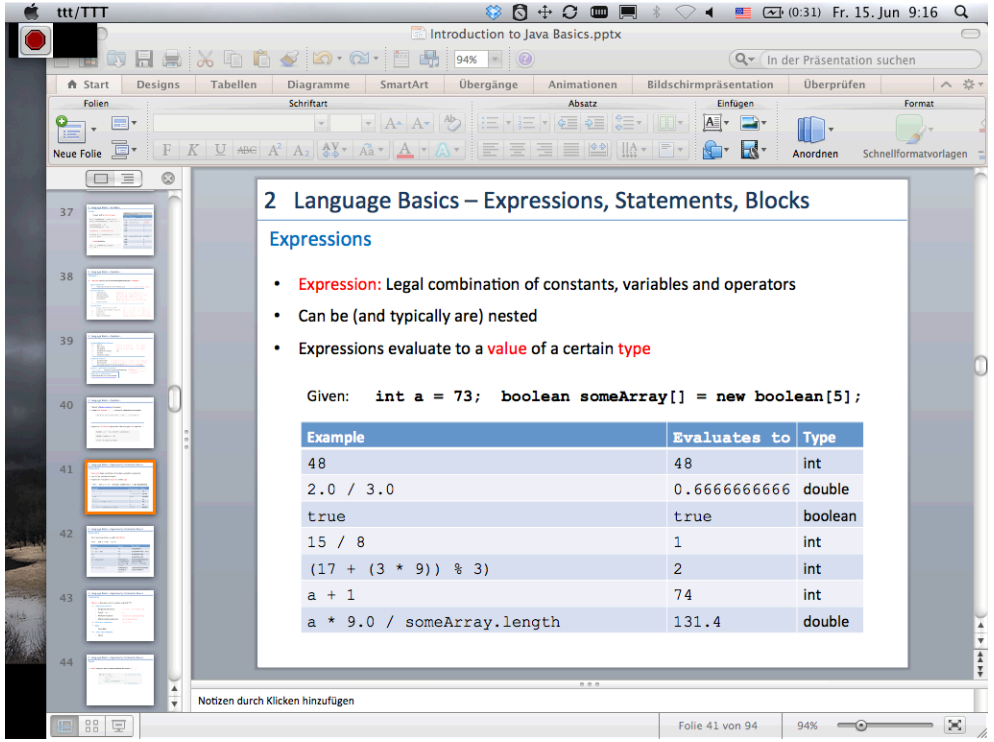
Script generated by TTT

Title: Lehmann: Uebung_Einf_HF (15.06.2012)

Date: Fri Jun 15 09:16:52 CEST 2012

Duration: 87:44 min

Pages: 64



nguage Basics – Expressions, Statements, Blocks

Expressions

- Expression: Legal combination of constants, variables and operators
- Can be (and typically are) nested
- Expressions evaluate to a value of a certain type

Given: `int a = 73; boolean someArray[] = new boolean[5];`

Example	Evaluates to	Type
`48`	`48`	int
`2.0 / 3.0`	`0.6666666666`	double
`true`	`true`	boolean
`15 / 8`	`1`	int
`(17 + (3 * 9)) % 3)`	`2`	int
`a + 1`	`74`	int
`a * 9.0 / someArray.length`	`131.4`	double

nguage Basics – Expressions, Statements, Blocks

Expressions

- Some expressions have so-called side-effects

Given: `int a = 73; int b;`

Example	Value	Side-effect
`a = 84`	`84`	Assign 84 to `a`
`b = (a = 48)`	`48`	Assign 48 to both `a` and `b`
`a++`	`48`	Assign 49 to `a` (!)
`++a`	`50`	Assign 50 to `a` (!)
`new Bicycle()`	Reference to a new instance of `Bicycle`, e.g. `<1150>`	Create and initialize new instance of class `Bicycle` in memory
`new double[10]`	Reference to a new array of `double`	Create and initialize new array in memory

Statements

- Statement: Complete unit of execution (ends with ";")
  - Expression statements:
    - Assigment expressions `a = (17 + (3 * 9)) % 3;`
    - Use of ++ or -- `a++;`
    - Method invocations `someObject.methodOne();`
    - Object creation expressions `new SomeClass();`
  - Declaration statements `int a = 0;`
  - Blocks
    - (next slide)
  - Control flow statements
    - (later)

Blocks

- Block: Group of zero or more statements enclosed in "{" ... "}"

```
if (a == b) {              // begin block
      c = 17;
      f++;
      bbb.someMethod();
}                          // end block
```


Blocks

- Variables declared inside a block are only visible from within that block:

```
int a = 7, b = 6;

if (a != b) {              // begin block
  int c;
  c = a * b;
  System.out.println(c);
}                          // end block

System.out.println(c);   // ERROR: c unavailable
```


Control Flow Statements

- Control flow statements:
  Allow for deviation of control flow from sequential order of statements:
  - conditionals: `if, if else, switch`
  - loops: `while, do while, for`
  - branches: `break, continue, return`

nguage Basics – Control Flow Statements

- if and if else have a straightforward meaning:

```
void applyBrakes(){
     if (speed > 0) {
          speed = speed - 1;
     }
}
```

```
void applyBrakes(){
     if (speed > 0) {
          speed--;
     } else {
          System.err.println(
              "The bicycle has already stopped!");
     }
}
```

- switch: Equivalent to sequence of chained if else statements

nguage Basics – Control Flow Statements

- while: do something as long as some condition (boolean expression) is true

```
int count = 1;
while (count < 8) {
       System.out.print("#:" + count + " ");
       count++;
}
```

⇨ output will be: `#:1 #:2 #:3 #:4 #:5 #:6 #:7`

- do while: similar to "while", but check condition at the end of execution of something instead of at the beginning

```
int count = 1;
do {
       System.out.print("#:" + count + " ");
       count++;
} while (count < 8)
```

⇨ output will be: `#:1 #:2 #:3 #:4 #:5 #:6 #:7`

nguage Basics – Control Flow Statements

- for: usually means to do something for a fixed number of times:

```
for (int i=0; i<7; i++) { // loop will be executed 7 times
       System.out.print("#:" + i + " ");
}
```

⇨ output will be: `#:0 #:1 #:2 #:3 #:4 #:5 #:6`

- General form:

```
for (initialization; termination; update) {
       statement*
}
```

  - initialization expression: Executed once at the beginning of first loop
  - termination expression: If `true` then execute statement(s), else exit loop
  - update expression: Executed after each iteration of the loop

nguage Basics – Control Flow Statements

- for: usually means to do something for a fixed number of times:

```
for (int i=0; i<7; i++) { // loo                    imes
       System.out.print("#:" + i
}
```

⇨ output will be: `#:0 #:`

- for equivalent to while

```
initialization;
while (termination) {
      statement*
      update;
}
```

- General form:

```
for (initialization; termination; update) {
       statement*
}
```

  - initialization expression: Executed once at the beginning of first loop
  - termination expression: If `true` then execute statement(s), else exit loop
  - update expression: Executed after each iteration of the loop

- break: force termination of a loop
- continue: skip current iteration of a loop

} can be avoided in almost all relevant cases

```
for (int i=0; i<10; i++) {
      if (i == 8) {
            break;
      } else if (i % 2 == 0) {
            continue;
      }
      System.out.print("#:" + i + " ");
}
```

⟹ output will be: `#:1 #:3 #:5 #:7`

- return: terminate current method and return control flow to where the method was invoked from (will be covered shortly in more detail)

3 Classes, Objects, Inheritance

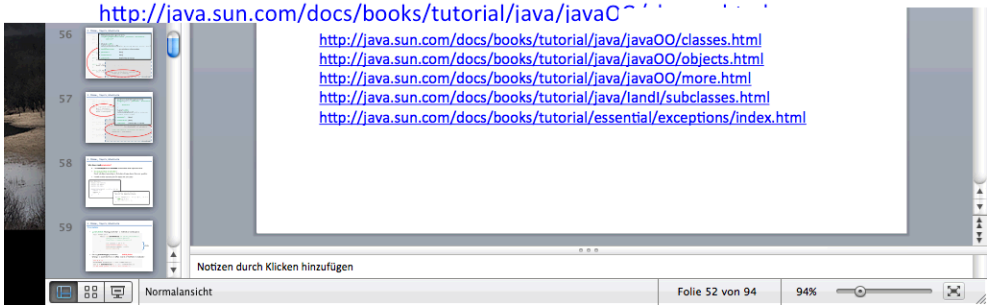
Deepening readings:

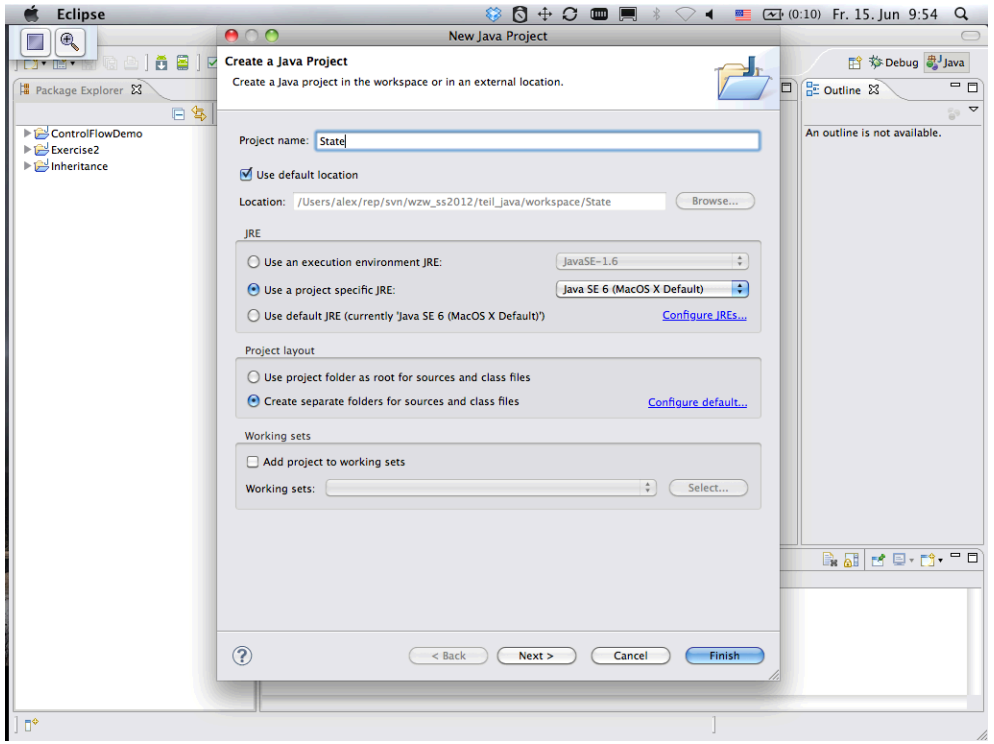
http://java.sun.com/docs/books/tutorial/java/javaOO/classes.html
http://java.sun.com/docs/books/tutorial/java/javaOO/objects.html
http://java.sun.com/docs/books/tutorial/java/javaOO/more.html
http://java.sun.com/docs/books/tutorial/java/IandI/subclasses.html
http://java.sun.com/docs/books/tutorial/essential/exceptions/index.html

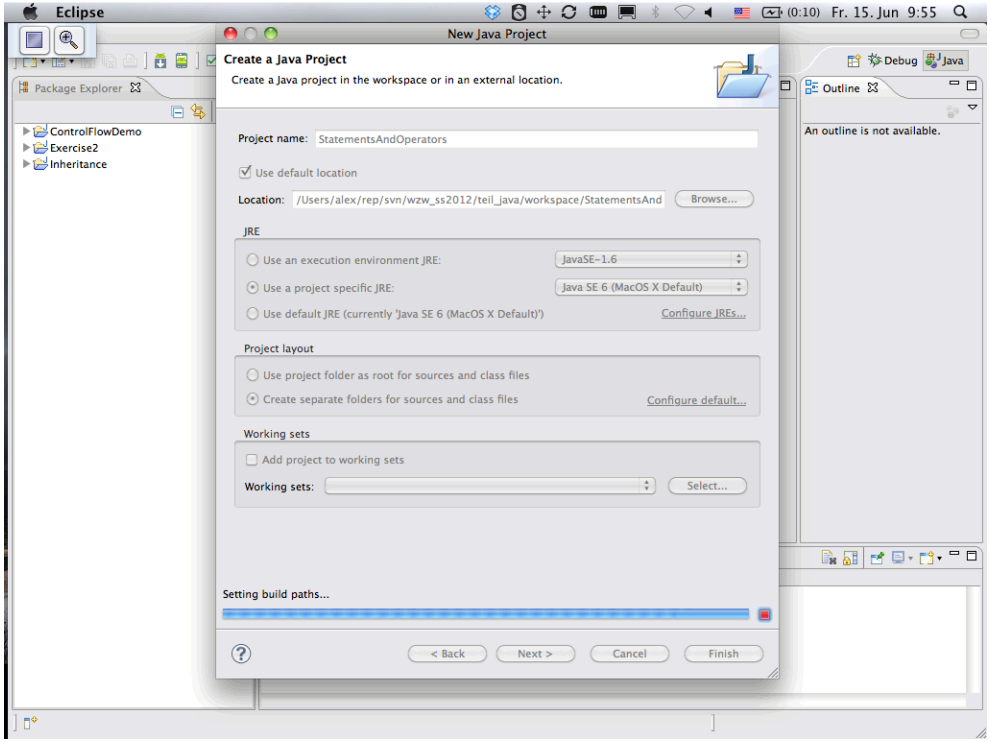
3 Classes, Objects, Inheritance

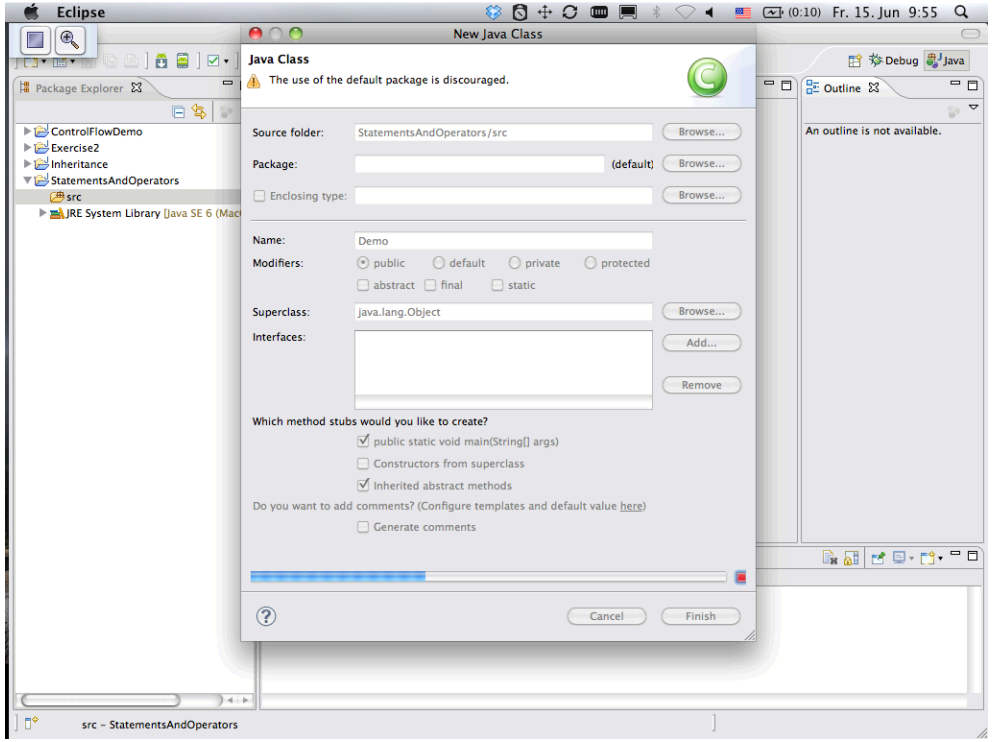
Deepening readings:

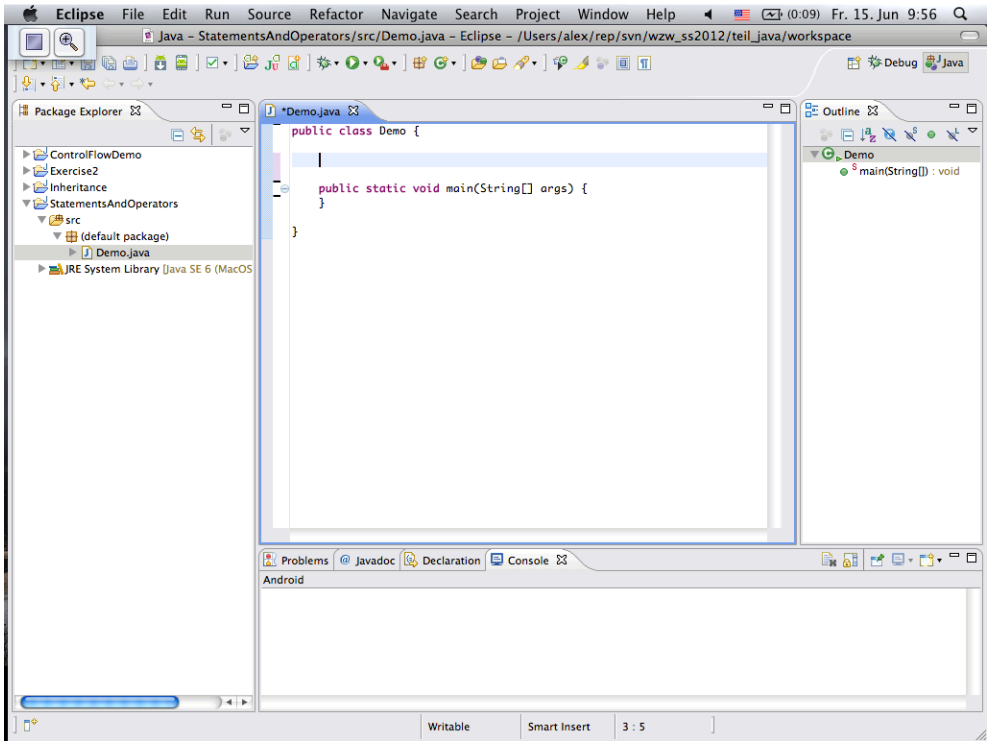
http://java.sun.com/docs/books/tutorial/java/javaO

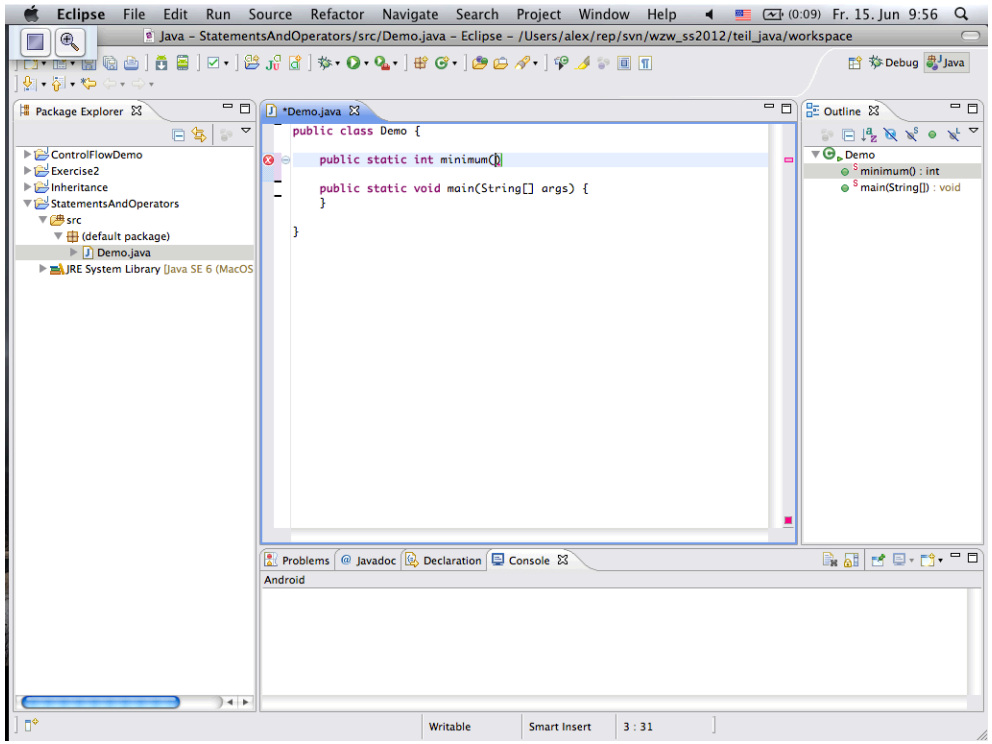


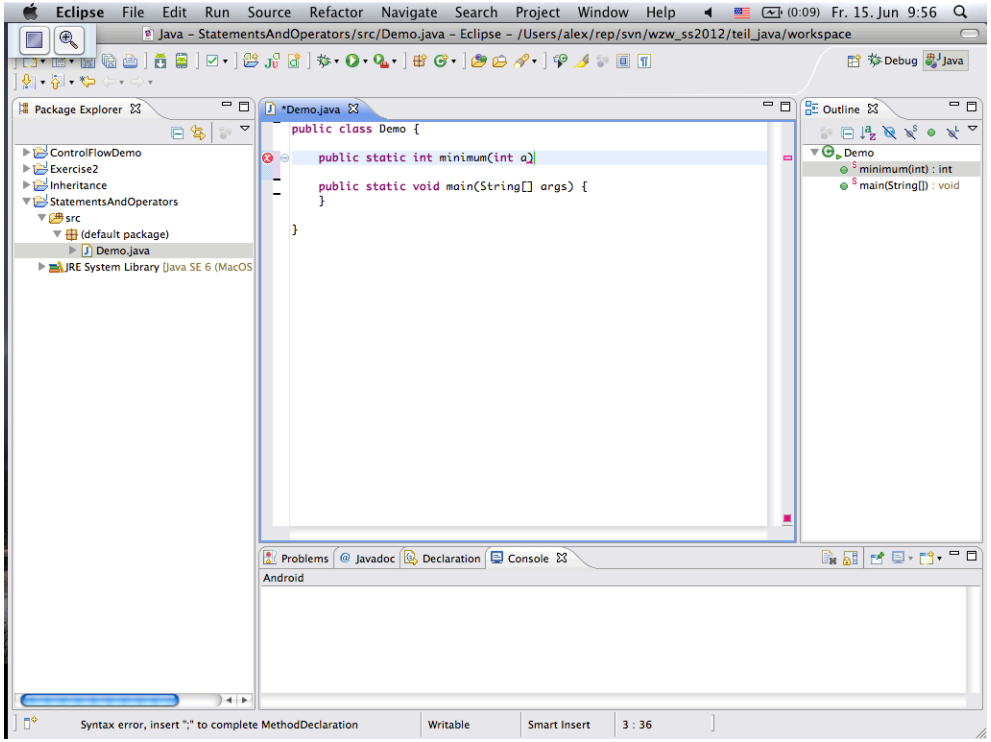


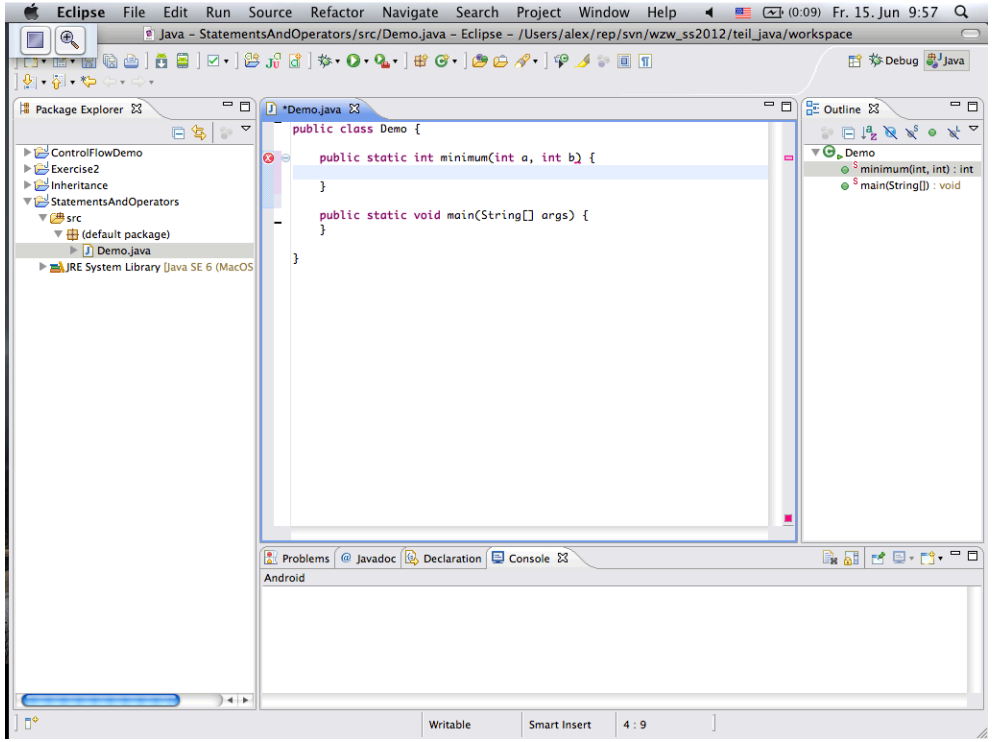

Eclipse
New Java Project
Create a Java Project
Create a Java project in the workspace or in an external location.
Project name: StatementsAndOperators
Use default location
Location: /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace/StatementsAnd
Browse...
JRE
Use an execution environment JRE: JavaSE-1.6
Use a project specific JRE: Java SE 6 (MacOS X Default)
Use default JRE (currently 'Java SE 6 (MacOS X Default)')
Configure JREs...
Project layout
Use project folder as root for sources and class files
Create separate folders for sources and class files
Configure default...
Working sets
Add project to working sets
Working sets:
Select...
Setting build paths...
< Back
Next >
Cancel
Finish
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
Outline
An outline is not available.
Debug
Java

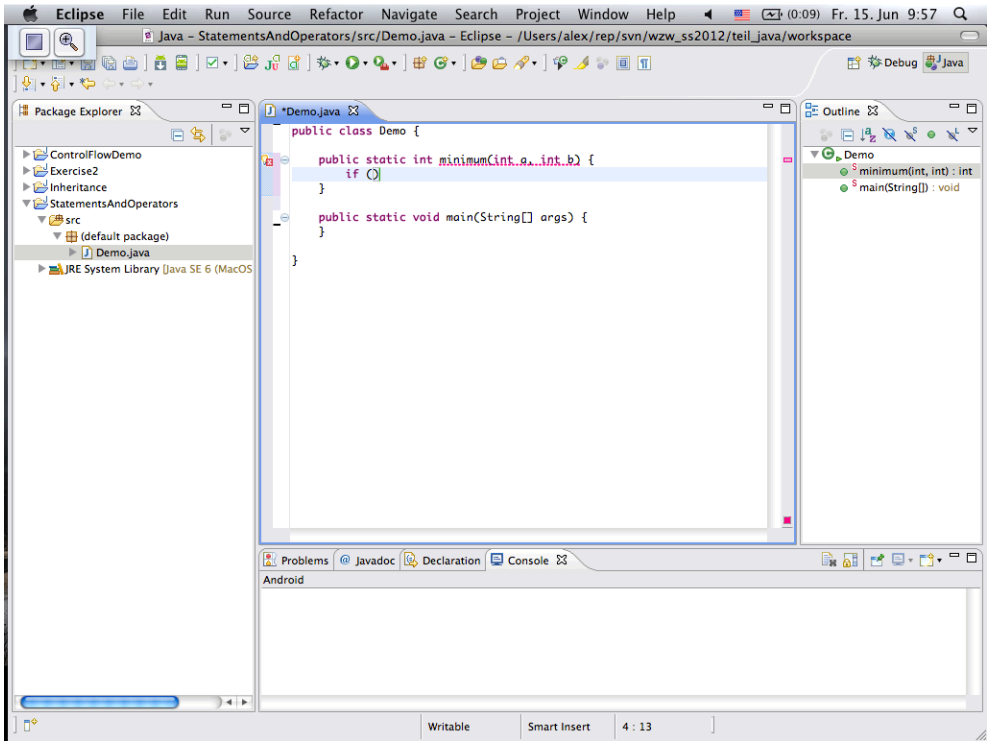

Eclipse
New Java Class
Java Class
The use of the default package is discouraged.
Source folder: StatementsAndOperators/src
Package: (default)
Enclosing type:
Name: Demo
Modifiers: public default private protected
abstract final static
Superclass: java.lang.Object
Interfaces:
Add...
Remove
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Cancel
Finish
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
JRE System Library [Java SE 6 (Mac
Outline
An outline is not available.
Debug
Java
src – StatementsAndOperators

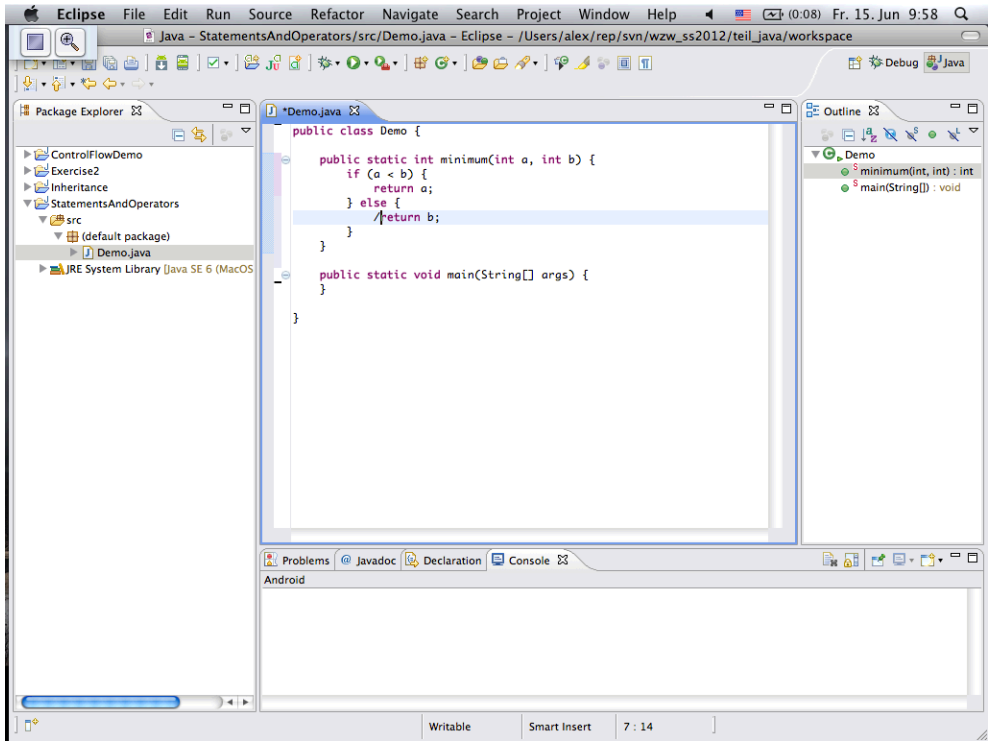

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
*Demo.java
public class Demo {
public static void main(String[] args) {
}
}
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
Outline
Demo
main(String[]) : void
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
3 : 5
Debug
Java

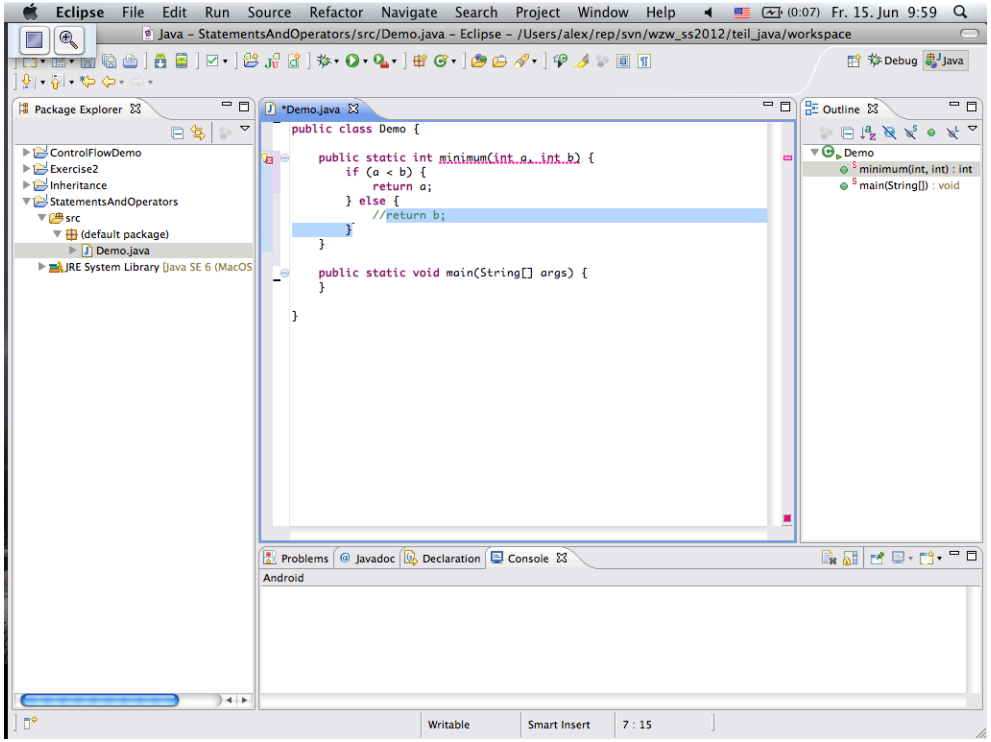

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
*Demo.java
public class Demo {
public static int minimum()
public static void main(String[] args) {
}
}
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
Outline
Demo
minimum() : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
3 : 31
Debug
Java

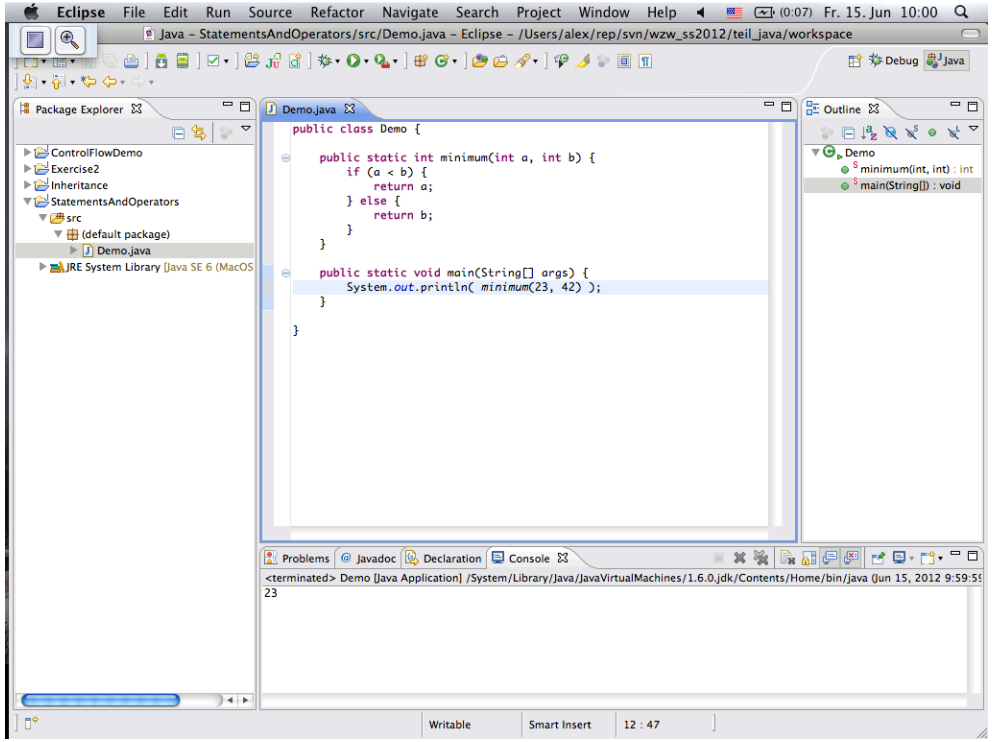

Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
*Demo.java
public class Demo {
public static int minimum(int a)
public static void main(String[] args) {
}
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
Outline
Demo
minimum(int) : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
Android
Syntax error, insert ";" to complete MethodDeclaration
Writable
Smart Insert
3 : 36

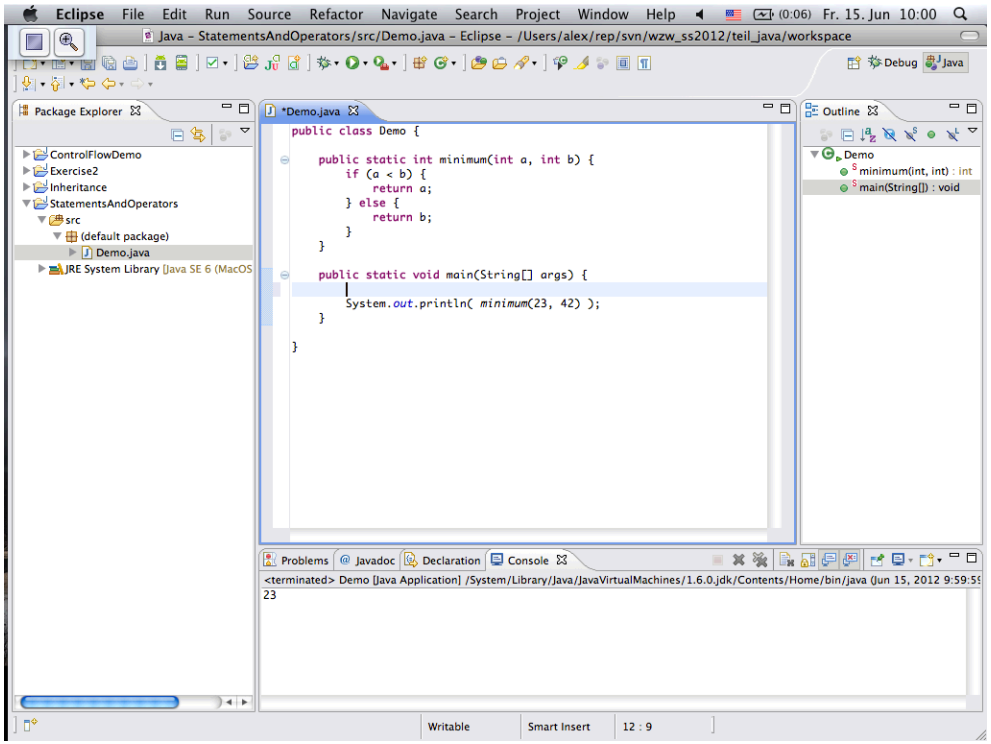

Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
}
public static void main(String[] args) {
}
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
Outline
Demo
minimum(int, int) : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
4 : 9

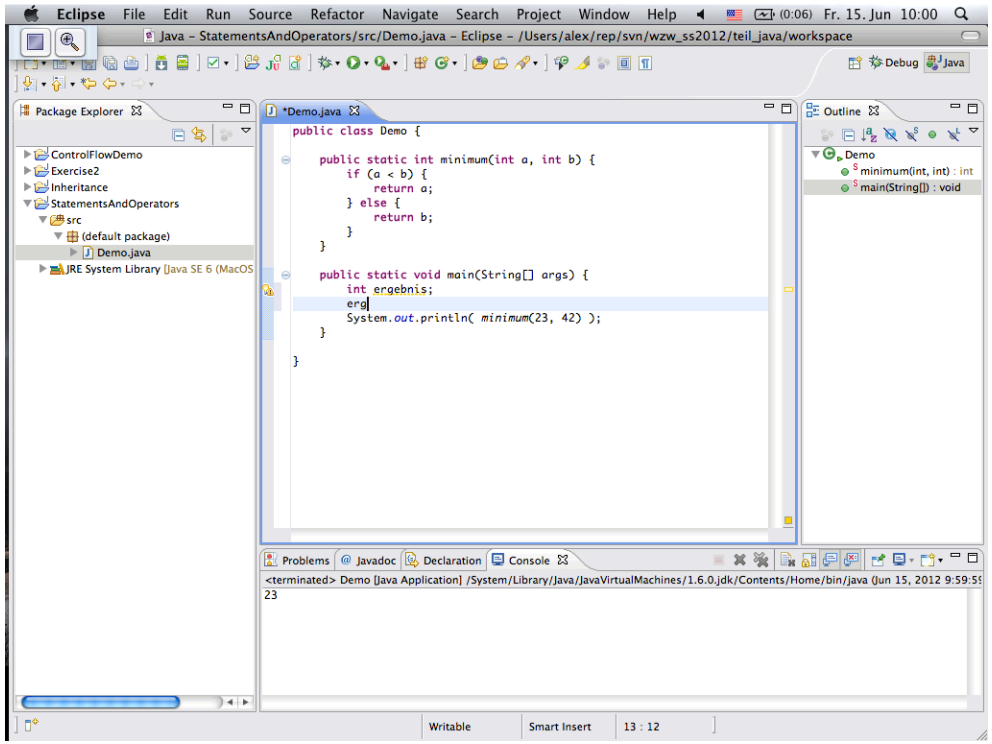

Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if ()
}
public static void main(String[] args) {
}
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
Outline
Demo
minimum(int, int) : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
4 : 13

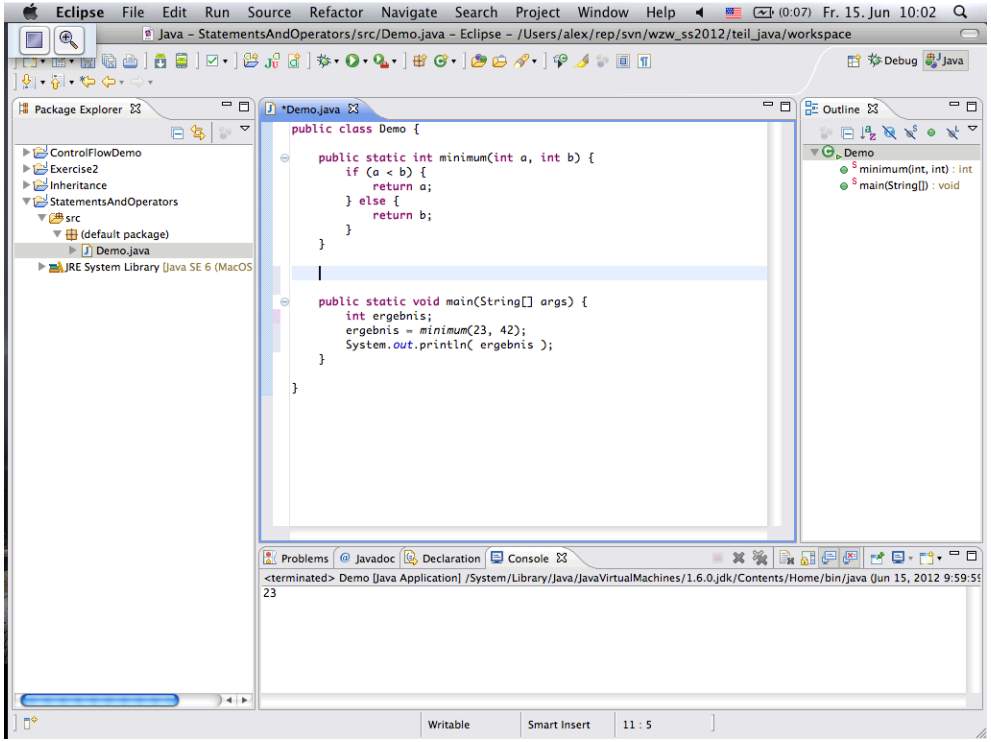

Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static void main(String[] args) {
}
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
Outline
Demo
minimum(int, int) : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
7 : 14

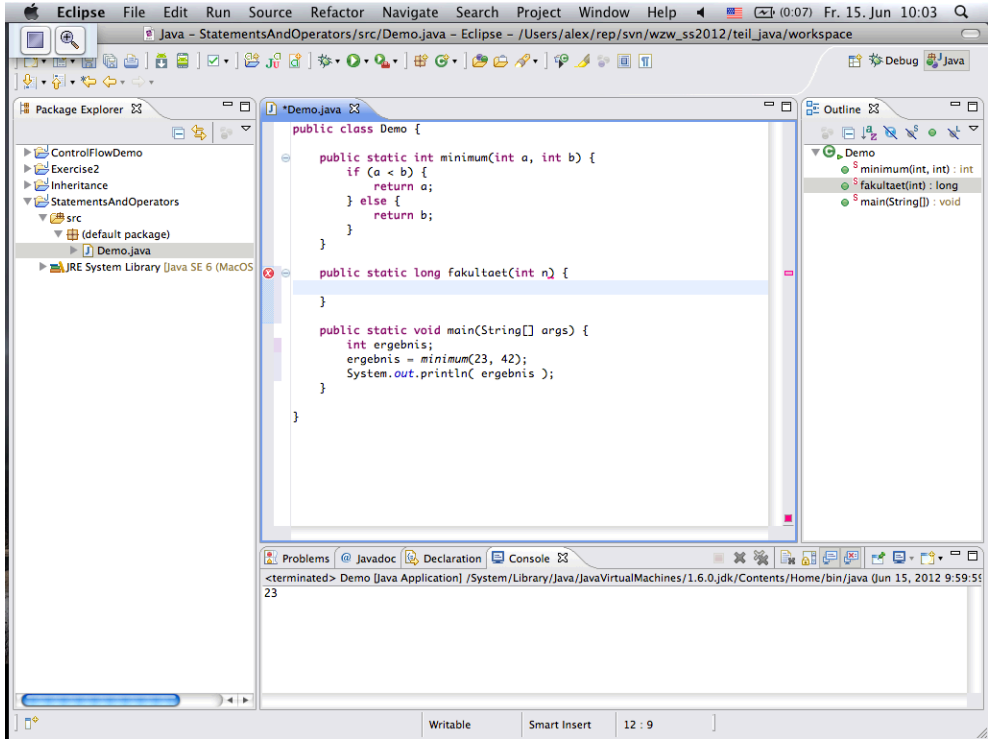

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
//return b;
}
}
public static void main(String[] args) {
}
}
Outline
Demo
minimum(int, int) : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
7 : 15

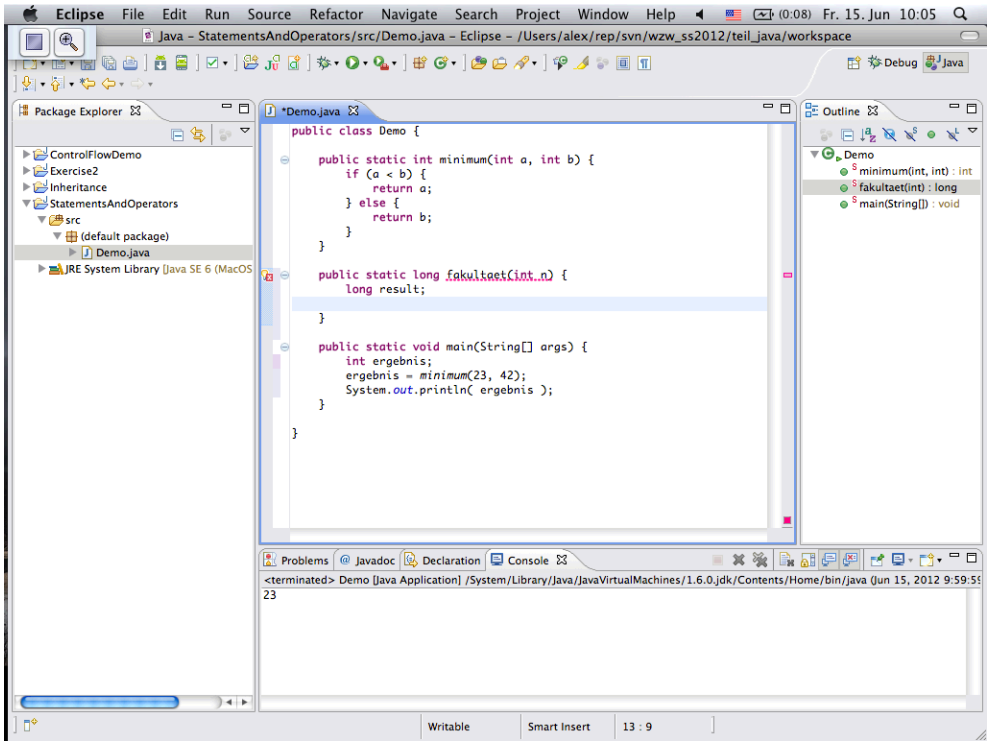

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static void main(String[] args) {
System.out.println(minimum(23, 42));
}
}
Outline
Demo
minimum(int, int) : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:5
23
Writable
Smart Insert
12 : 47

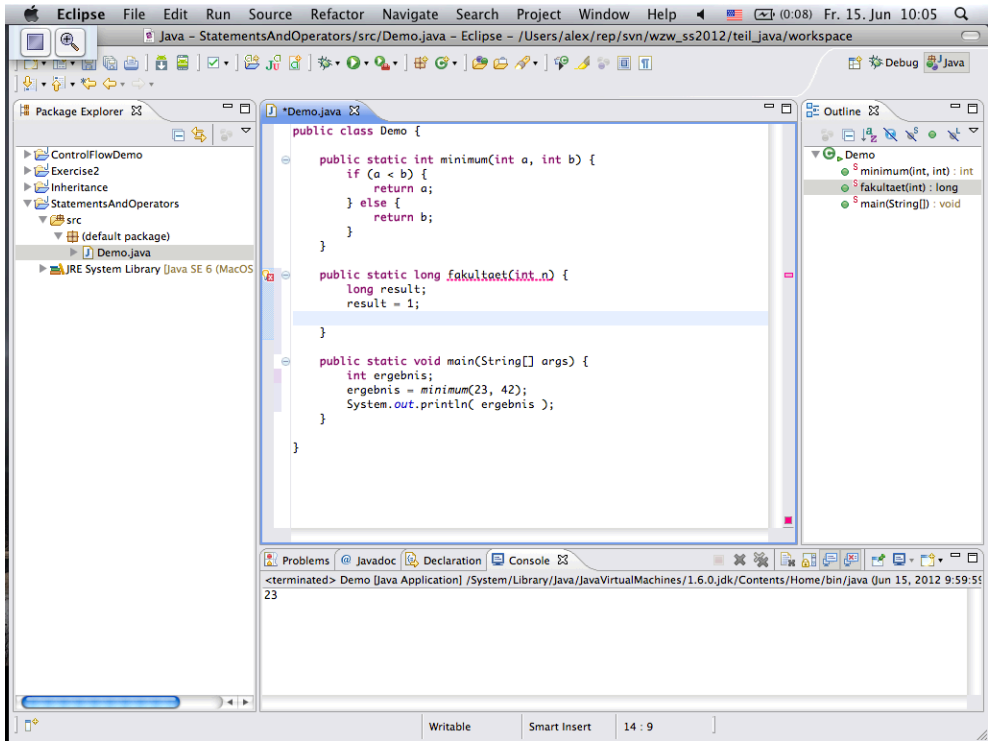

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static void main(String[] args) {
System.out.println(minimum(23, 42));
}
}
Outline
Demo
minimum(int, int) : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:5
23
Writable
Smart Insert
12 : 9

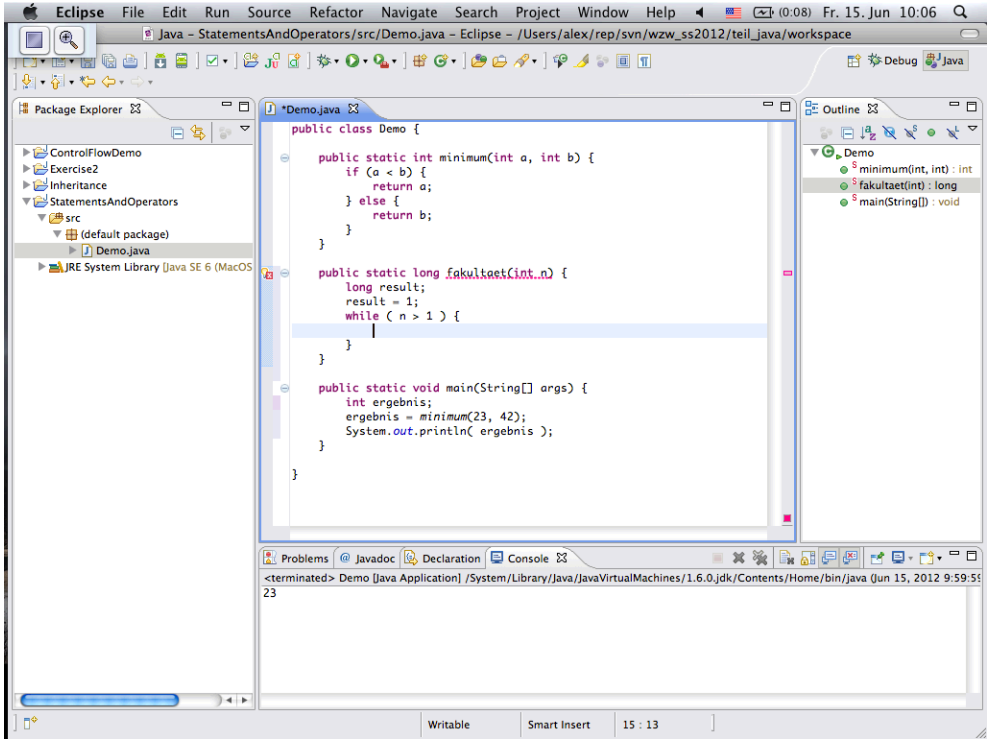

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static void main(String[] args) {
int ergebnis;
erg
System.out.println(minimum(23, 42));
}
}
Outline
Demo
minimum(int, int) : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:5
23
Writable
Smart Insert
13 : 12

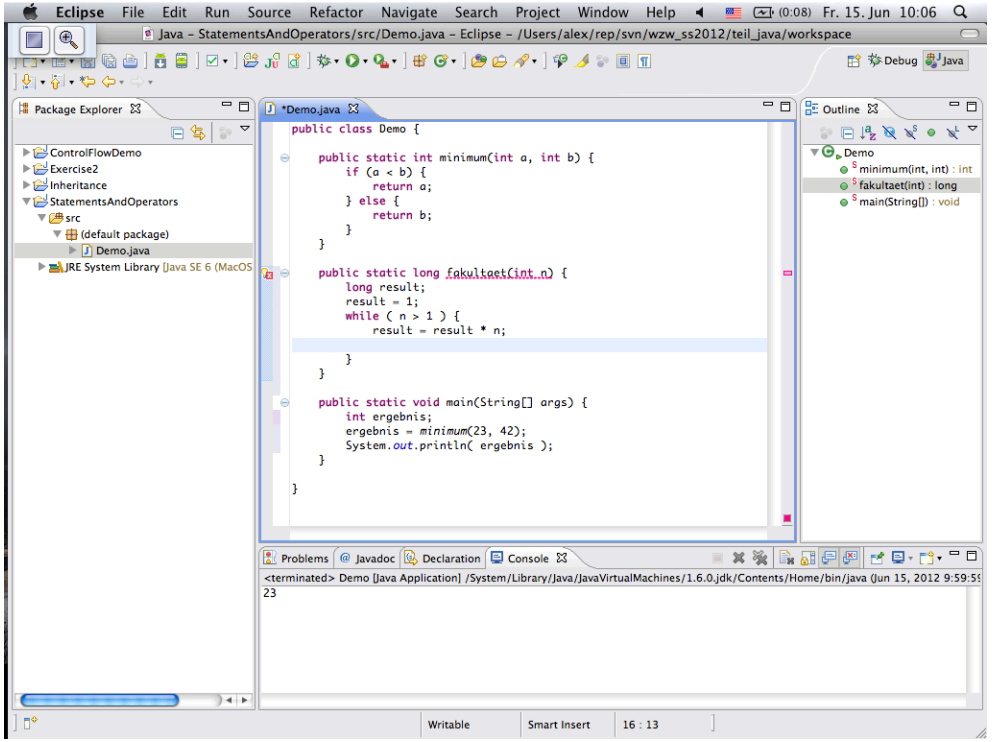

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 15. Jun 10:02
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
}
}
Outline
Demo
minimum(int, int) : int
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:5
23
Writable Smart Insert 11 : 5

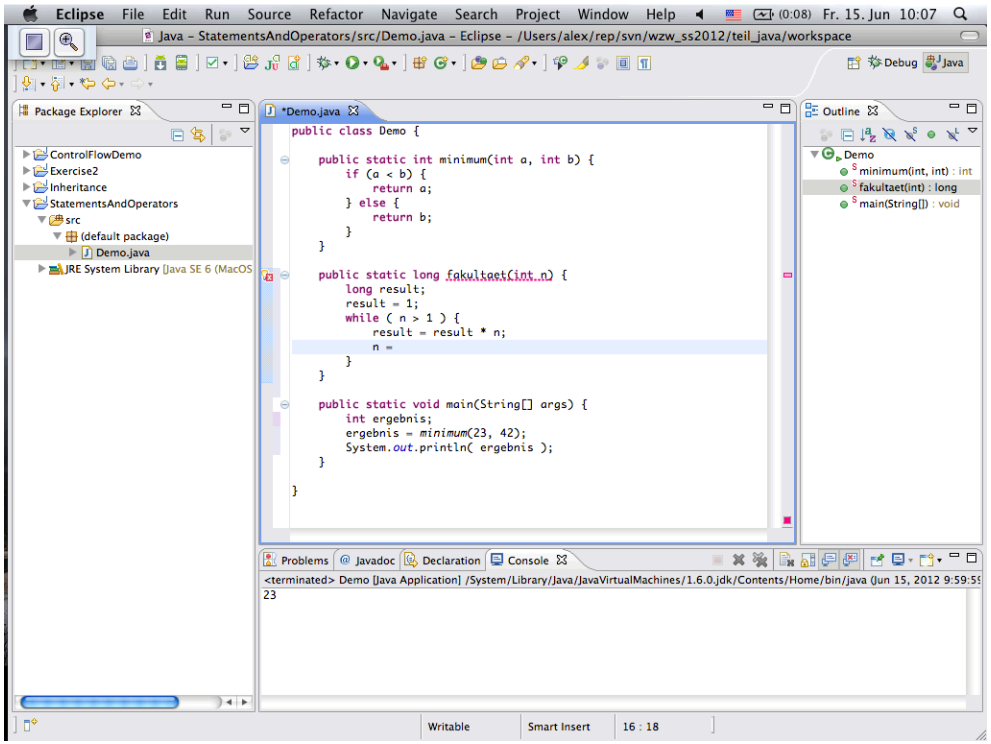

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 15. Jun 10:03
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
}
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:5
23
Writable Smart Insert 12 : 9

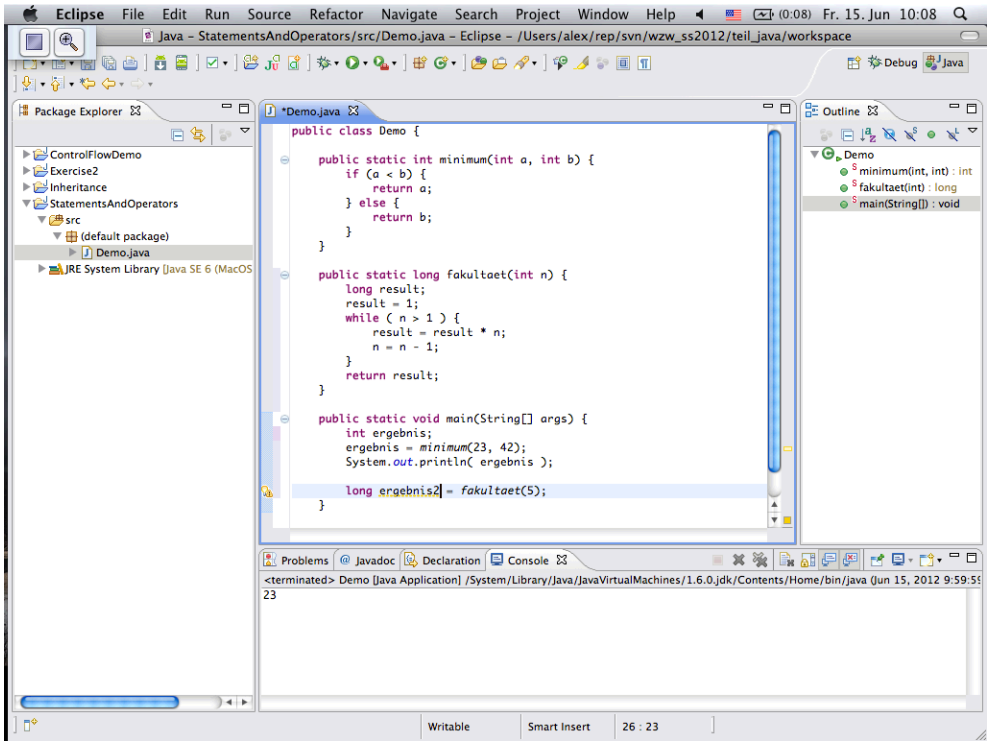

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 15. Jun 10:05
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
}
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:5
23
Writable Smart Insert 13 : 9

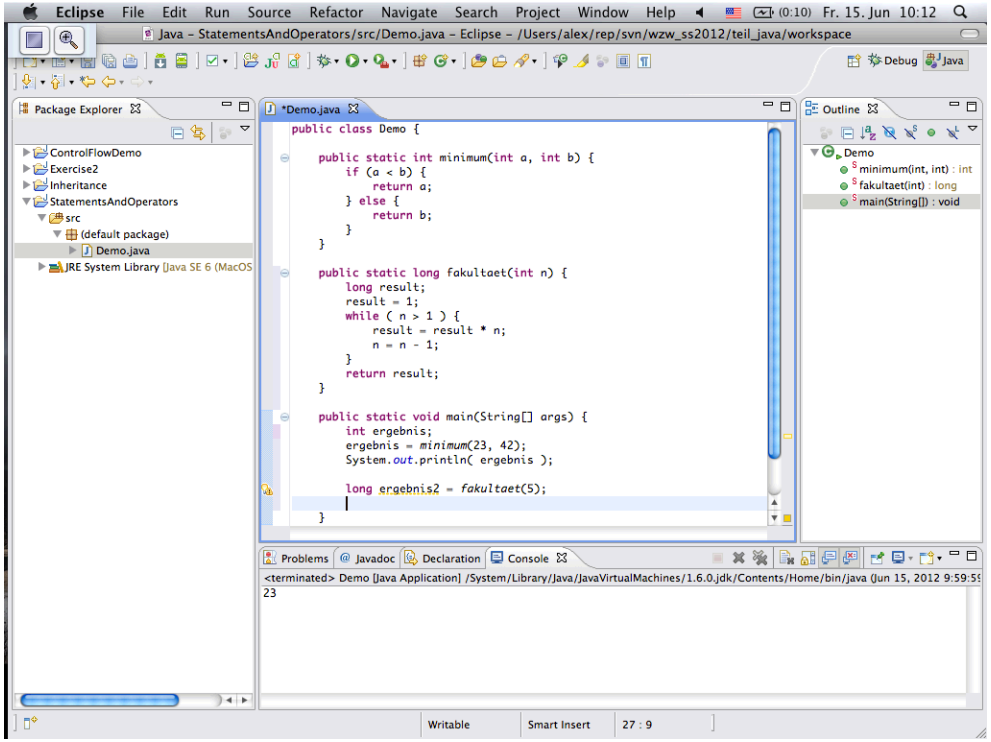

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 15. Jun 10:05
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
}
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:5
23
Writable Smart Insert 14 : 9

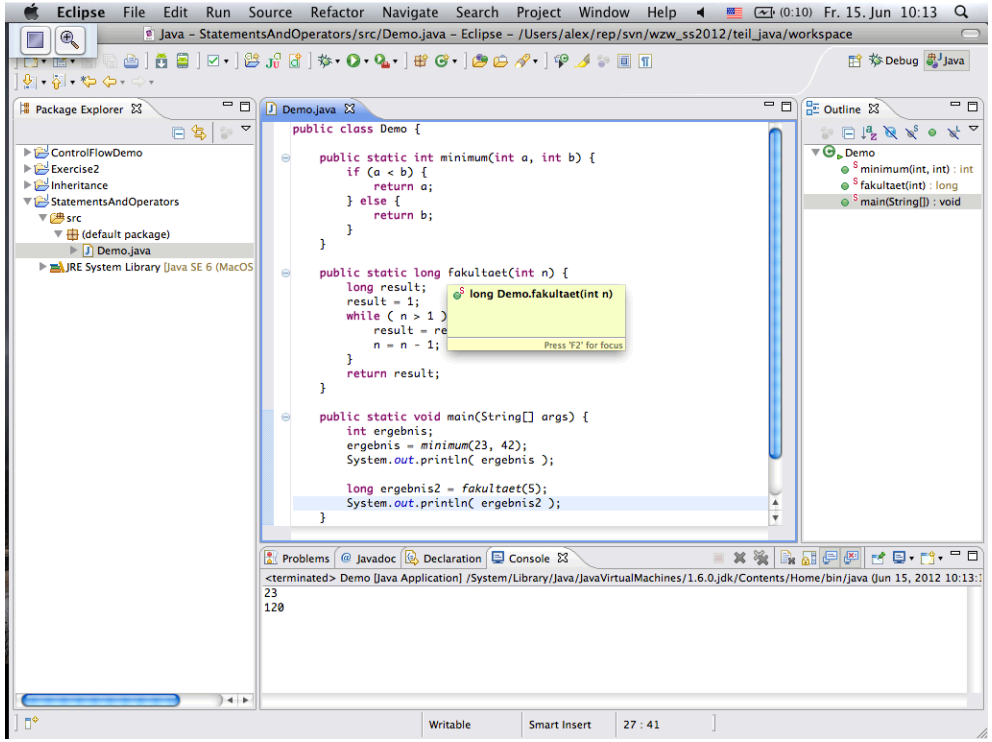

Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
}
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
}
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Problems
Javadoc
Declaration
Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:55
23
Writable
Smart Insert
15 : 13

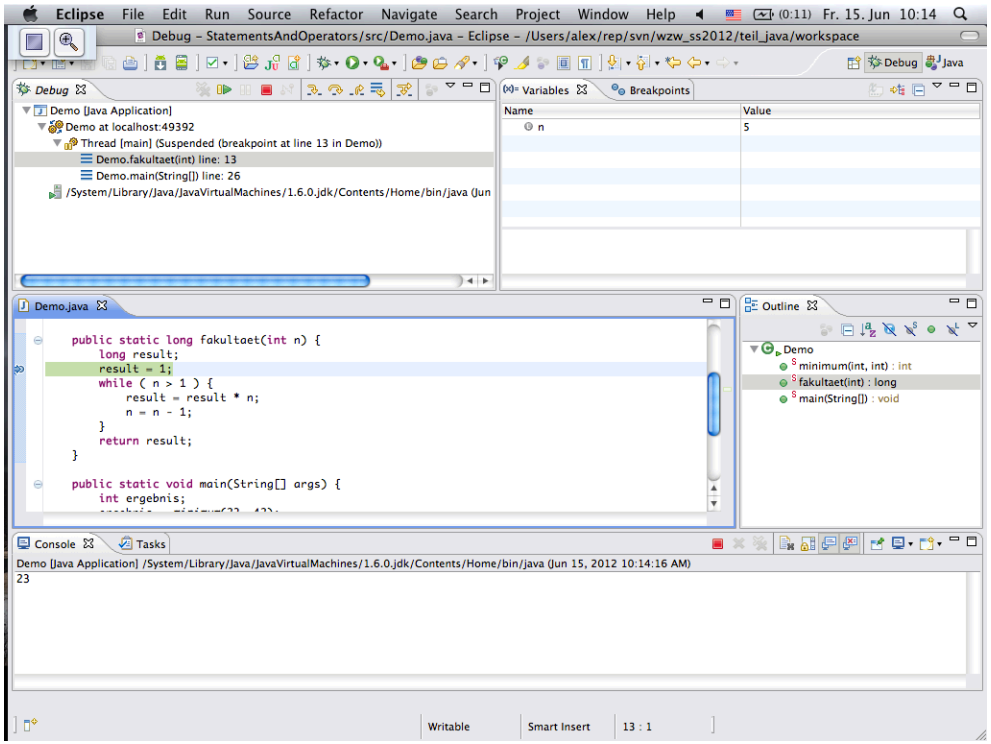

Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
}
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
}
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Problems
Javadoc
Declaration
Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:55
23
Writable
Smart Insert
16 : 13

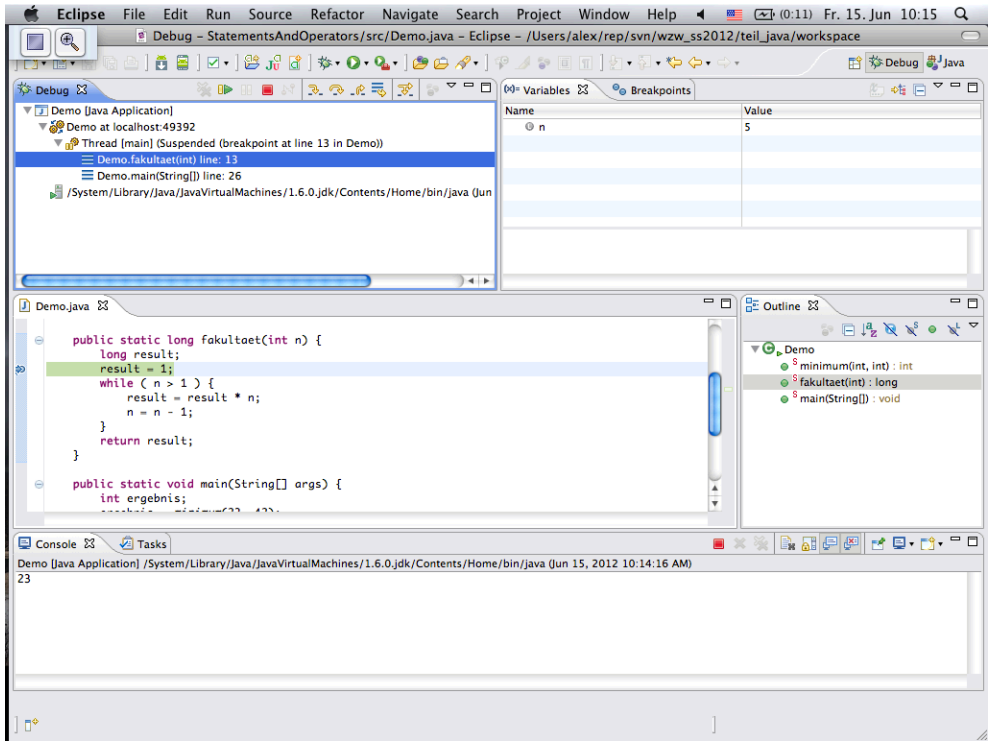

Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n =
}
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
}
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Problems
Javadoc
Declaration
Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:55
23
Writable
Smart Insert
16 : 18

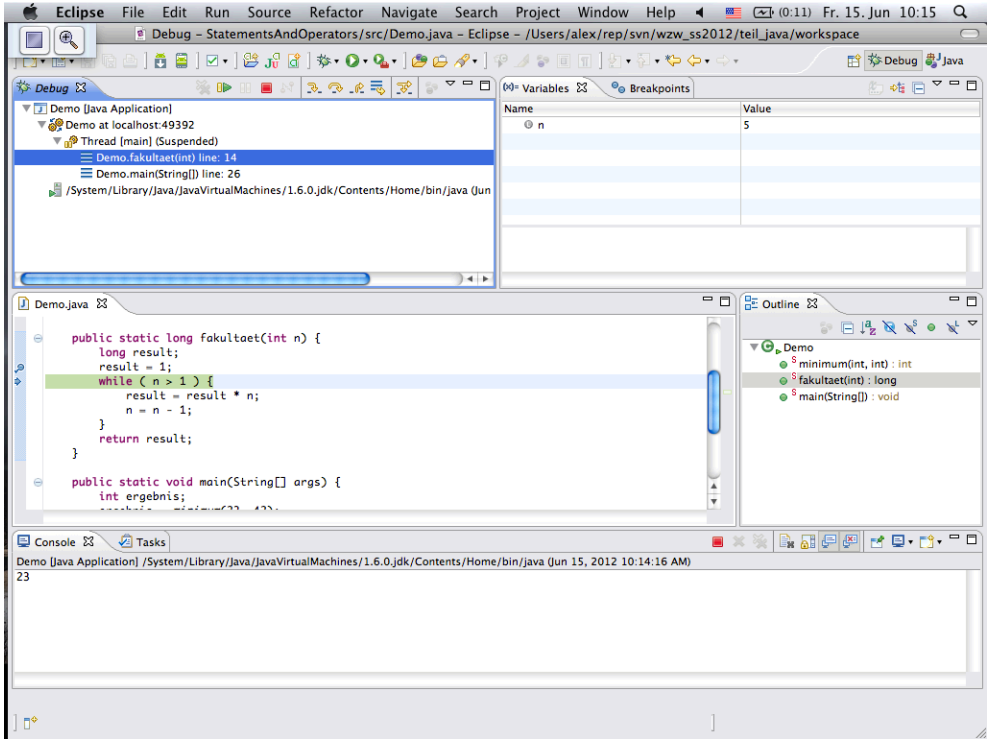

Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
long ergebnis2 = fakultaet(5);
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Problems
Javadoc
Declaration
Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:55
23
Writable
Smart Insert
26 : 23

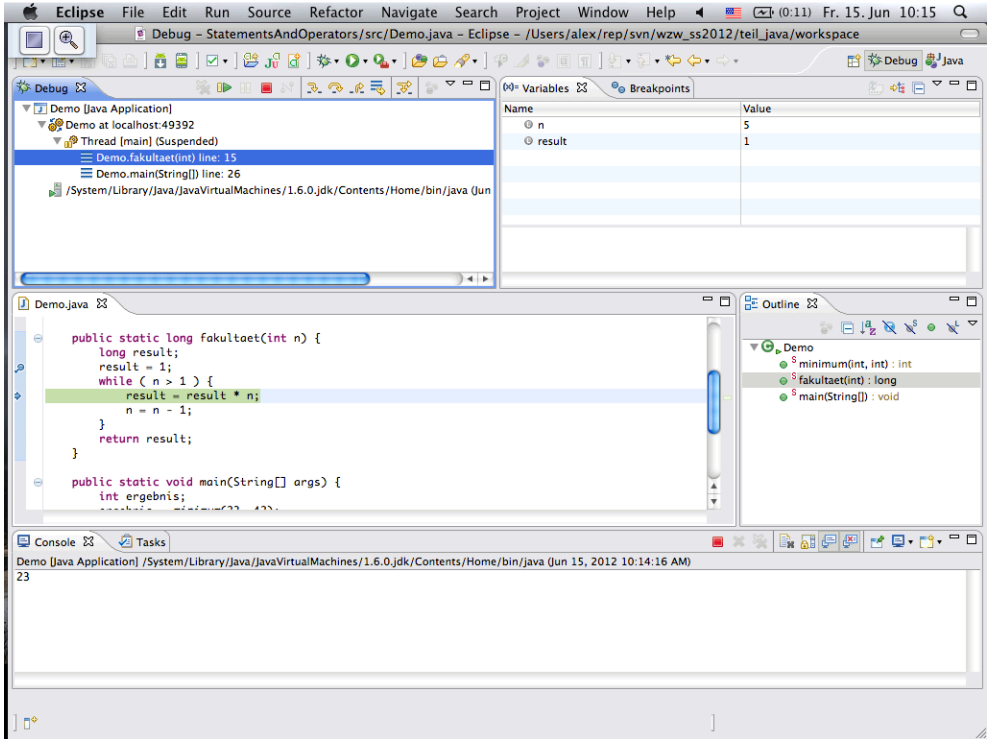

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 15. Jun 10:12
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
long ergebnis2 = fakultaet(5);
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 9:59:55
23
Writable Smart Insert 27 : 9

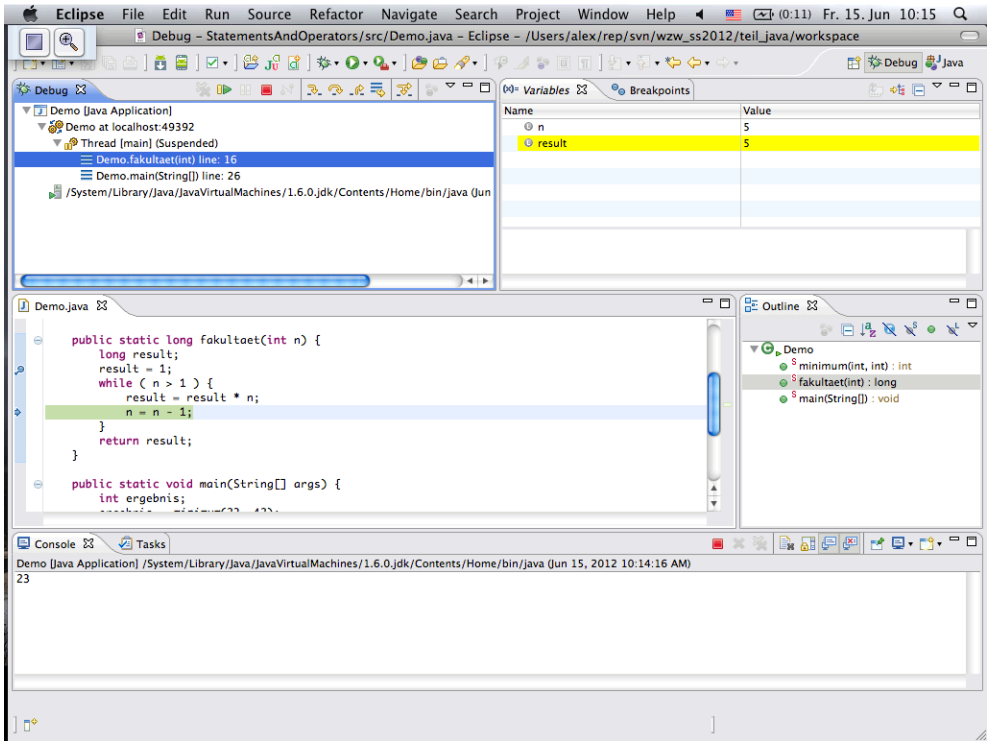

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 15. Jun 10:13
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
Demo.java
public class Demo {
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1)
result = re
n = n - 1;
}
return result;
}
long Demo.fakultaet(int n)
Press 'F2' for focus
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
long ergebnis2 = fakultaet(5);
System.out.println(ergebnis2);
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:13:
23
120
Writable Smart Insert 27 : 41

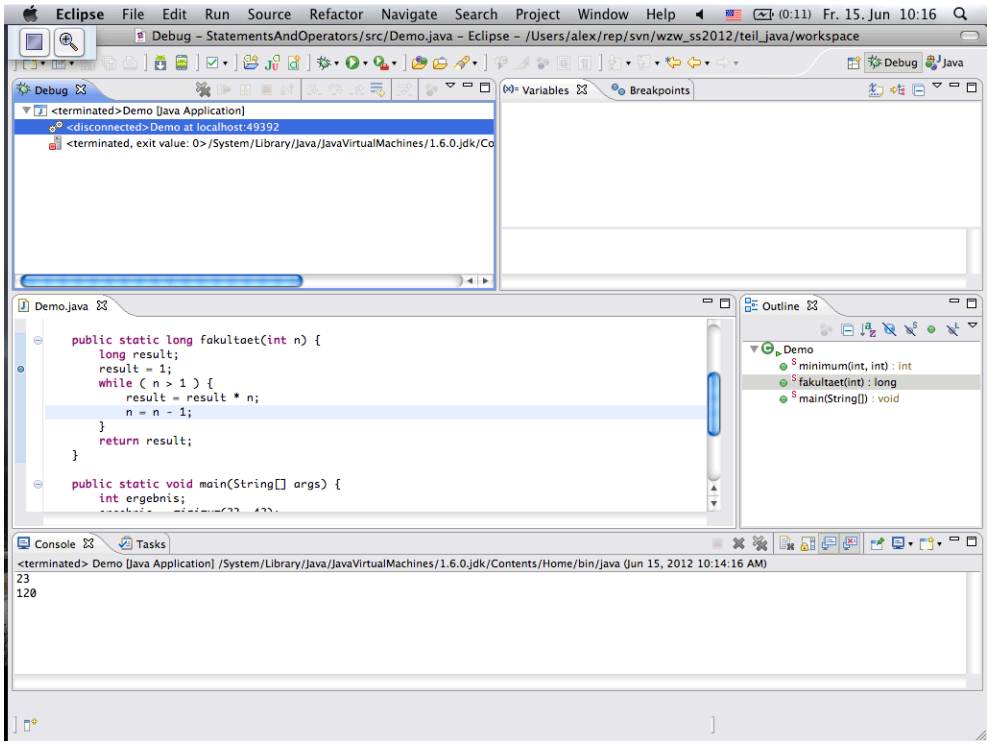

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 15. Jun 10:14
Debug – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Demo [Java Application]
Demo at localhost:49392
Thread [main] (Suspended (breakpoint at line 13 in Demo))
Demo.fakultaet(int) line: 13
Demo.main(String[]) line: 26
/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
Variables
Breakpoints
Name
Value
n
5
Demo.java
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static void main(String[] args) {
int ergebnis;
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Console
Tasks
Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:16 AM)
23
Writable Smart Insert 13 : 1

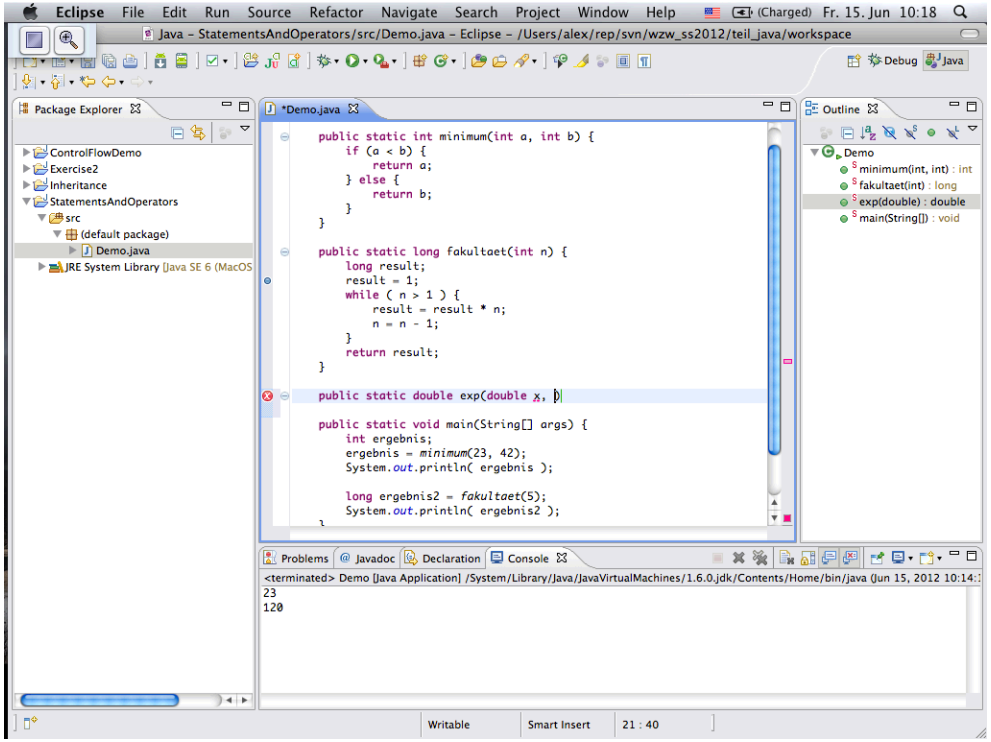

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help Fr. 15. Jun 10:15
Debug – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Demo [Java Application]
Demo at localhost:49392
Thread [main] (Suspended (breakpoint at line 13 in Demo))
Demo.fakultaet(int) line: 13
Demo.main(String[]) line: 26
/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
Variables
Breakpoints
Name
Value
n
5
Demo.java
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static void main(String[] args) {
int ergebnis;
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Console
Tasks
Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:16 AM)
23

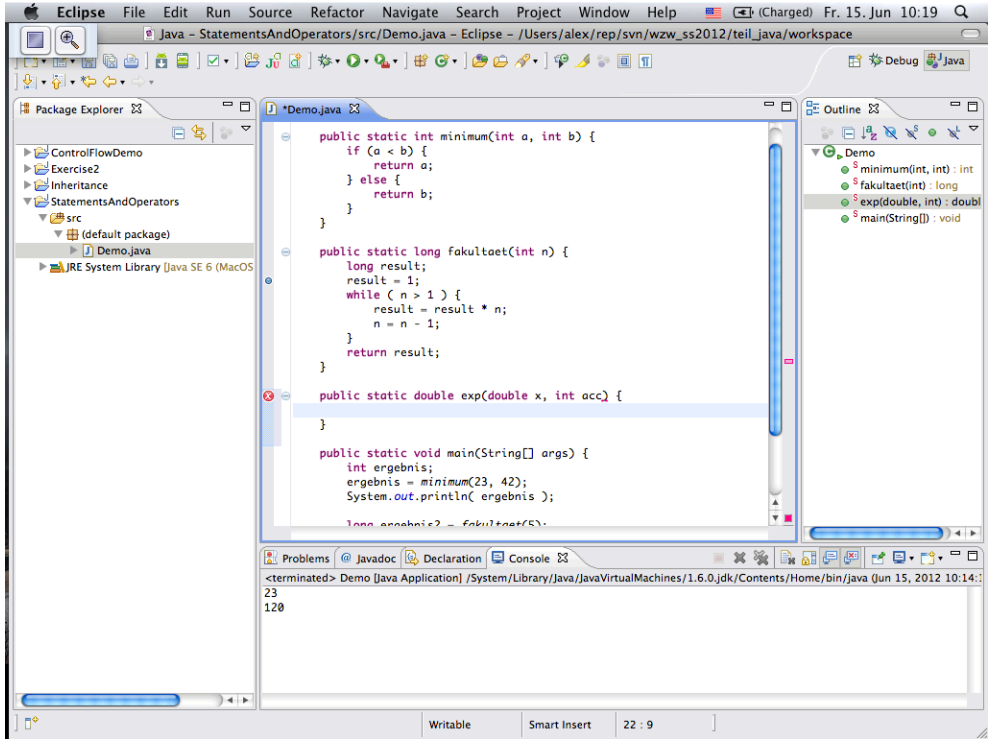

Debug – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Demo [Java Application]
Demo at localhost:49392
Thread [main] (Suspended)
Demo.fakultaet(int) line: 14
Demo.main(String[]) line: 26
/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
Name
Value
n
5
Demo.java
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static void main(String[] args) {
int ergebnis;
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Console
Tasks
Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:16 AM)
23

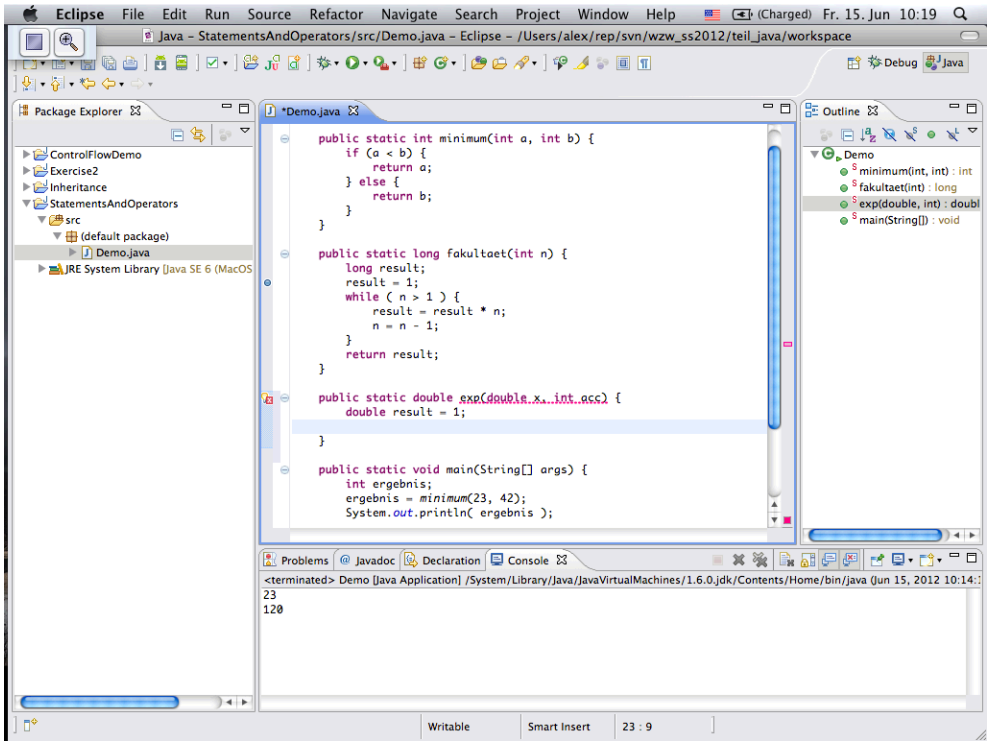

Debug – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Demo [Java Application]
Demo at localhost:49392
Thread [main] (Suspended)
Demo.fakultaet(int) line: 15
Demo.main(String[]) line: 26
/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
Name
Value
n
5
result
1
Demo.java
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static void main(String[] args) {
int ergebnis;
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Console
Tasks
Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:16 AM)
23

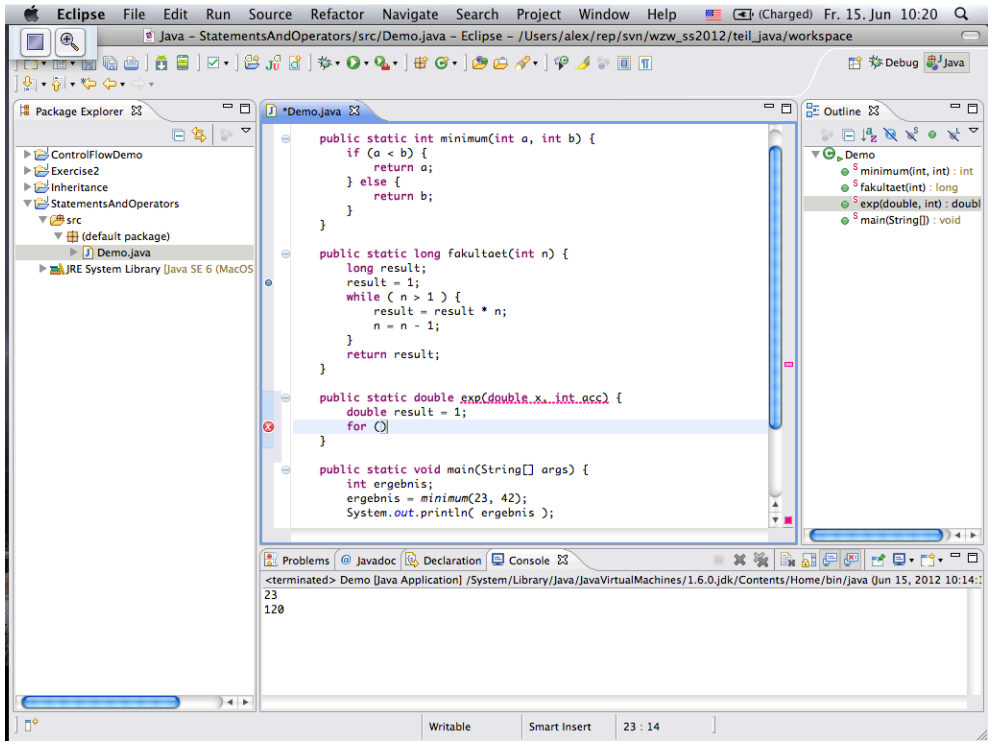

Debug – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Demo [Java Application]
Demo at localhost:49392
Thread [main] (Suspended)
Demo.fakultaet(int) line: 16
Demo.main(String[]) line: 26
/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
Name
Value
n
5
result
5
Demo.java
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static void main(String[] args) {
int ergebnis;
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Console
Tasks
Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:16 AM)
23

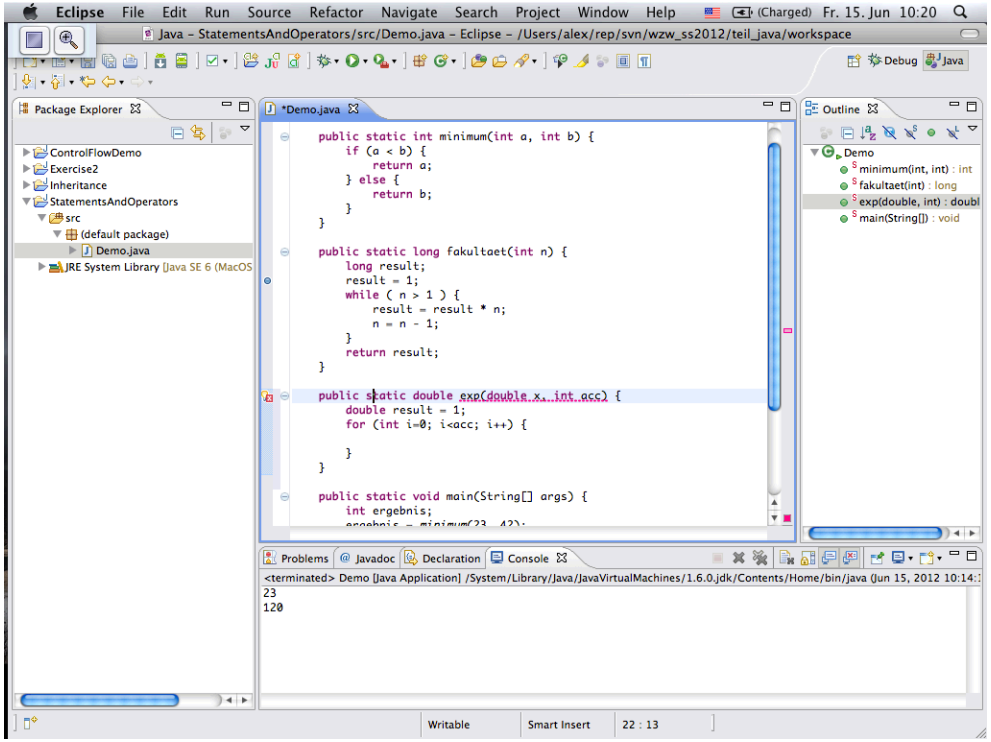

Debug – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
<terminated>Demo [Java Application]
<disconnected>Demo at localhost:49392
<terminated, exit value: 0> /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Co
Demo.java
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static void main(String[] args) {
int ergebnis;
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
main(String[]) : void
Console
Tasks
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:16 AM)
23
120

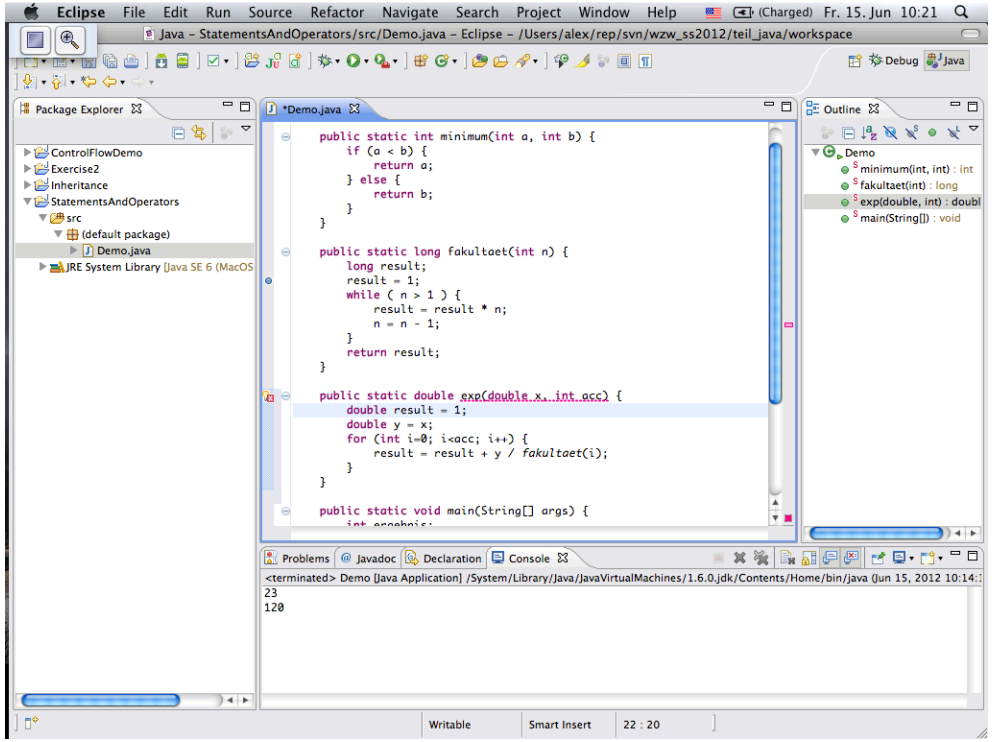

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:18
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x,)
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
long ergebnis2 = fakultaet(5);
System.out.println(ergebnis2);
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double) : double
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:
23
120
Writable Smart Insert 21 : 40

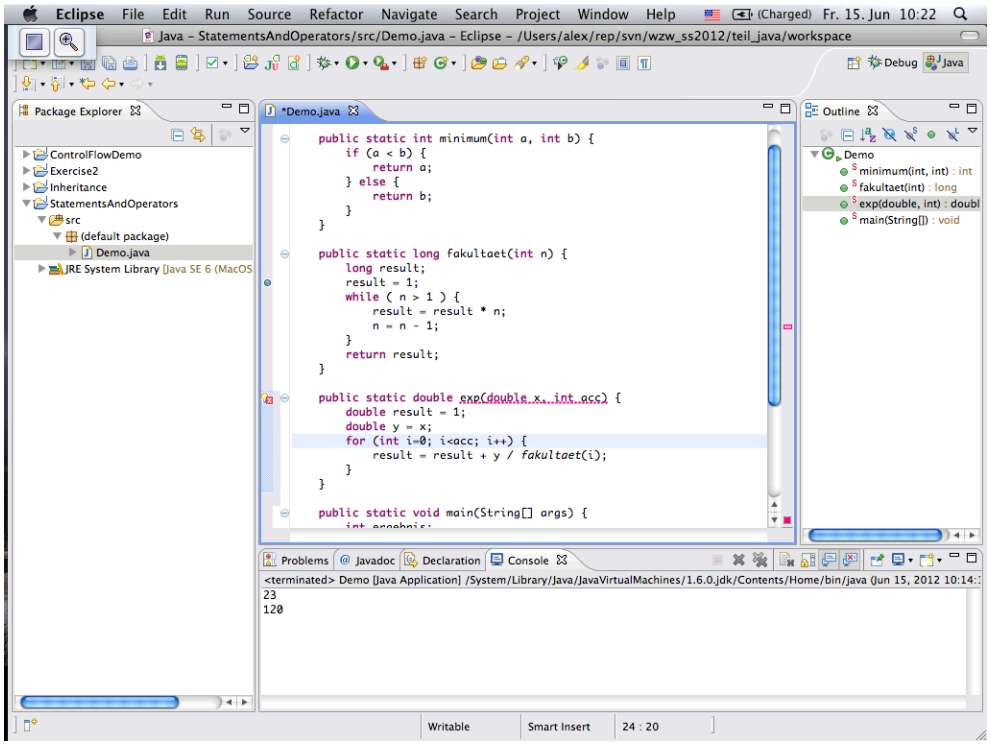

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:19
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:
23
120
Writable Smart Insert 22 : 9

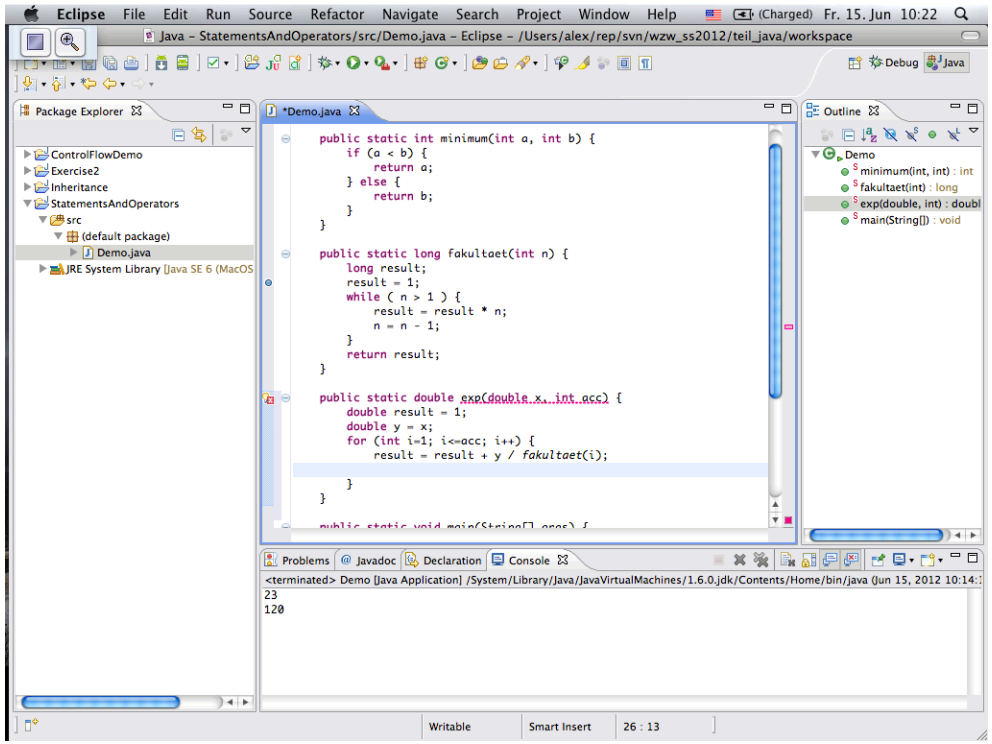

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:19
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
double result = 1;
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:
23
120
Writable Smart Insert 23 : 9

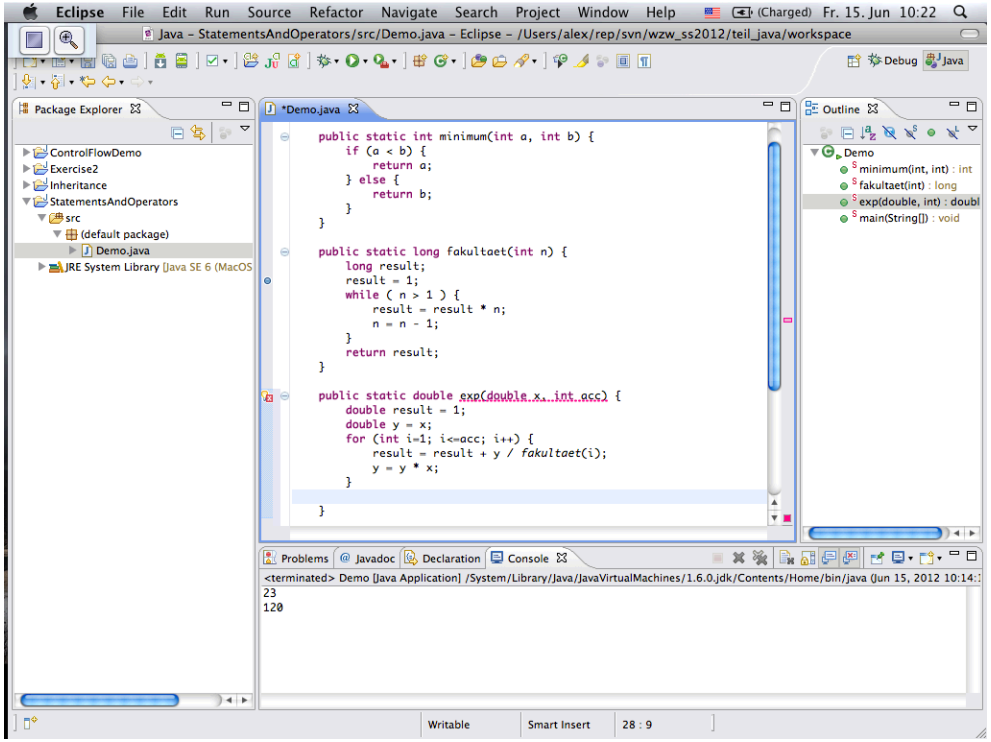

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:20
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
double result = 1;
for ()
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:
23
120
Writable Smart Insert 23 : 14

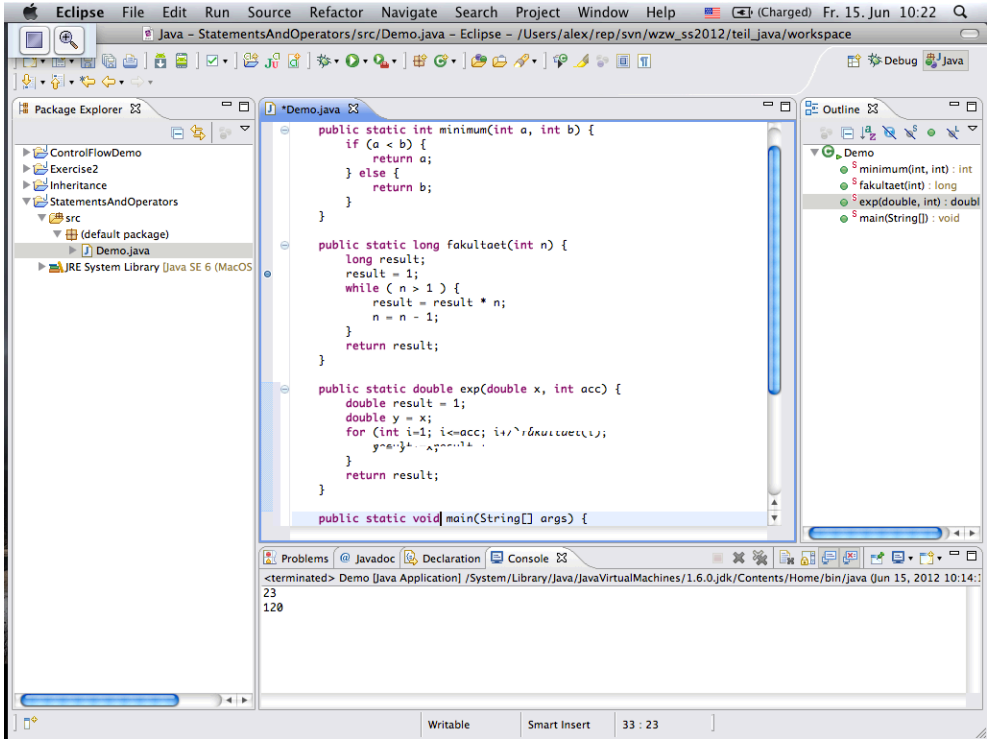

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:20
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
double result = 1;
for (int i=0; i<acc; i++) {
}
}
public static void main(String[] args) {
int ergebnis;
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:
23
120
Writable Smart Insert 22 : 13

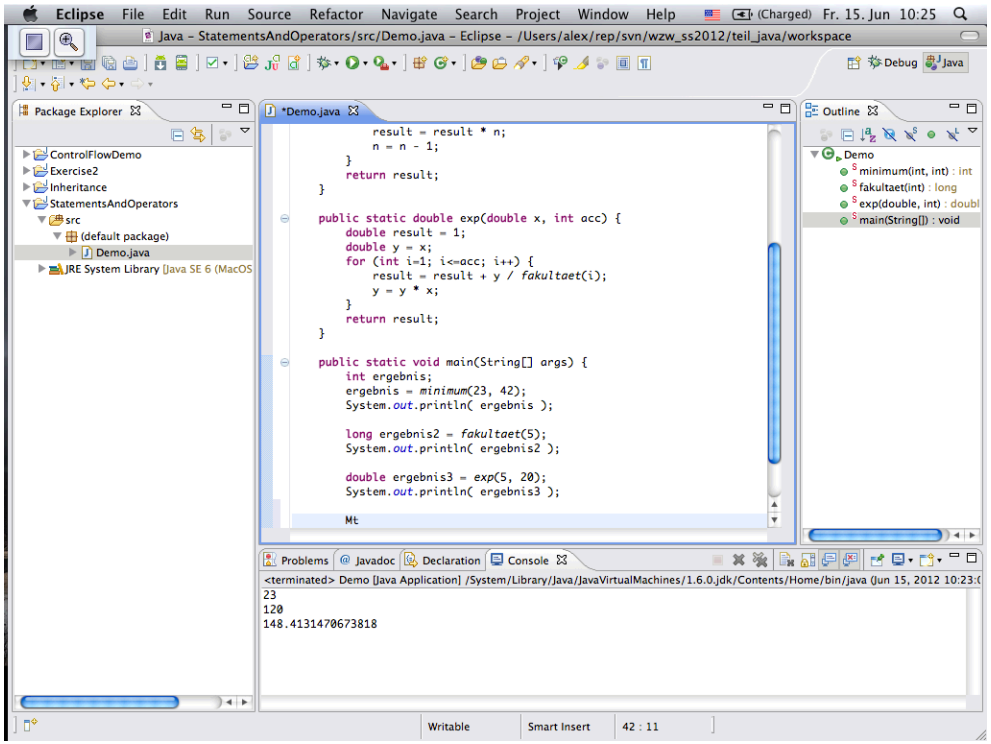

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:21
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
double result = 1;
double y = x;
for (int i=0; i<acc; i++) {
result = result + y / fakultaet(i);
}
}
public static void main(String[] args) {
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:
23
120
Writable Smart Insert 22 : 20

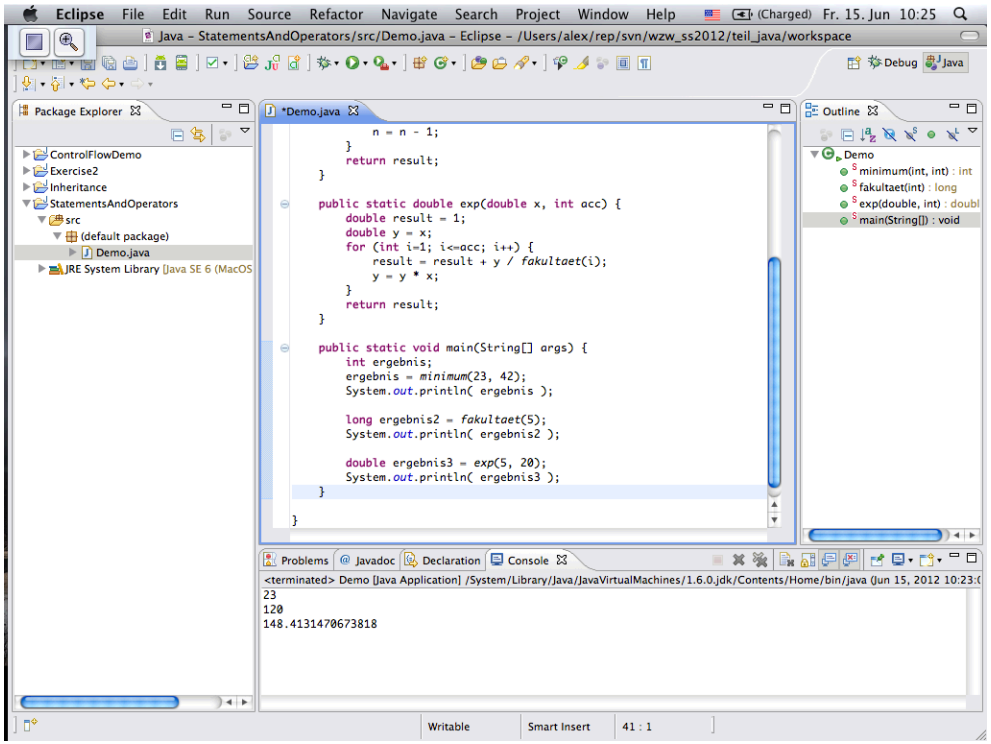

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:22
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
double result = 1;
double y = x;
for (int i=0; i<acc; i++) {
result = result + y / fakultaet(i);
}
}
public static void main(String[] args) {
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:
23
120
Writable Smart Insert 24 : 20


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:22
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
double result = 1;
double y = x;
for (int i=1; i<=acc; i++) {
result = result + y / fakultaet(i);
}
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:
23
120
Writable Smart Insert 26 : 13


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:22
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
double result = 1;
double y = x;
for (int i=1; i<=acc; i++) {
result = result + y / fakultaet(i);
y = y * x;
}
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:
23
120
Writable Smart Insert 28 : 9


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:22
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
public static int minimum(int a, int b) {
if (a < b) {
return a;
} else {
return b;
}
}
public static long fakultaet(int n) {
long result;
result = 1;
while (n > 1) {
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
double result = 1;
double y = x;
for (int i=1; i<=acc; i++) {
}
return result;
}
public static void main(String[] args) {
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:14:
23
120
Writable Smart Insert 33 : 23


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:25
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
result = result * n;
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
double result = 1;
double y = x;
for (int i=1; i<=acc; i++) {
result = result + y / fakultaet(i);
y = y * x;
}
return result;
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
long ergebnis2 = fakultaet(5);
System.out.println(ergebnis2);
double ergebnis3 = exp(5, 20);
System.out.println(ergebnis3);
Mt
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:23:
23
120
148.4131470673818
Writable Smart Insert 42 : 11


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 15. Jun 10:25
Java – StatementsAndOperators/src/Demo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
src
(default package)
Demo.java
JRE System Library [Java SE 6 (MacOS
*Demo.java
n = n - 1;
}
return result;
}
public static double exp(double x, int acc) {
double result = 1;
double y = x;
for (int i=1; i<=acc; i++) {
result = result + y / fakultaet(i);
y = y * x;
}
return result;
}
public static void main(String[] args) {
int ergebnis;
ergebnis = minimum(23, 42);
System.out.println(ergebnis);
long ergebnis2 = fakultaet(5);
System.out.println(ergebnis2);
double ergebnis3 = exp(5, 20);
System.out.println(ergebnis3);
}
}
Outline
Demo
minimum(int, int) : int
fakultaet(int) : long
exp(double, int) : doubl
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 15, 2012 10:23:
23
120
148.4131470673818
Writable Smart Insert 41 : 1

3 Classes, Objects, Inheritance

Deepening readings:

http://java.sun.com/docs/books/tutorial/java/javaOO/classes.html
http://java.sun.com/docs/books/tutorial/java/javaOO/objects.html
http://java.sun.com/docs/books/tutorial/java/javaOO/more.html
http://java.sun.com/docs/books/tutorial/java/landl/subclasses.html
http://java.sun.com/docs/books/tutorial/essential/exceptions/index.html

Classes, Objects, Inheritance

```
class Bicycle {
        public int cadence = 0;
        public int speed = 0;
        public int gear = 1;

        public Bicycle(int startCadence, int startSpeed, int startGear) {
            gear = startGear;
            cadence = startCadence;
            speed = startSpeed;
        }

        public void changeCadence(int newValue) {
            cadence = newValue;
        }

        public void chang
            gear = newVa
        }

        public void speed
            speed = spee
        }

        public void apply
            speed = spee
        }
}
```

```
public class MountainBike extends Bicycle {
    public int seatHeight;

    public MountainBike(int startHeight, int startCadence,
                        int startSpeed, int startGear)
    {
        super(startCadence, startSpeed, startGear);
        seatHeight = startHeight;
    }

    public void setHeight(int newValue) {
        seatHeight = newValue;
    }
}
```

Source: [JTutorial]

Classes, Objects, Inheritance

```
class Bicycle {
```

- Class definition (general form):
  modifier `class MyClass extends MySuperClass implements YourInterface1, ..., YourInterfaceN`
  `{ // fields, constructors, methods }`
- *(Access) modifier*:
  certain combinations of { `public, protected, private, static, final` }

```
public class MountainBike extends Bicycle {
    public int seatHeight;

    public MountainBike(int startHeight, int startCadence,
                        int startSpeed, int startGear)
    {
        super(startCadence, startSpeed, startGear);
        seatHeight = startHeight;
    }

    public void setHeight(int newValue) {
        seatHeight = newValue;
    }
}
```

Source: [JTutorial]

Classes, Objects, Inheritance

```
public int cadence = 0;
public int speed = 0;
public int gear = 1;
```

- Field declaration (general form):
  modifier type name ;
- *(Access) modifier*:
  certain combinations of { `public, protected, private, static, final` }
- *type*: Any primitive or reference type

```
public int seatHeight;
```

Source: [JTutorial]

Classes, Objects, Inheritance

- Method declaration (general form):
  modifier typeOfReturnValue name (parameter) throwsClause {*
  *statement**
  }
- *(Access) modifier*:
  certain combinations of { `public, protected, private, static, final` }
- *typeOfReturnValue*: Any primitive or reference type
- *parameter**: (later)
- *throwsClause**: (later)
- *statement**: statement(s) to execute

```
public MountainBike(int startHeight, int startCadence,
                    int startSpeed, int startGear)
{
    super(startCadence, startSpeed, startGear);
    seatHeight = startHeight;
}

public void setHeight(int newValue) {
    seatHeight = newValue;
}
```

Source: [JTutorial]

Classes, Objects, Inheritance

- Constructor declaration (general form):
  modifier `MyClass` *(parameter*) throwsClause*
  {
  *statement**
  }
- *(Access) modifier*:
  certain combinations of { `public, protected, private, static, final` }
- *parameter**: (later)
- *throwsClause**: (later)
- *statement**: statement(s) to execute

```
public Bicycle(int startCaden
    gear = startGear;
    cadence = startCadence;
    speed = startSpeed;
}

public MountainBike(int startHeight, int startCadence,
                    int startSpeed, int startGear)
{
    super(startCadence, startSpeed, startGear);
    seatHeight = startHeight;
}
```

Source: [JTutorial]

Classes, Objects, Inheritance

Why do we need constructors?

- Ensure **complete** and **consistent** initialization after object creation
- Access superclass constructors:
  Construct object according to definition of superclass, then add specifics
- Provide several constructors for respective use-cases

```
class Bicycle {
  public int cadence;
  public int speed;
  public int gear;

  public Bicycle(int c, int s, int g) {
    cadence = c;
    speed = s;
    gear = g;
  }
}
```

```
class Tandem extends Bicycle {
  public int numberOfDrivers;

  public Tandem(int c, int s, int g, int n) {
    super(c, s, g);
    numberOfDrivers = n;
  }
}
```

Classes, Objects, Inheritance

Why do we need constructors?

- Ensure **complete** and **consistent** initialization after object creation
- Access superclass constructors:
  Construct object according to definition of superclass, then add specifics
- Provide several constructors for respective use-cases

```
class Bicycle {
  public int cadence;
  public int speed;
  public int gear;

  public Bicycle(int c, int s, int g) {
    cadence = c;
    speed = s;
    gear = g;
  }
}
```

```
class Tandem extends Bicycle {
  public int numberOfDrivers;

  public Tandem(int c, int s, int g, int n) {
    super(c, s, g);
    numberOfDrivers = n;
  }
}
```

Why do we need constructors?

- Ensure **complete** and **consistent** initialization after object creation
- Access superclass constructors:
  Construct object according to definition of superclass, then add specifics
- Provide several constructors for respective use-cases

```
class Bicycle {
  public int cadence;
  public int speed;
  public int gear;

  public Bicycle(int c, int s, int g) {
    cadence = c;
    speed = s;
    gear = g;
  }
}
```

```
class Tandem extends Bicycle {
  public int numberOfDrivers;

  public Tandem(int c, int s, int g, int n) {
    super(c, s, g);
    numberOfDrivers = n;
  }
}
```