt, numbers, pivotIndex+1, right);
}
}
static void gebNAus(int n) {
if (n > 0) {
gebNAus()
System.out.println(n);
}
}
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));
int i = 1;
while (i <= 10) {
System.out.println(i);
i = i + 1;
}
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 (Ma
StatementsAndOperators
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
Writable Smart Insert 51 : 21


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 12:13
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
BinarySearch.java
return binarySearch(elt, numbers, left, pivotIndex-1);
} else {
return binarySearch(elt, numbers, pivotIndex+1, right);
}
}
static void gebNAus(int n) {
if (n > 0) {
gebNAus(n-1);
System.out.println(n);
}
}
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));
int i = 1;
while (i <= 10) {
System.out.println(i);
i = i + 1;
}
gebNAus(3);
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 (Ma
StatementsAndOperators
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
Writable Smart Insert 49 : 33

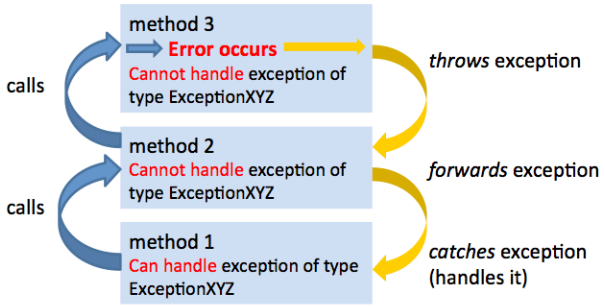

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 12:14
Java – SimpleRecursion/src/BinarySearch.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
BinarySearch.java
return binarySearch(elt, numbers, left, pivotIndex-1);
} else {
return binarySearch(elt, numbers, pivotIndex+1, right);
}
}
static void gebNAus(int n) {
if (n > 0) {
gebNAus(n-1);
System.out.println(n);
}
}
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));
int i = 1;
while (i <= 10) {
System.out.println(i);
i = i + 1;
}
gebNAus(3);
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 (Ma
StatementsAndOperators
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
Writable Smart Insert 49 : 1


Preview File Edit View Go Tools Bookmarks Window Help Fr. 12. Jul 12:15
Introduction to Java Basics (page 131 of 168)
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.

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.



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();
}
```


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 occured or not.
    if (fileWriter != null) {
        fileWriter.close();
    }
}
```


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 occured or not.
    if (fileWriter != null) {
        fileWriter.close();
    }
}
```

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();
}
```


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 {
        // ...
    }

}
```


5 Error Handling & Exceptions

Exceptions

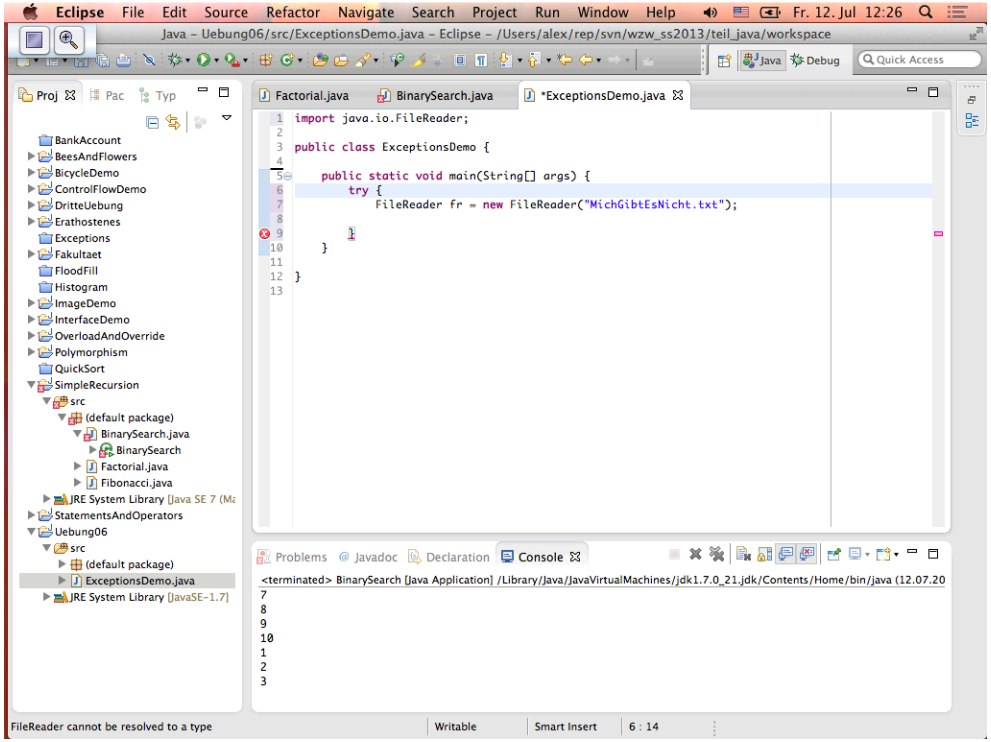
- Checked Exceptions
  - Subclasses of java.lang.Exception
  - *Must* be caught or forwarded where they possibly occur (try/catch or throws)
  - Example: Opening a file (is likely to go wrong)
- Unchecked ("Runtime") Exceptions
  - Subclasses of java.lang.RuntimeException or java.lang.Error
  - *Can*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

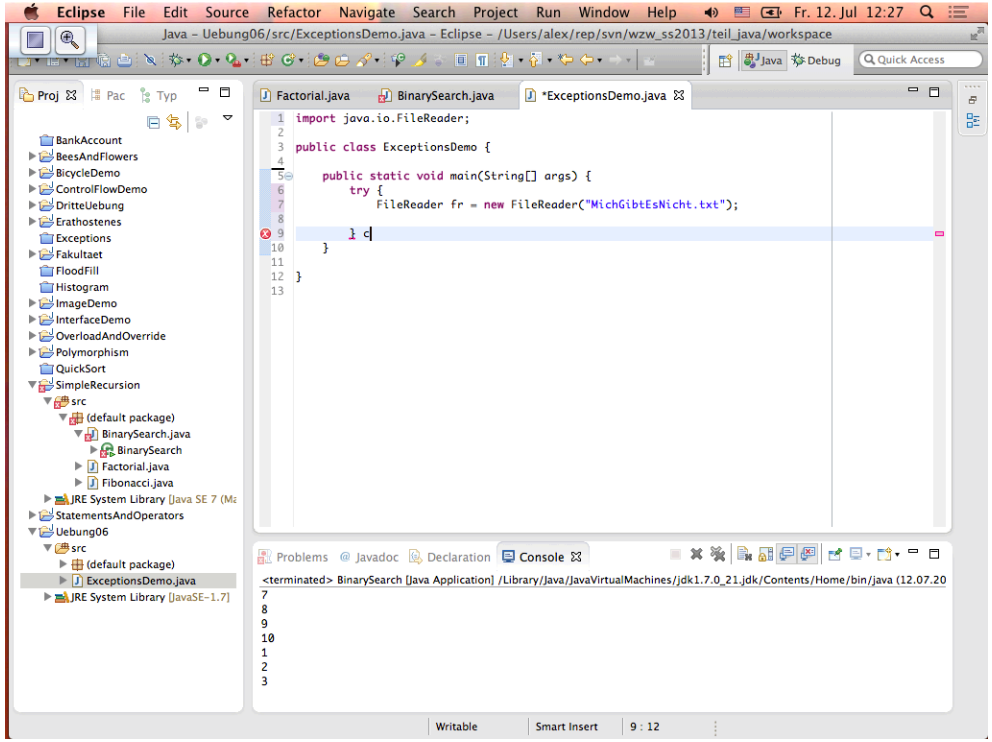
```
static long factorial(int n) {
  if (n < 0) {
    throw new RuntimeException("Check your math!")
  } else if (n == 1) {
    return 1;
  }
  return n * factorial(n-1);
}
```

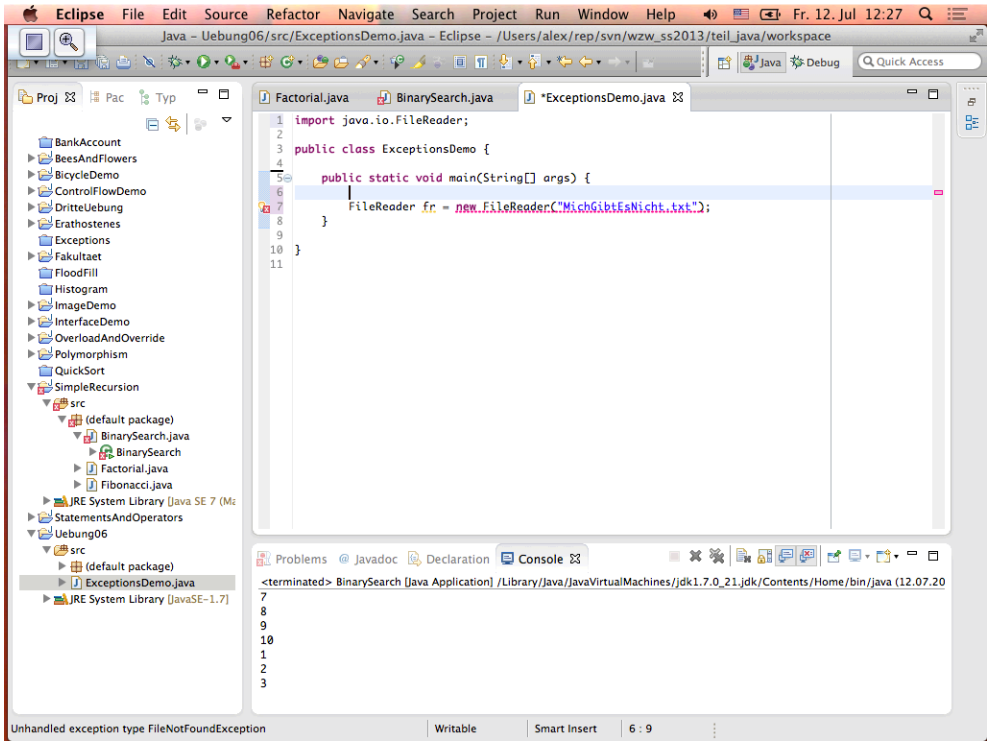

6 Coding & Naming Conventions

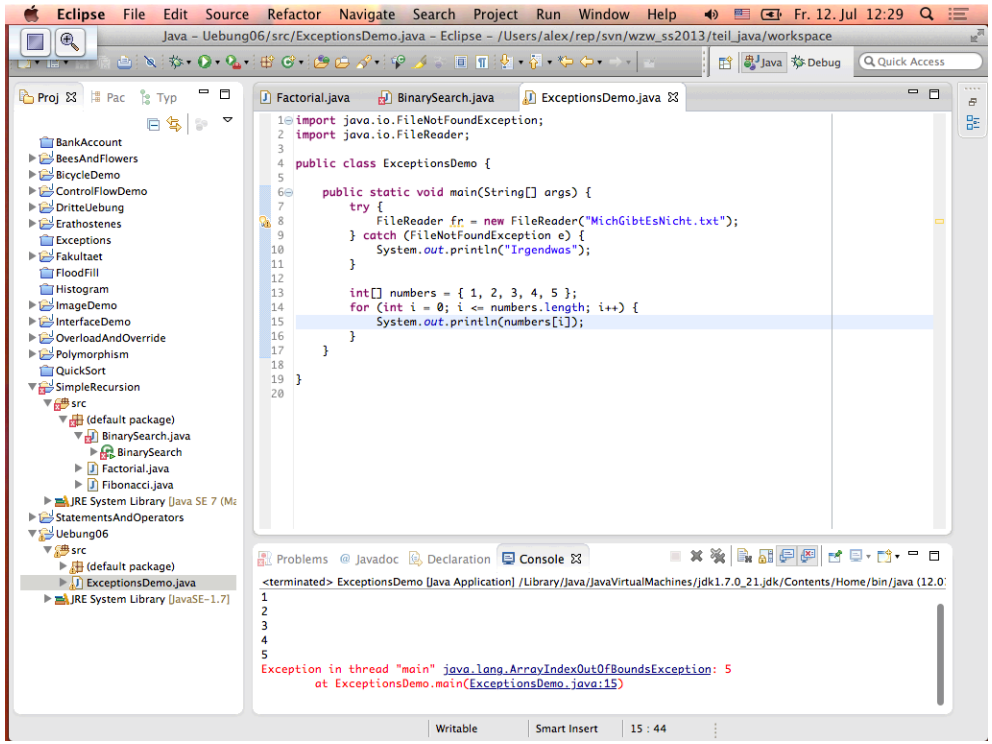
Deepening readings:

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 Search Project Run Window Help Fr. 12. Jul 12:26
Java – Uebung06/src/ExceptionsDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
BinarySearch.java
*ExceptionsDemo.java
import java.io.FileReader;
public class ExceptionsDemo {
public static void main(String[] args) {
try {
FileReader fr = new FileReader("MichGibtEsNicht.txt");
}
}
}
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
FileReader cannot be resolved to a type
Writable
Smart Insert
6 : 14


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 12:27
Java – Uebung06/src/ExceptionsDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
BinarySearch.java
*ExceptionsDemo.java
import java.io.FileReader;
public class ExceptionsDemo {
public static void main(String[] args) {
try {
FileReader fr = new FileReader("MichGibtEsNicht.txt");
} c
}
}
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
Writable
Smart Insert
9 : 12


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 12:27
Java – Uebung06/src/ExceptionsDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
BinarySearch.java
*ExceptionsDemo.java
import java.io.FileReader;
public class ExceptionsDemo {
public static void main(String[] args) {
FileReader fr = new FileReader("MichGibtEsNicht.txt");
}
}
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
Unhandled exception type FileNotFoundException
Writable
Smart Insert
6 : 9


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 12. Jul 12:29
Java – Uebung06/src/ExceptionsDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Factorial.java
BinarySearch.java
ExceptionsDemo.java
import java.io.FileNotFoundException;
import java.io.FileReader;
public class ExceptionsDemo {
public static void main(String[] args) {
try {
FileReader fr = new FileReader("MichGibtEsNicht.txt");
} catch (FileNotFoundException e) {
System.out.println("Irgendwas");
}
int[] numbers = { 1, 2, 3, 4, 5 };
for (int i = 0; i <= numbers.length; i++) {
System.out.println(numbers[i]);
}
}
}
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
15 : 44

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

Java – Uebung06/src/ExceptionsDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

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 (12.0

```
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 Search Project Run Window Help Fr. 12. Jul 12:29

Java – Uebung06/src/ExceptionsDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

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 (12.0

```
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!

7 Using the Java Class Library

7 Using the Java Class Library

Packages and Imports

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

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

- All types from package `java.lang` are imported automatically
- Other types need to be imported

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


7 Using the Java Class Library

Generics

- Some types may be parameterized with other types
- Typical examples are classes that implement data structures
- Advantages:
  - Type checks at compile time
  - Programmers can implement algorithms generically
- Examples:

```
Vector nonGenericVector = new Vector();
nonGenericVector.add( 1234 );
nonGenericVector.add( "Hello world" );
Object typeUnknown = nonGenericVector.get(0);

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


7 Using the Java Class Library

Generics

- Some types may be parameterized with other types
- Typical examples are classes that implement data structures
- Advantages:
  - Type checks at compile time
  - Programmers can implement algorithms generically
- Examples:

```
Vector nonGenericVector = new Vector();
nonGenericVector.add( 1234 );
nonGenericVector.add( "Hello world" );
Object typeUnknown = nonGenericVector.get(0);

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

7 Using the Java Class Library

Wrapper-Classes

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

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

etc.

- Primitive types are automatically boxed and unboxed when necessary

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


8 Solving a Problem

Deepening readings (optional):

The Internet

Main reference:

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