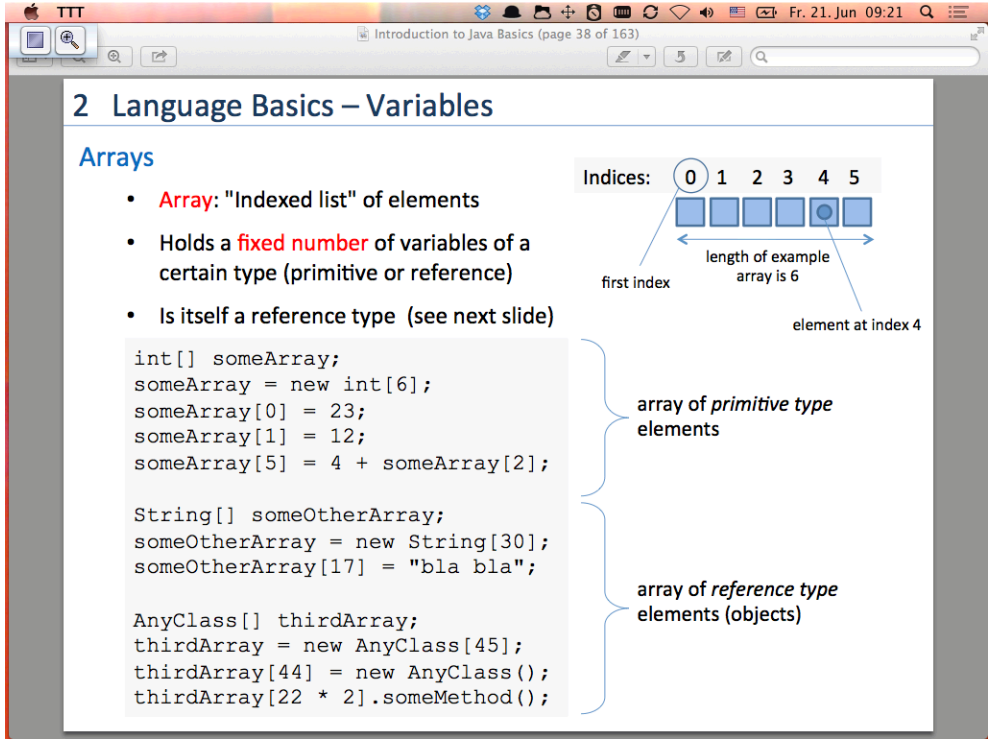
Script generated by TTT

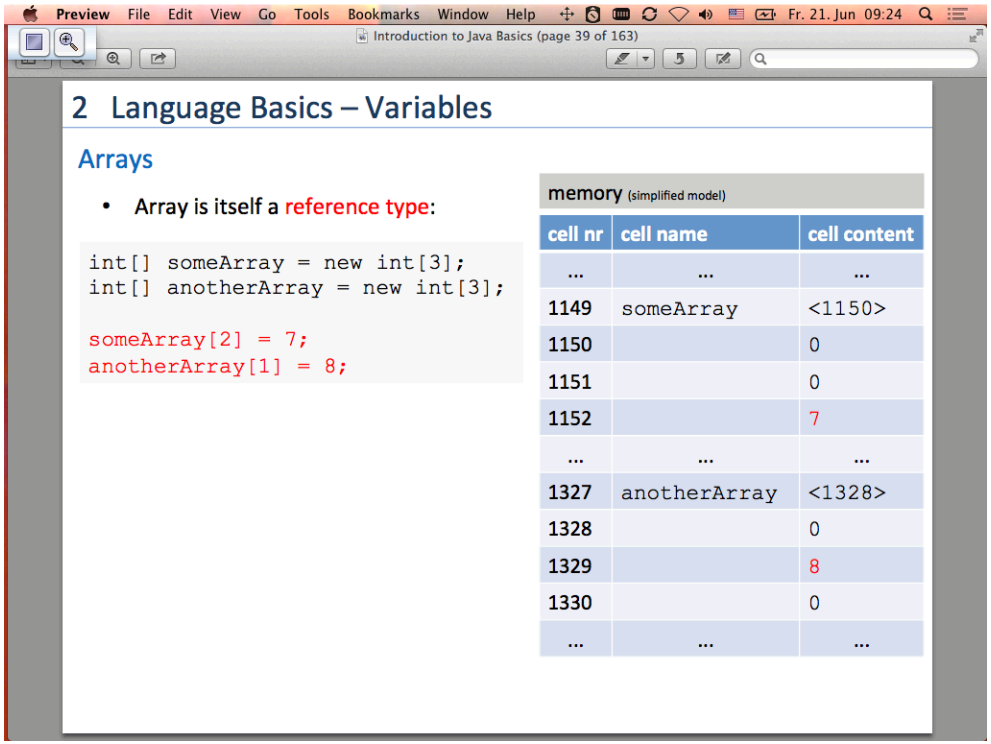
Title: Lehmann: Uebung_Einf_HF (21.06.2013)

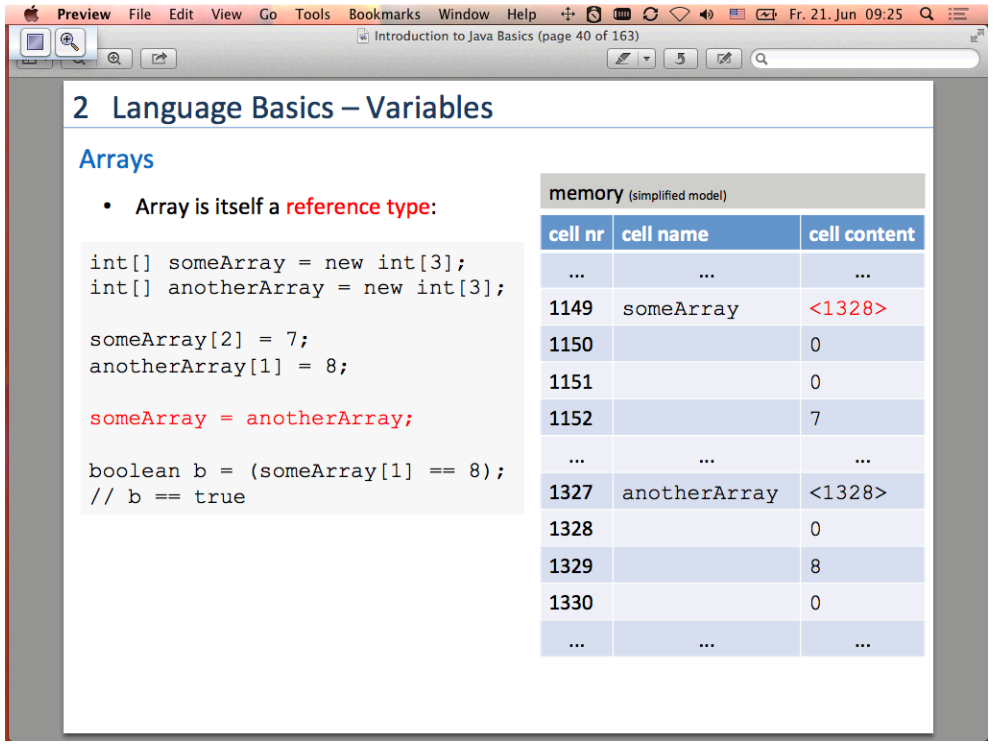
Date: Fri Jun 21 09:21:13 CEST 2013

Duration: 82:25 min

Pages: 63







2 Language Basics – Variables

Arrays

- Array is itself a reference type:

```
int[] someArray = new int[3];
int[] anotherArray = new int[3];

someArray[2] = 7;
anotherArray[1] = 8;

someArray = anotherArray;

boolean b = (someArray[1] == 8);
// b == true
```

- Length property:

```
int l = someArray.length;
// l == 3
```

memory (simplified model)

cell nr	cell name	cell content
...	...	...
1149	someArray	<1328>
1150		0
1151		0
1152		7
...	...	...
1327	anotherArray	<1328>
1328		0
1329		8
1330		0
...	...	...

2 Language 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

2 Language 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 the new instance of Bicycle, e.g. <1150>	Create and initialize new instance of class Bicycle in memory
new double[10]	Reference to the new array of double	Create and initialize new array in memory

2 Language Basics – Expressions, Statements, Blocks

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)

2 Language Basics – Control Flow Statements

- if and if else have a straightforward meaning:

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

```
void applyBrakes(){
     if (speed > 10) {
          speed = speed - 2;      // break really hard
     } else if (speed > 0) {
          speed--;                // soft brakes
     } else {
          System.err.println(
             "The bicycle has already stopped!");
     }
}
```

- switch: Equivalent to sequence of chained if else statements

2 Language 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

2 Language 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

2 Language Basics – Control Flow Statements

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

```
for (int i=0; i<7; i++) { // loo
       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

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

2 Language 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*
}
```

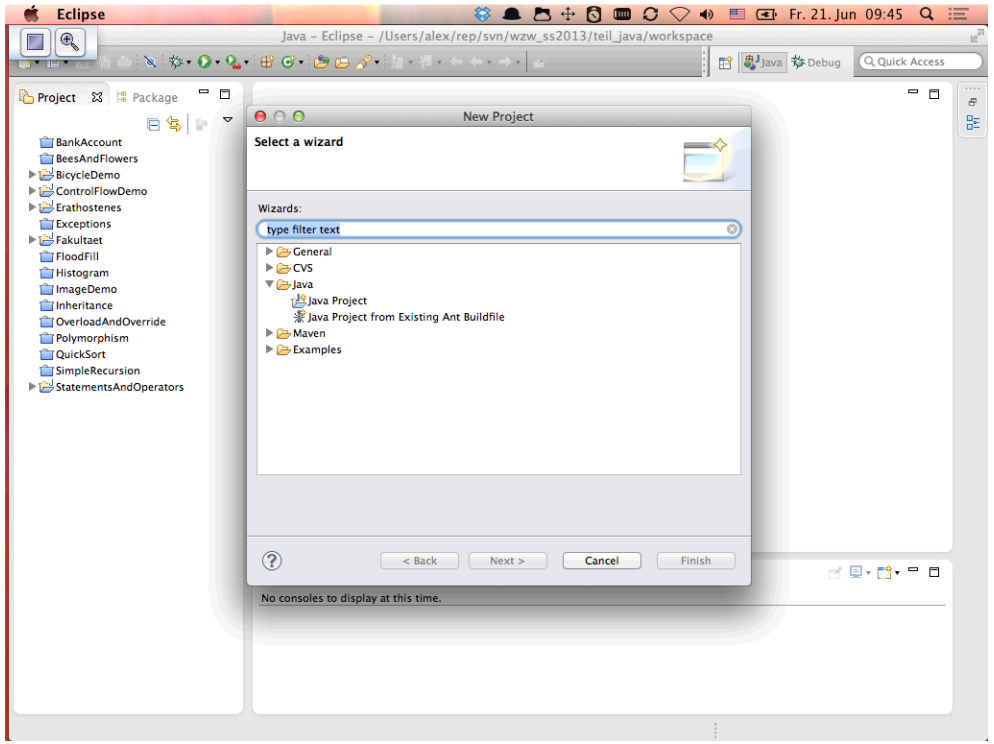
  - initialization expression: Executed once at the beginning of first loop
  - termination expression: If `true` then execute statement(s), else exit loop
  - update expression: Executed after each iteration of the loop

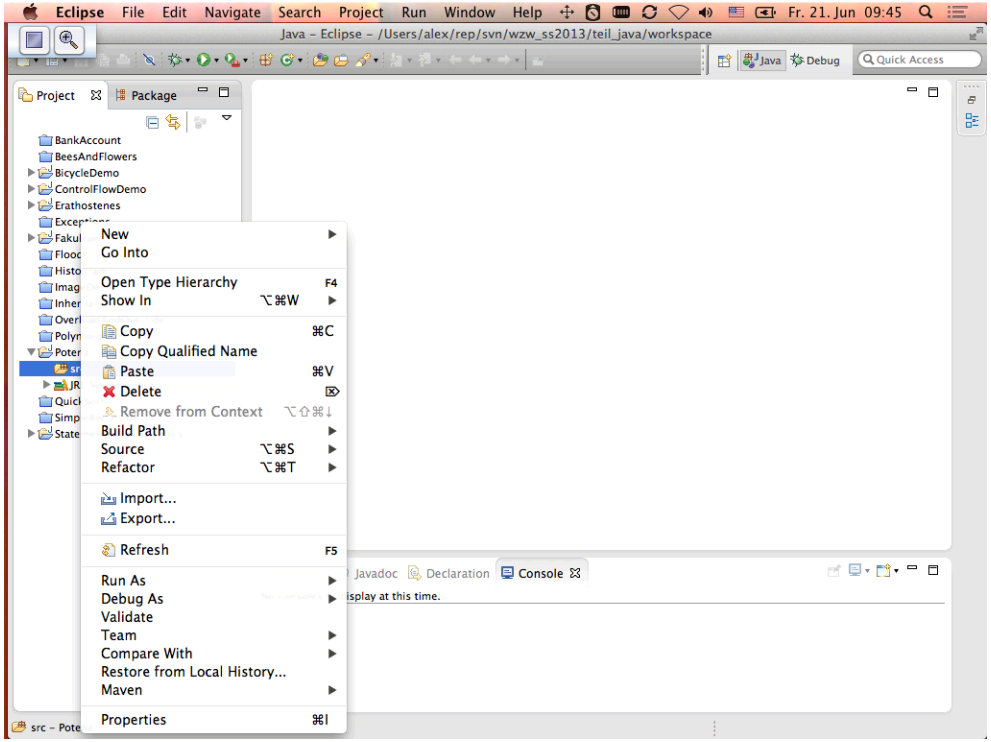
2 Language Basics – Expressions, Statements, Blocks

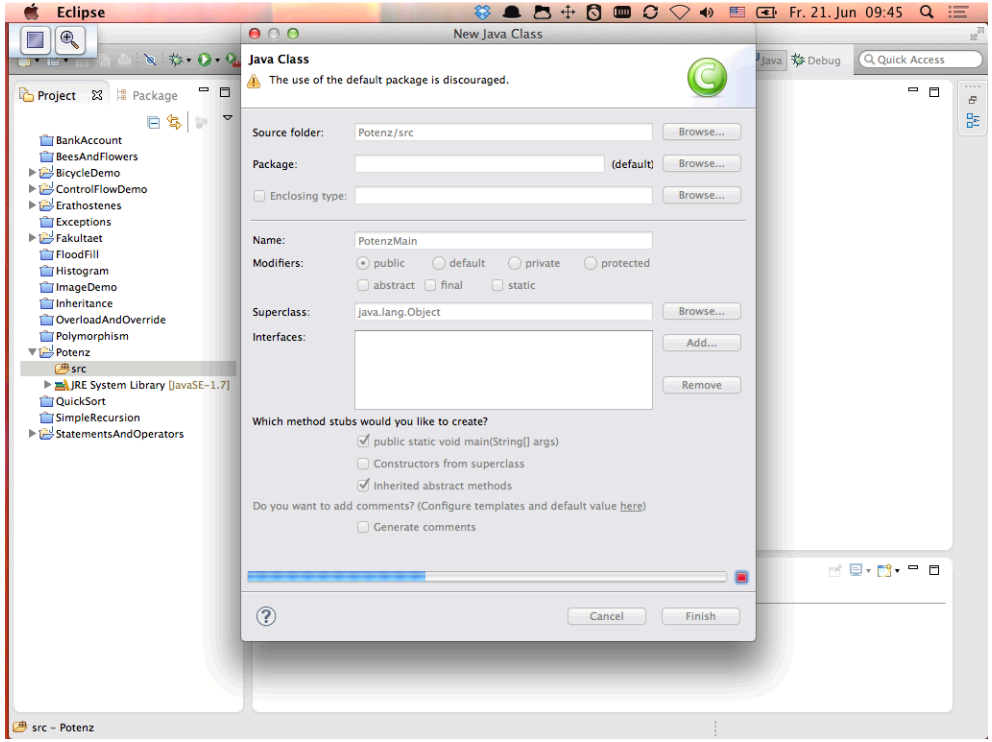
Blocks

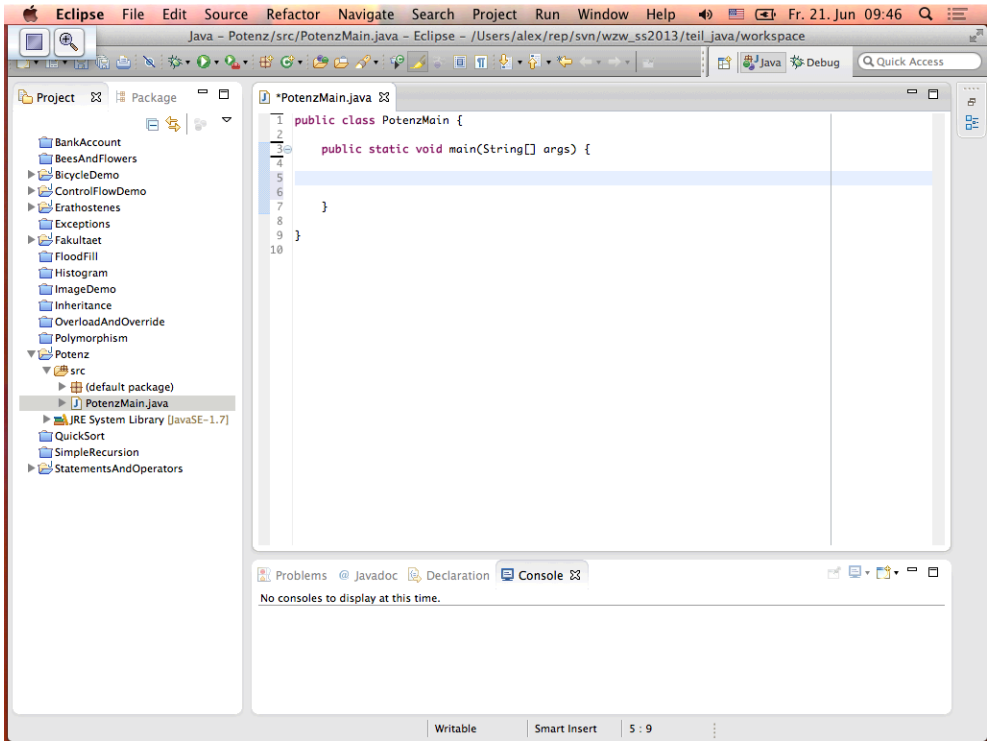
- Variables declared inside a block are only visible from within that block:

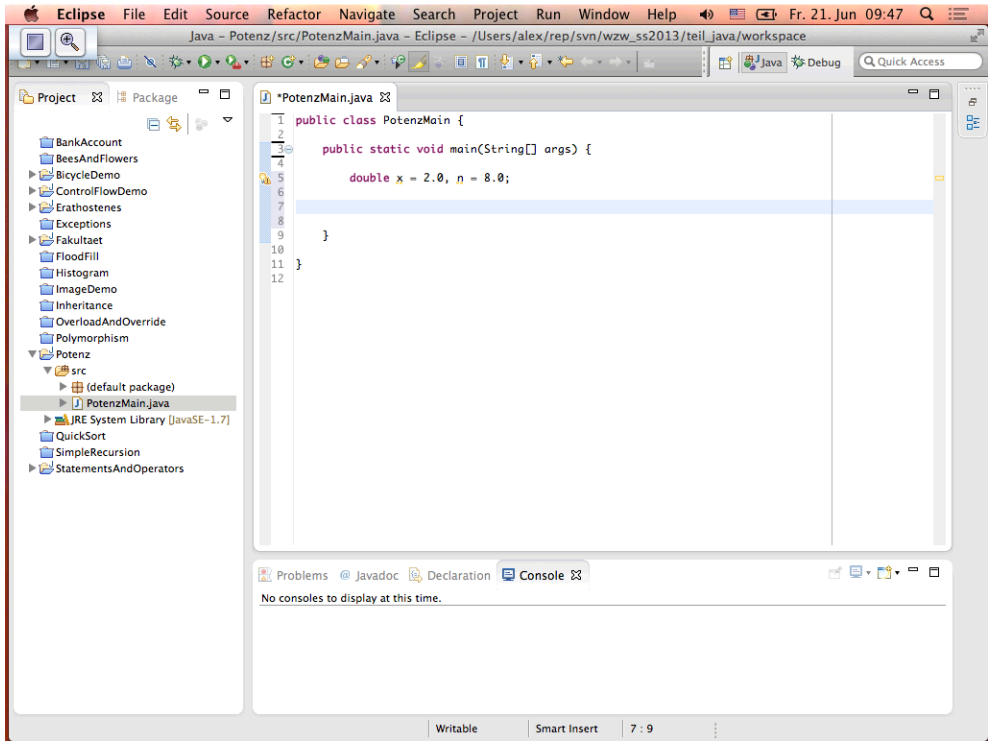
```
int a = 7, b = 6;

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

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

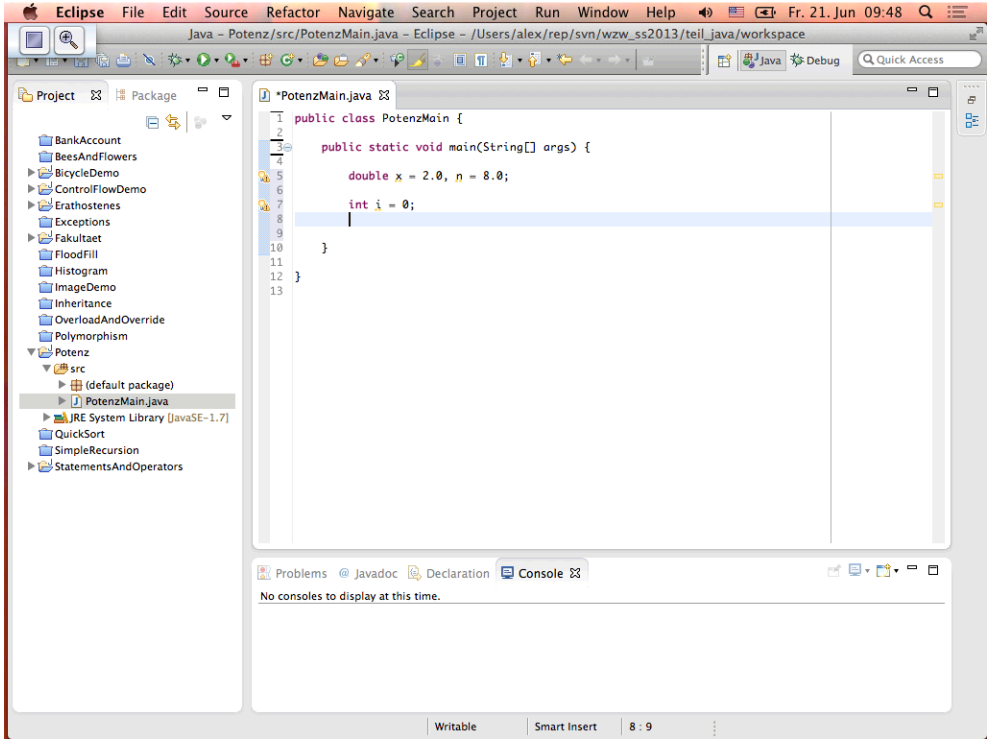


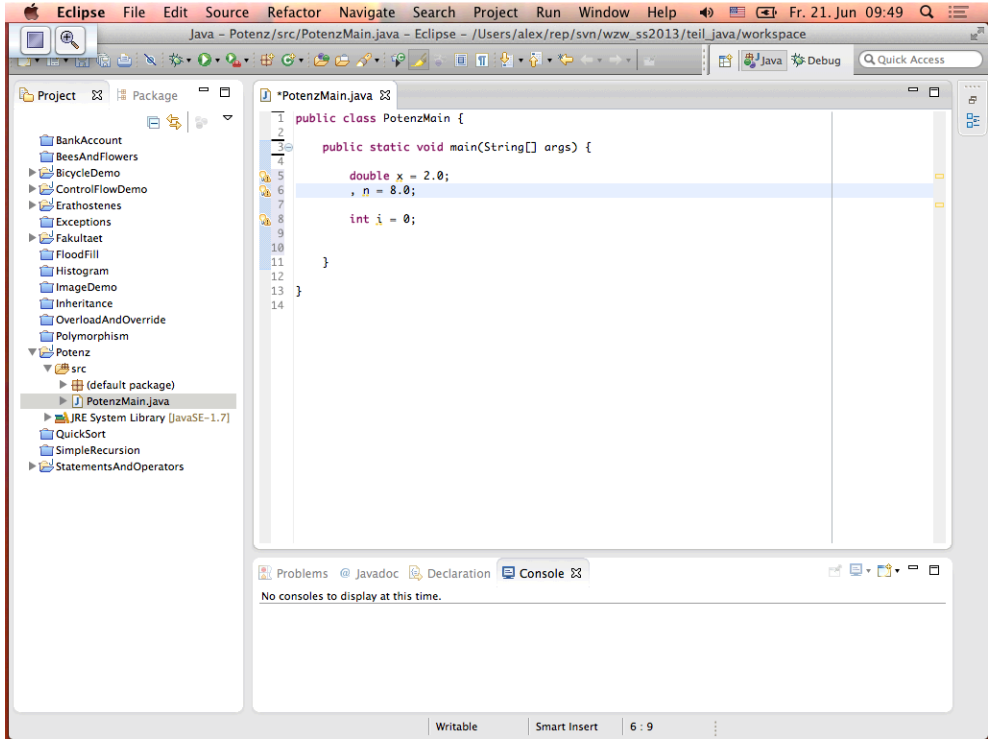

Eclipse File Edit Navigate Search Project Run Window Help Fr. 21. Jun 09:45
Java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
New
Go Into
Open Type Hierarchy F4
Show In
Copy
Copy Qualified Name
Paste
Delete
Remove from Context
Build Path
Source
Refactor
Import...
Export...
Refresh F5
Run As
Debug As
Validate
Team
Compare With
Restore from Local History...
Maven
Properties
Javadoc Declaration Console
Quick Access
Java Debug

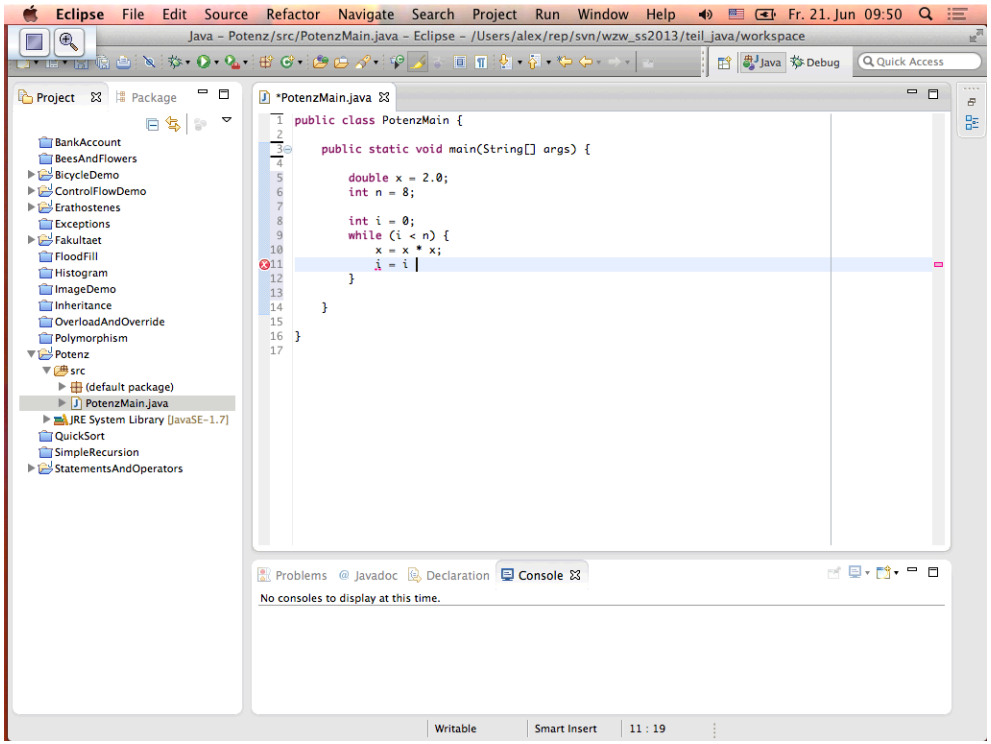

Eclipse Fr. 21. Jun 09:45
New Java Class
Java Class
The use of the default package is discouraged.
Source folder: Potenz/src
Package:
(default)
Enclosing type:
Name: PotenzMain
Modifiers: public default private protected
abstract final static
Superclass: java.lang.Object
Interfaces:
Which method stubs 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
Browse...
Add...
Remove

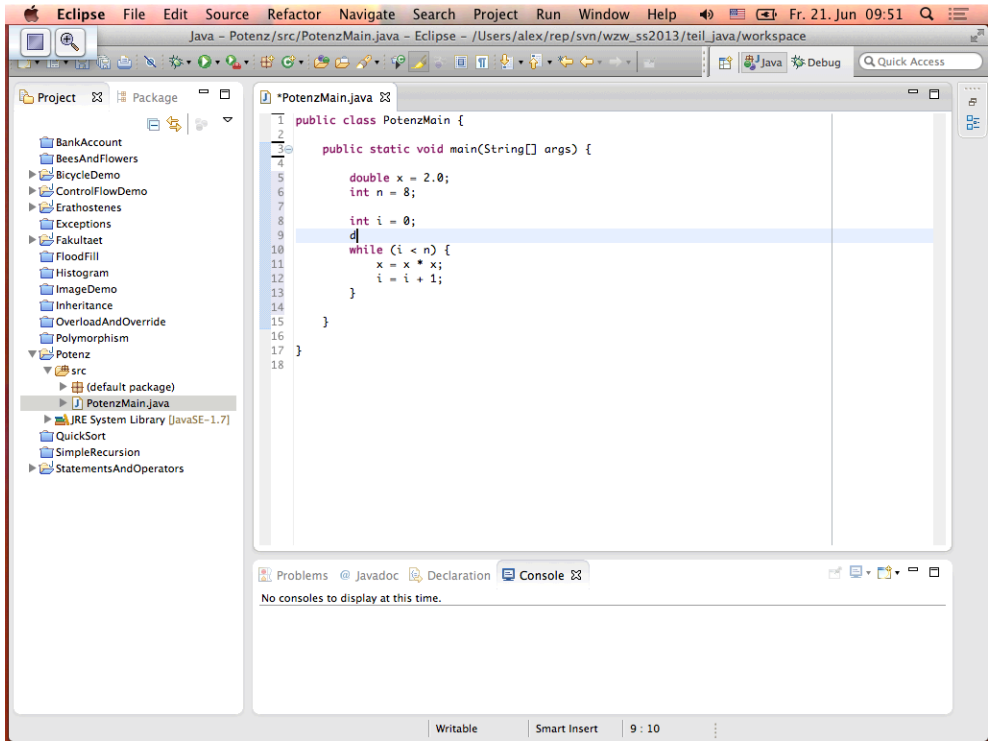

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 09:46
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 5 : 9

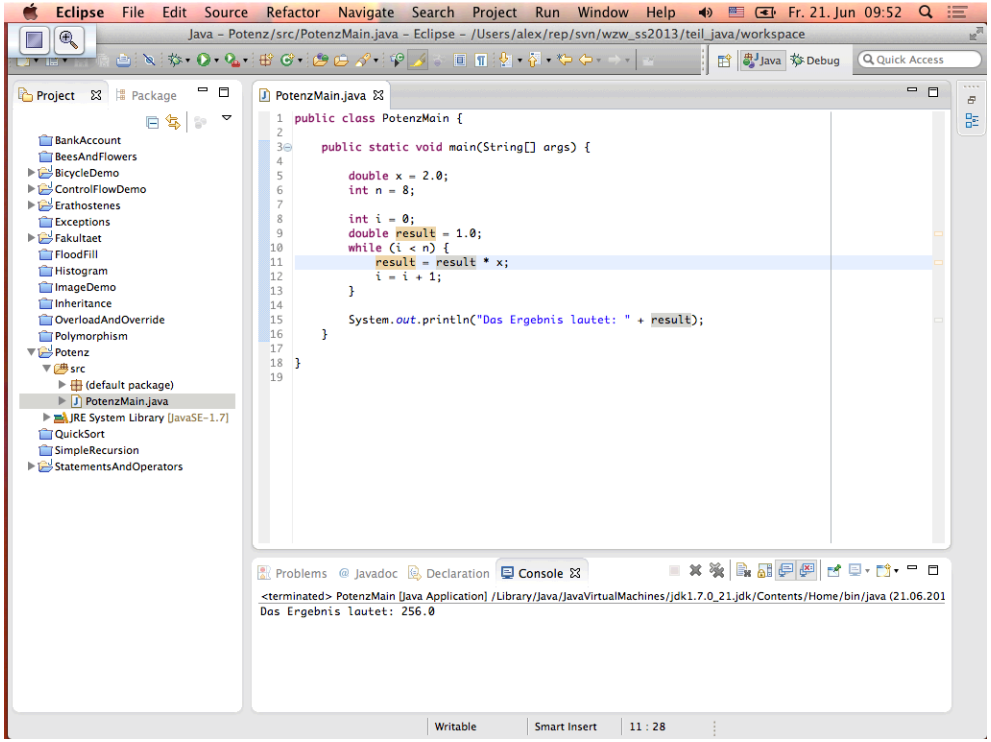

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 09:47
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
double x = 2.0, n = 8.0;
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 7 : 9

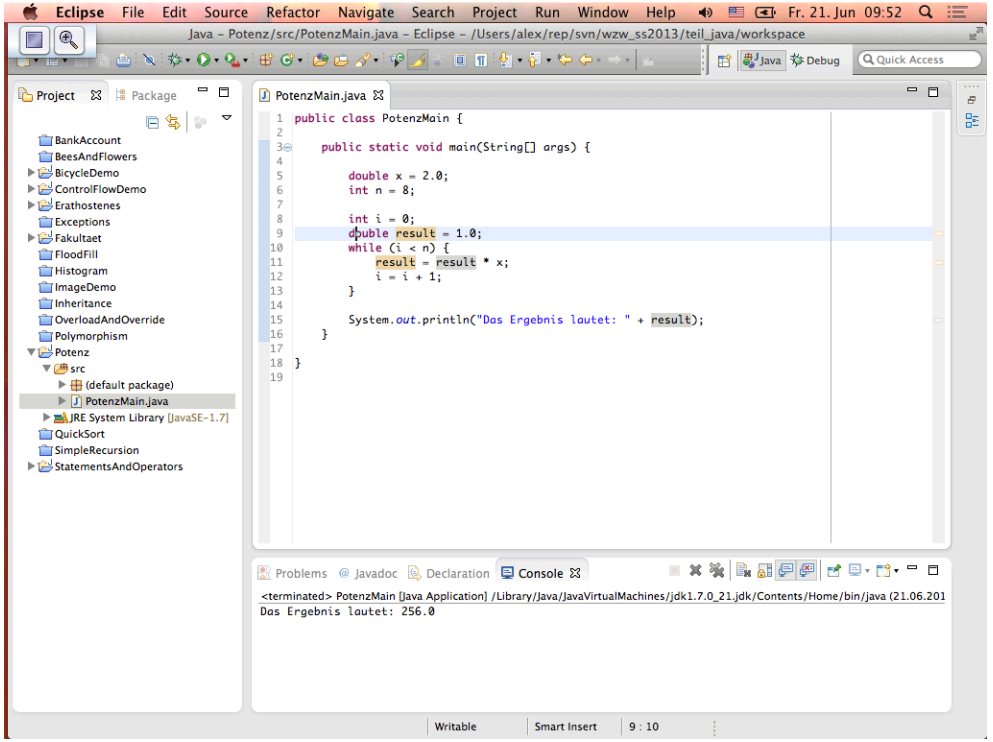

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 09:48
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
*PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
double x = 2.0, n = 8.0;
int i = 0;
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 8 : 9

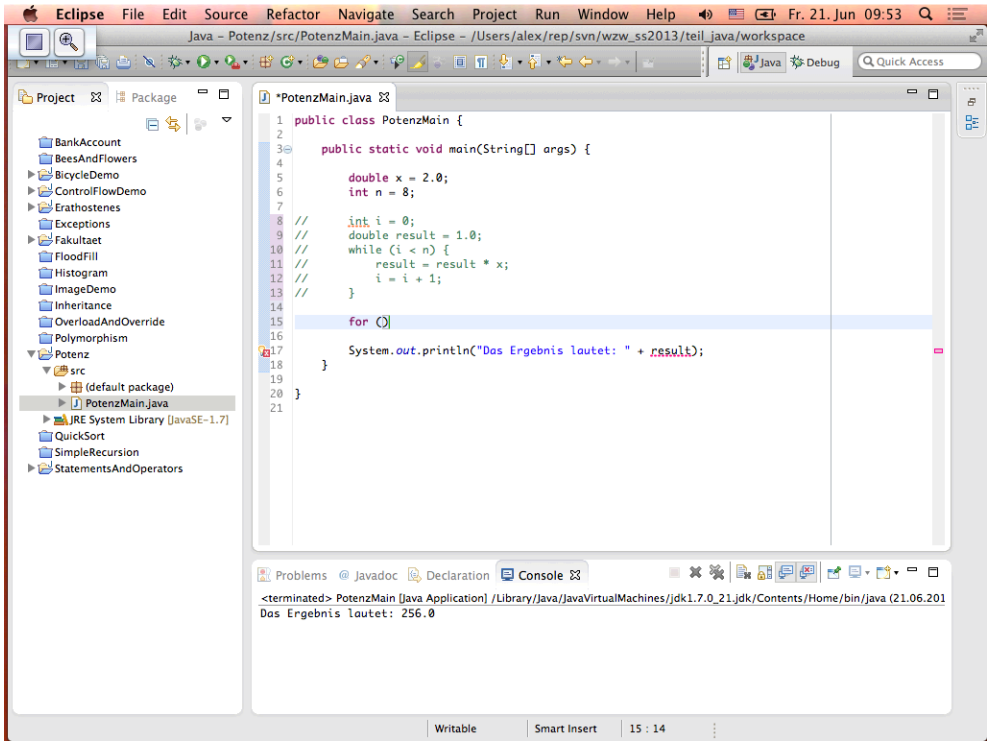

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 09:49
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
double x = 2.0;
, n = 8.0;
int i = 0;
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 6 : 9

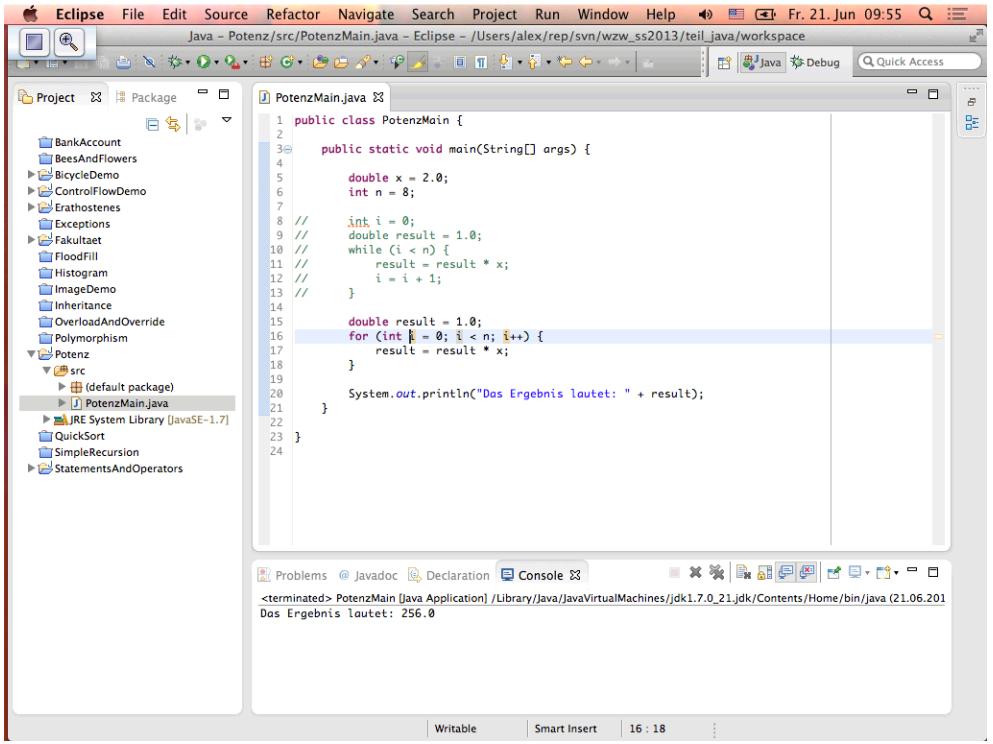

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 09:50
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
double x = 2.0;
int n = 8;
int i = 0;
while (i < n) {
x = x * x;
i = i
}
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 11 : 19

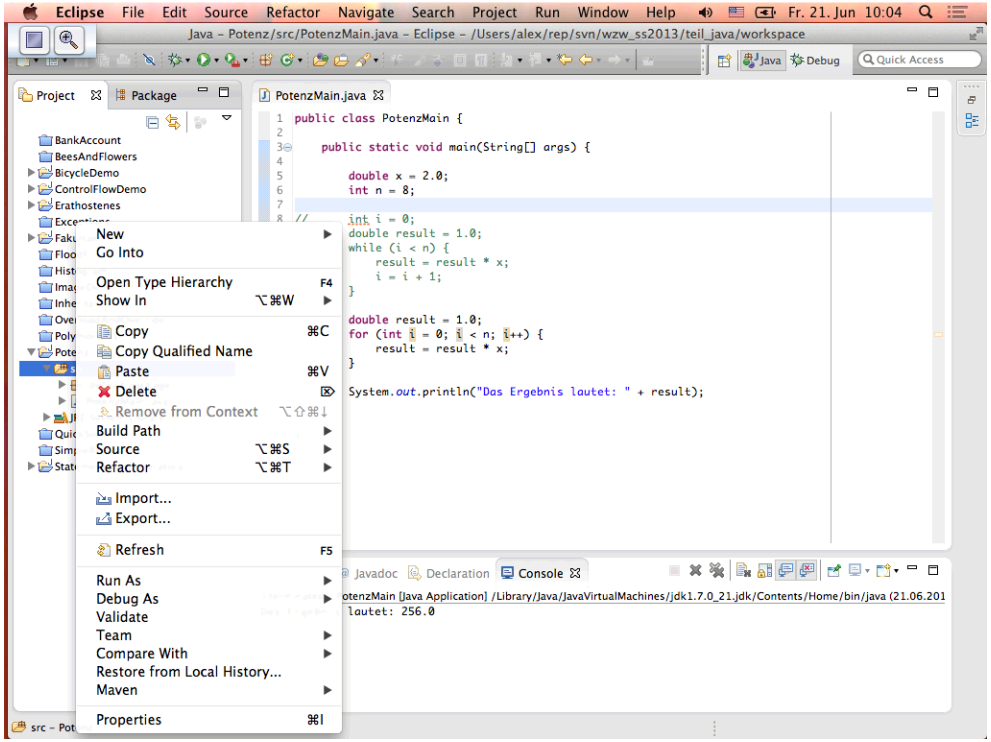

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 09:51
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
*PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
double x = 2.0;
int n = 8;
int i = 0;
d
while (i < n) {
x = x * x;
i = i + 1;
}
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 9 : 10

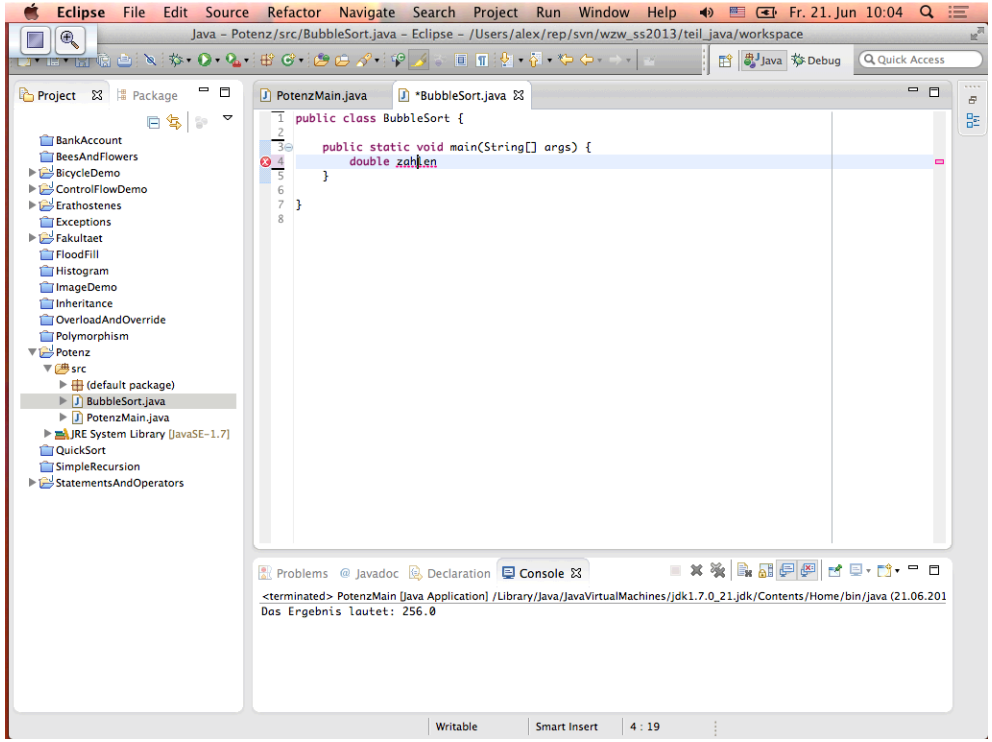

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 09:52
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
double x = 2.0;
int n = 8;
int i = 0;
double result = 1.0;
while (i < n) {
result = result * x;
i = i + 1;
}
System.out.println("Das Ergebnis lautet: " + result);
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 11 : 28

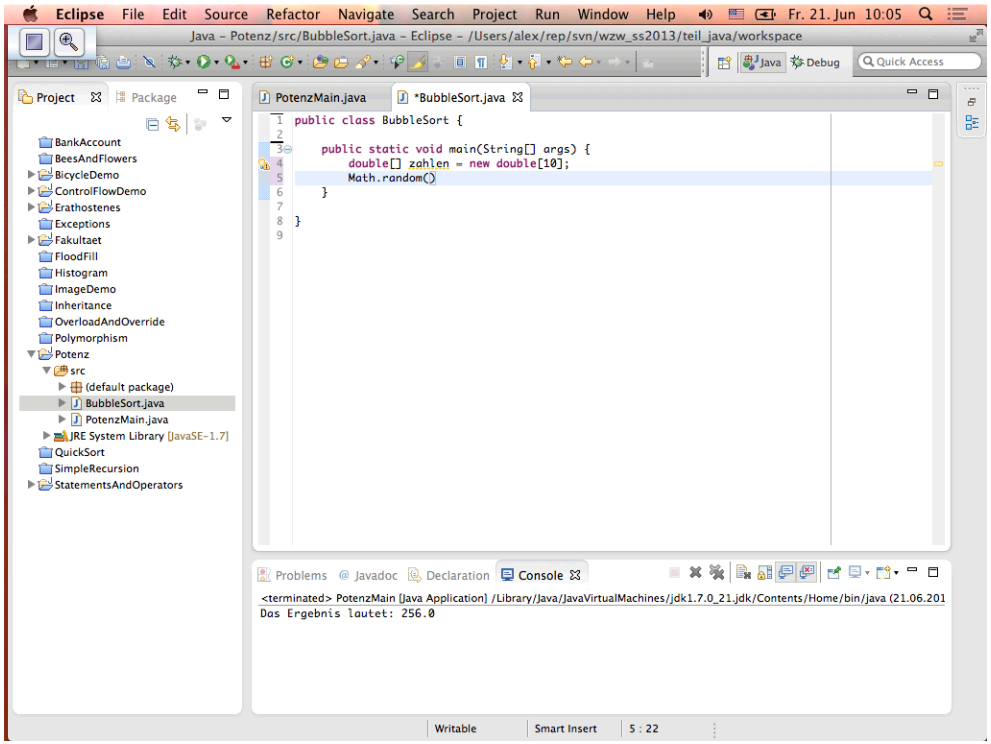

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 09:52
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
double x = 2.0;
int n = 8;
int i = 0;
double result = 1.0;
while (i < n) {
result = result * x;
i = i + 1;
}
System.out.println("Das Ergebnis lautet: " + result);
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 9 : 10

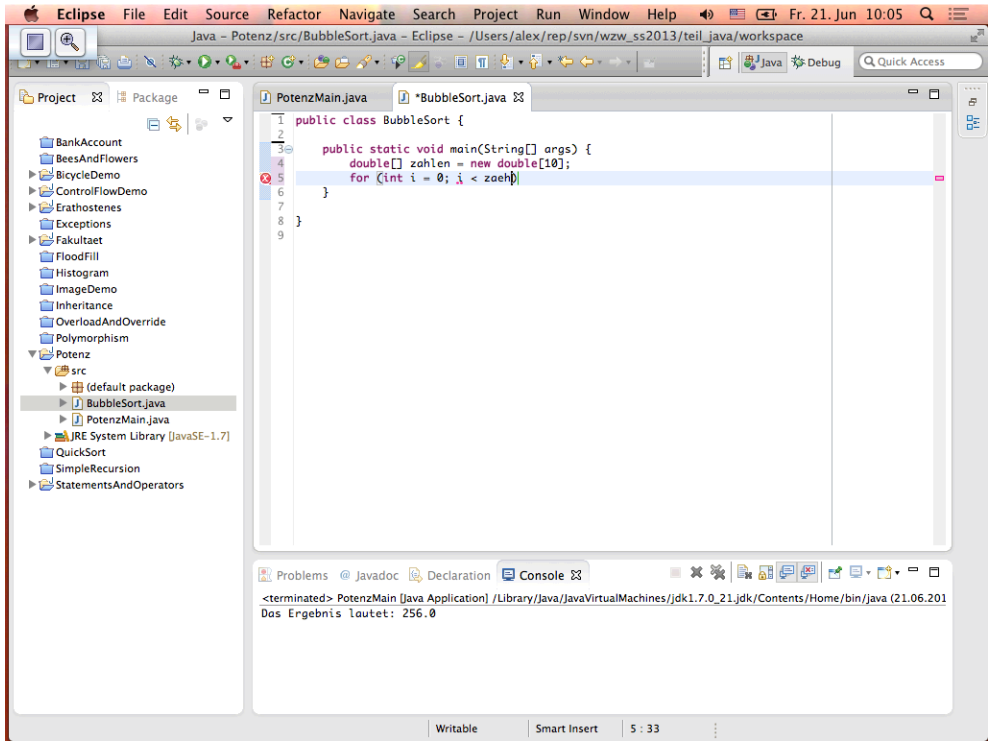

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 09:53
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
*PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
double x = 2.0;
int n = 8;
// int i = 0;
// double result = 1.0;
// while (i < n) {
// result = result * x;
// i = i + 1;
// }
for ()
System.out.println("Das Ergebnis lautet: " + result);
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 15 : 14

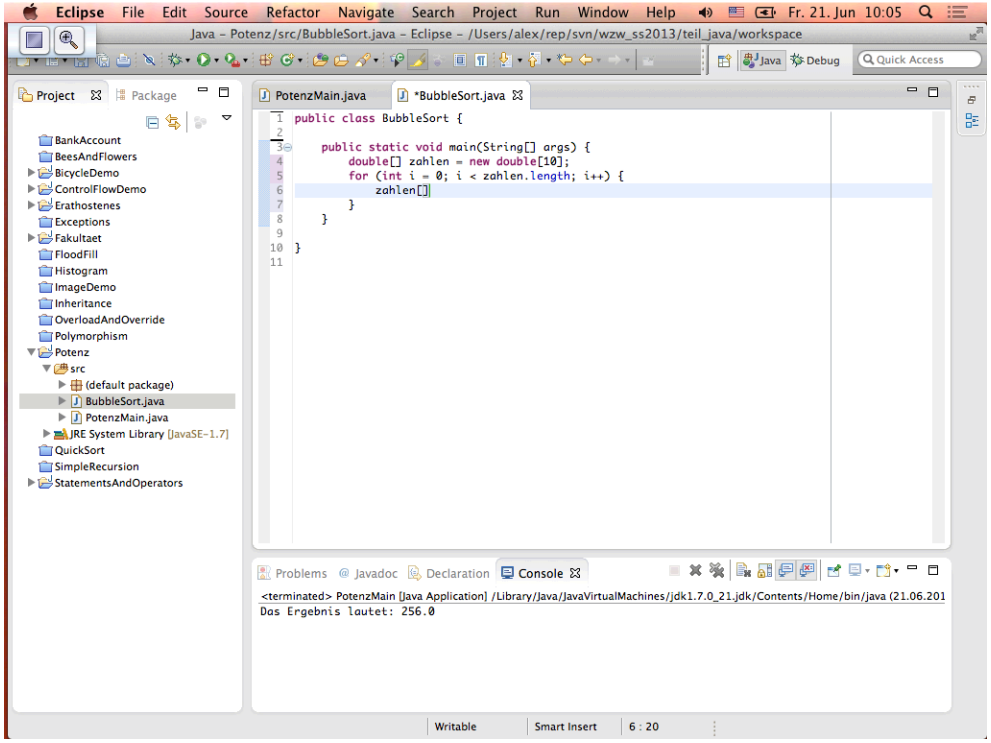

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 09:55
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
double x = 2.0;
int n = 8;
// int i = 0;
// double result = 1.0;
// while (i < n) {
// result = result * x;
// i = i + 1;
// }
double result = 1.0;
for (int i = 0; i < n; i++) {
result = result * x;
}
System.out.println("Das Ergebnis lautet: " + result);
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 16 : 18

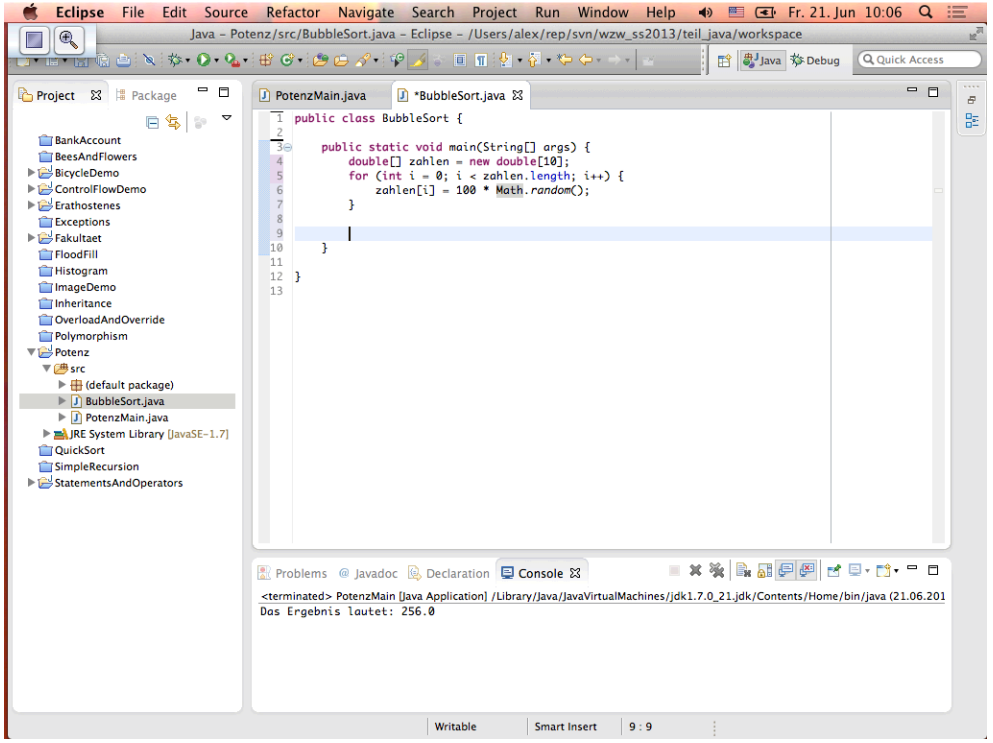

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:04
Java – Potenz/src/PotenzMain.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
PotenzMain.java
public class PotenzMain {
public static void main(String[] args) {
double x = 2.0;
int n = 8;
int i = 0;
double result = 1.0;
while (i < n) {
result = result * x;
i = i + 1;
}
double result = 1.0;
for (int i = 0; i < n; i++) {
result = result * x;
}
System.out.println("Das Ergebnis lautet: " + result);
New
Go Into
Open Type Hierarchy F4
Show In
Copy
Copy Qualified Name
Paste
Delete
Remove from Context
Build Path
Source
Refactor
Import...
Export...
Refresh F5
Run As
Debug As
Validate
Team
Compare With
Restore from Local History...
Maven
Properties
Javadoc Declaration Console
Das Ergebnis lautet: 256.0

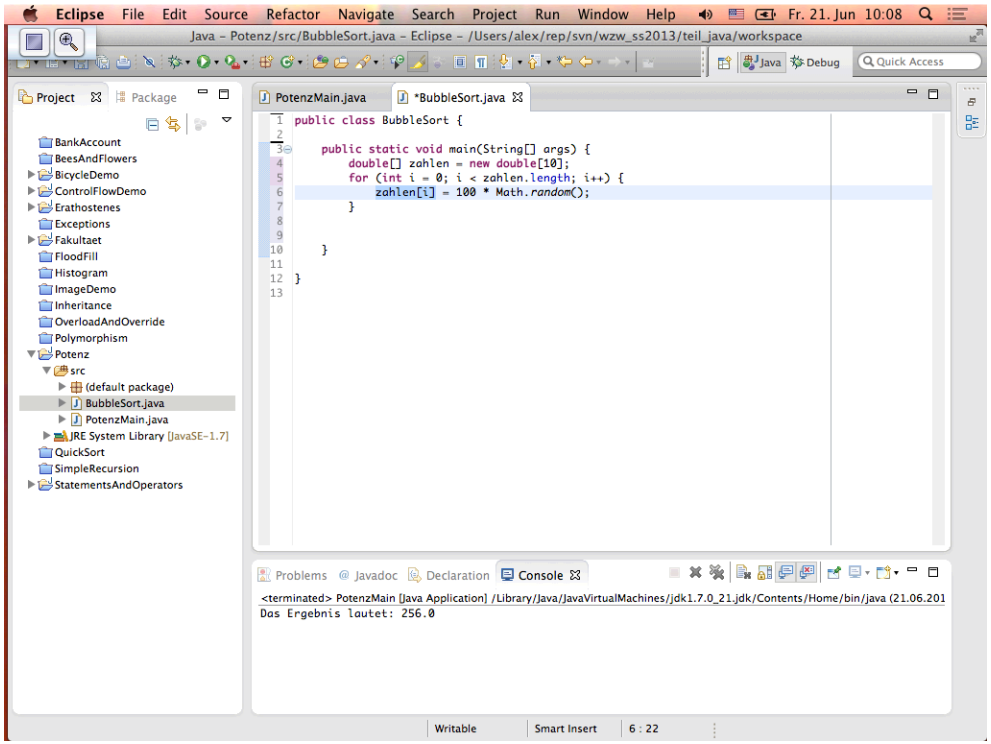

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:04
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double zahlen
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 4 : 19

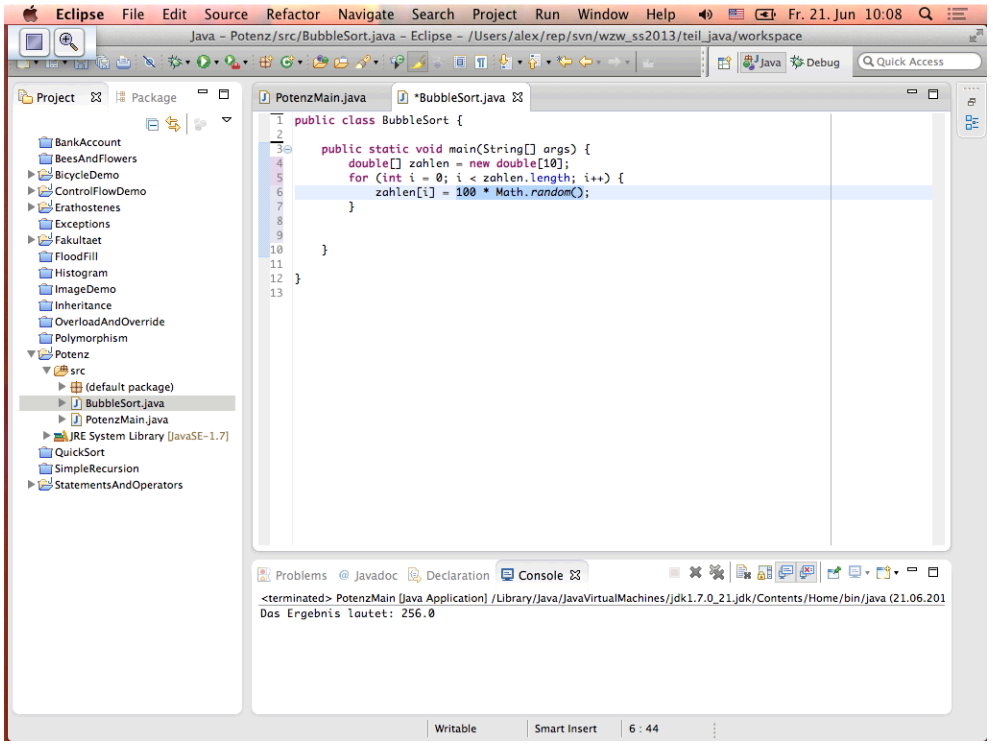

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:05
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
Math.random()
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 5 : 22

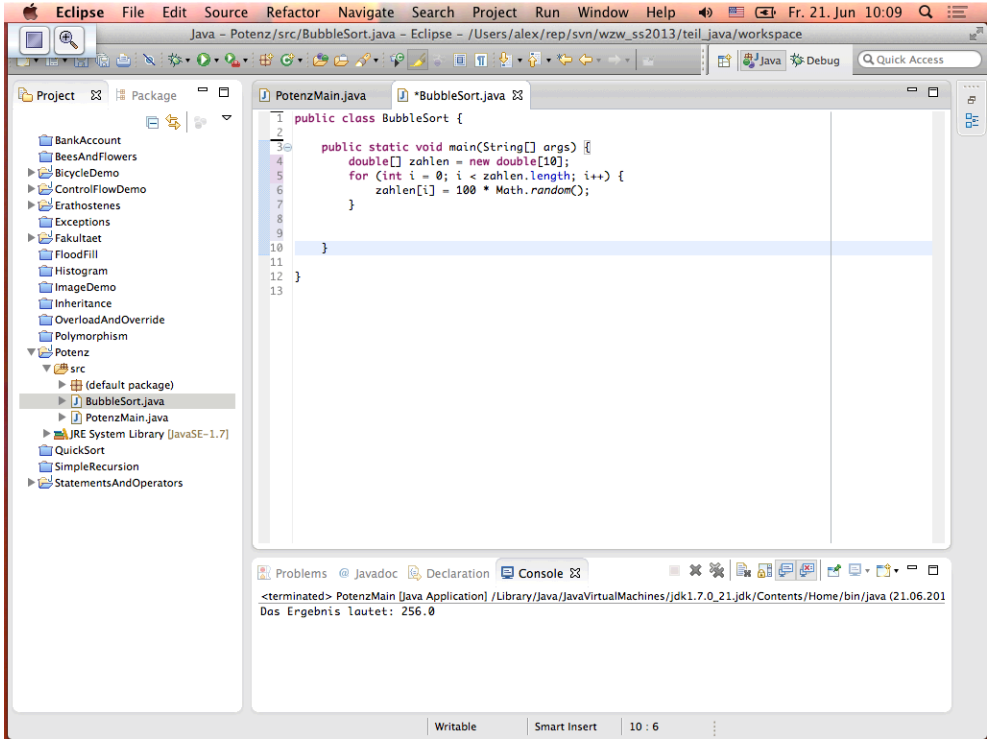

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:05
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zaehl
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 5 : 33

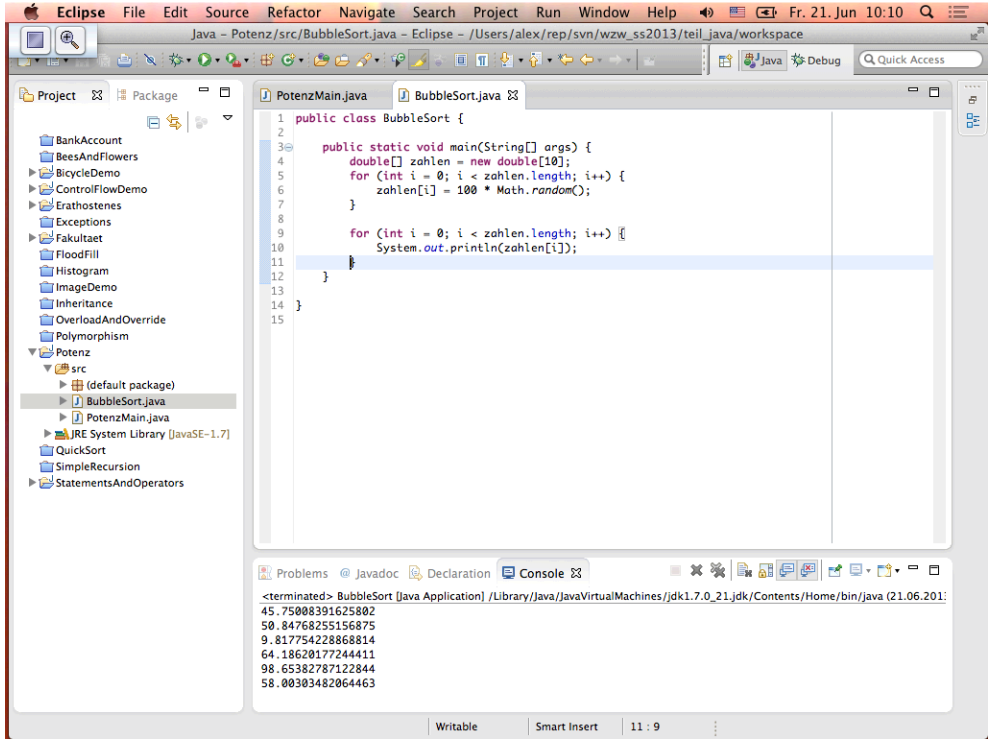

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:05
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[]
}
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 6 : 20

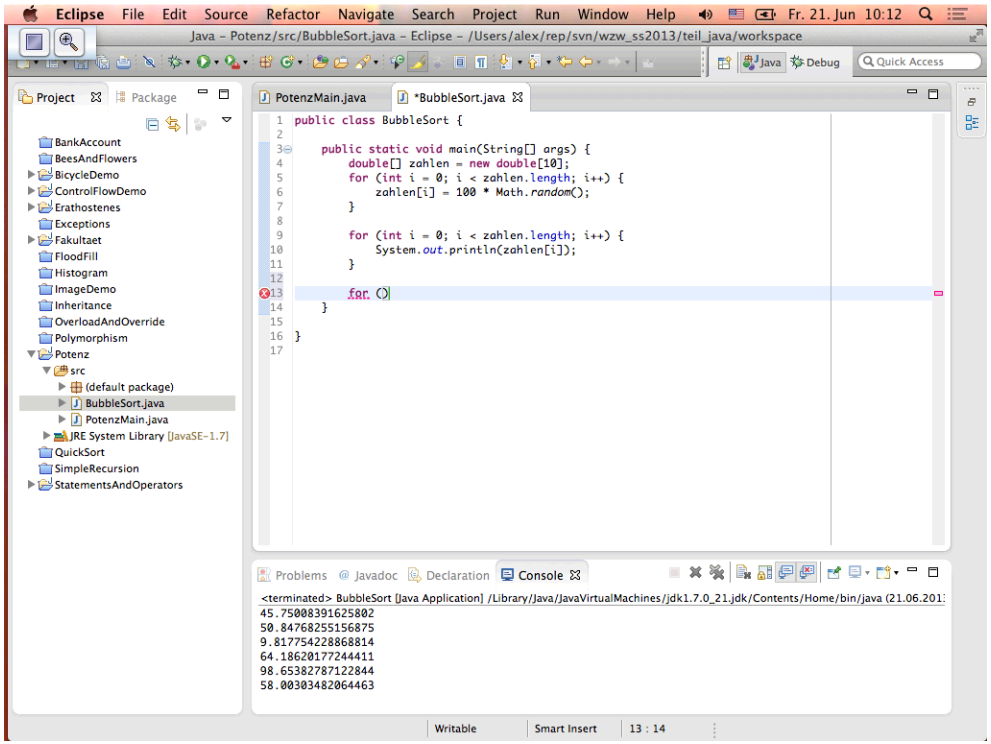

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:06
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 9 : 9

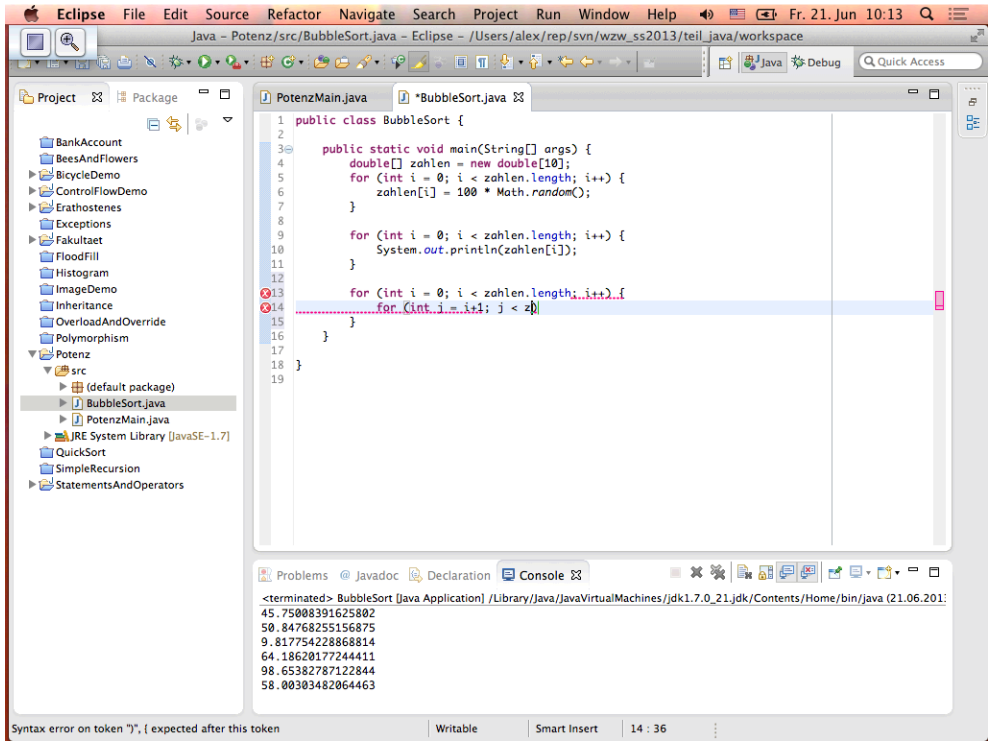

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:08
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 6 : 22

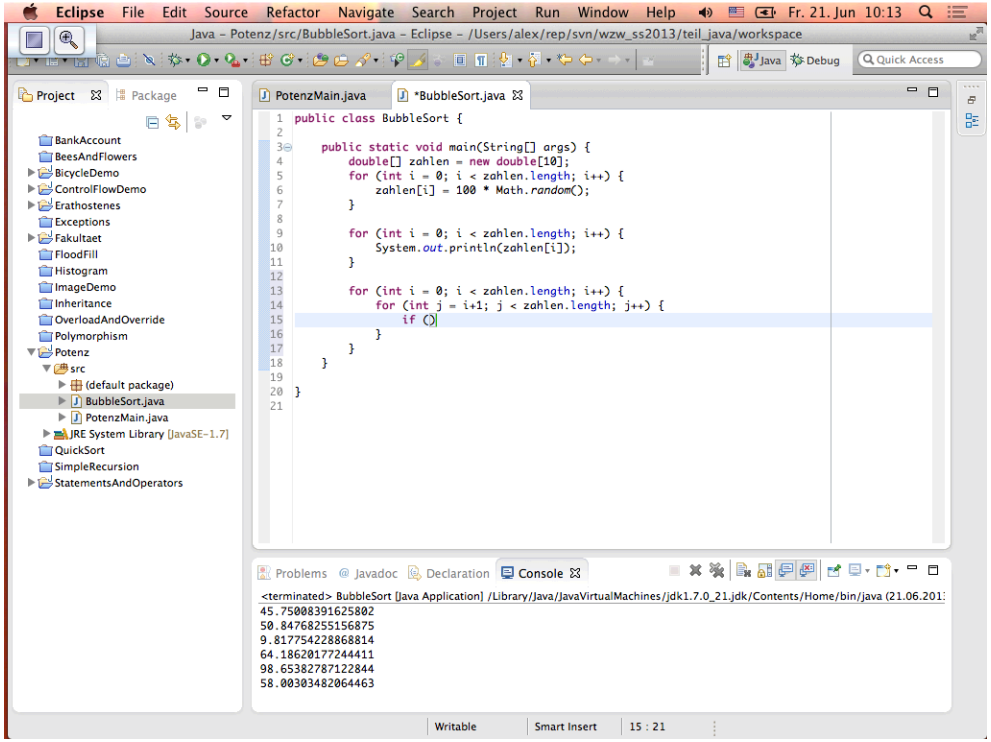

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:08
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 6 : 44

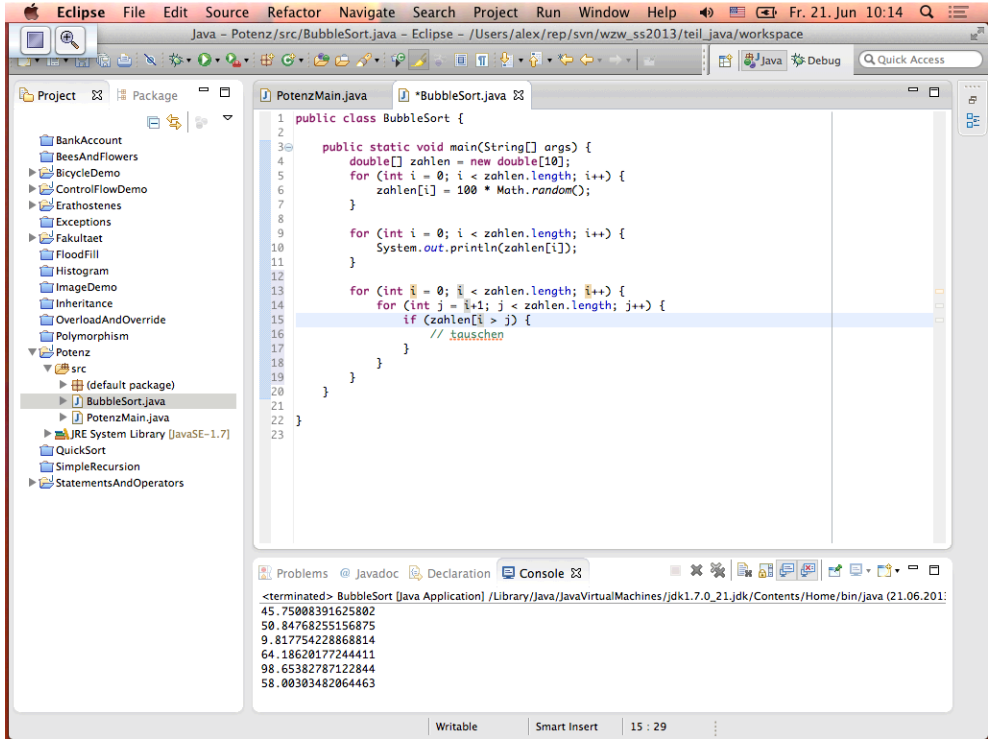

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:09
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
}
}
Problems Javadoc Declaration Console
<terminated> PotenzMain [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
Das Ergebnis lautet: 256.0
Writable Smart Insert 10 : 6

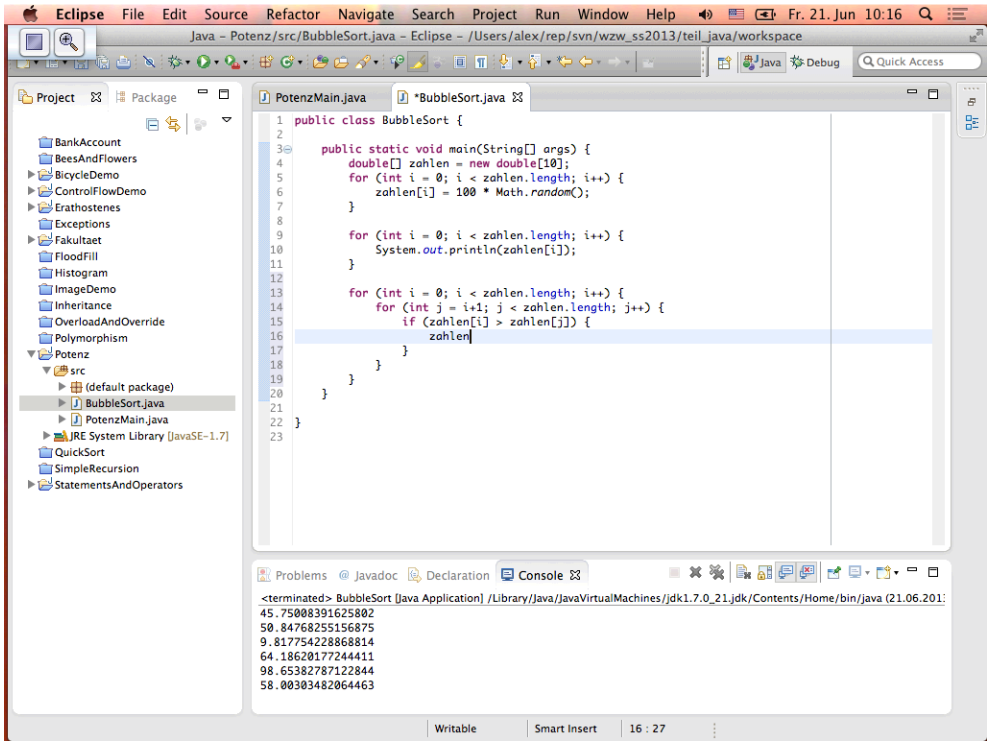

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:10
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
}
}
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
45.75008391625802
50.84768255156875
9.817754228868814
64.18620177244411
98.65382787122844
58.00303482064463
Writable Smart Insert 11 : 9

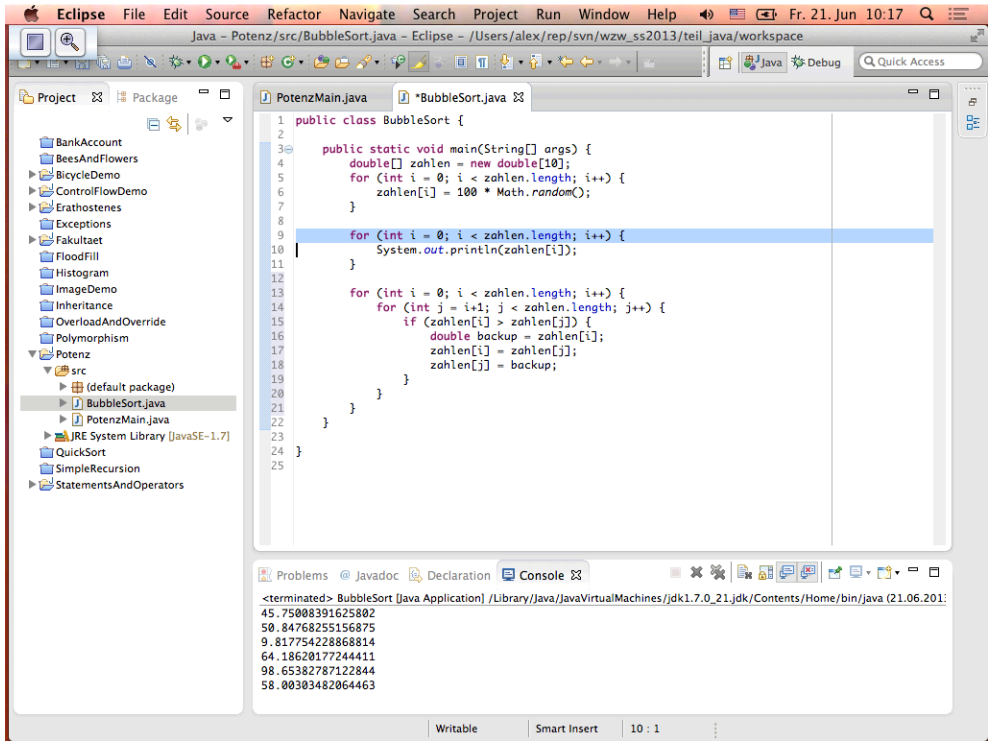

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:12
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for ()
}
}
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
45.75008391625802
50.84768255156875
9.817754228868814
64.18620177244411
98.65382787122844
58.00303482064463
Writable Smart Insert 13 : 14

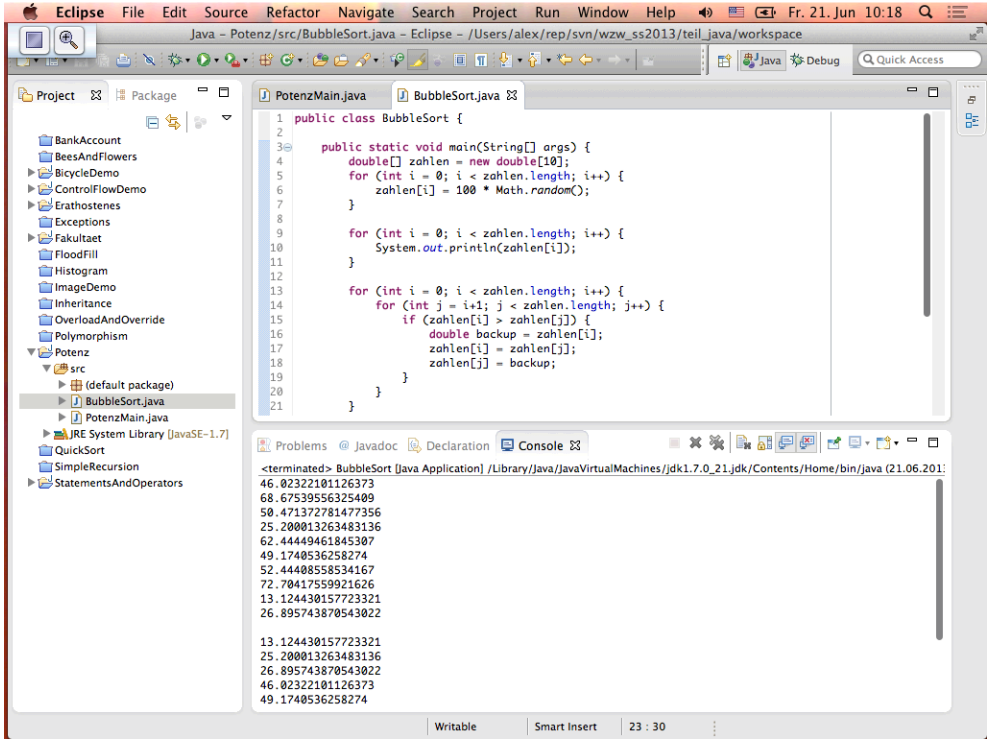

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:13
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < z
}
}
}
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
45.75008391625802
50.84768255156875
9.817754228868814
64.18620177244411
98.65382787122844
58.00303482064463
Syntax error on token ")", { expected after this token
Writable Smart Insert 14 : 36

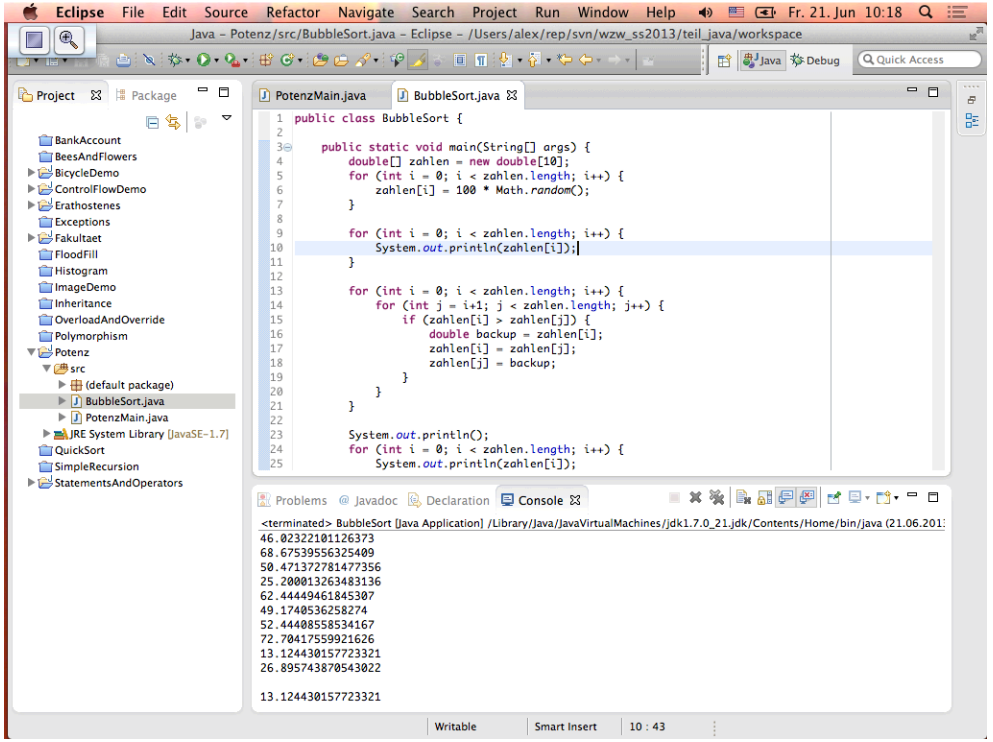

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:13
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if ()
}
}
}
}
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
45.75008391625802
50.84768255156875
9.817754228868814
64.18620177244411
98.65382787122844
58.00303482064463
Writable Smart Insert 15 : 21

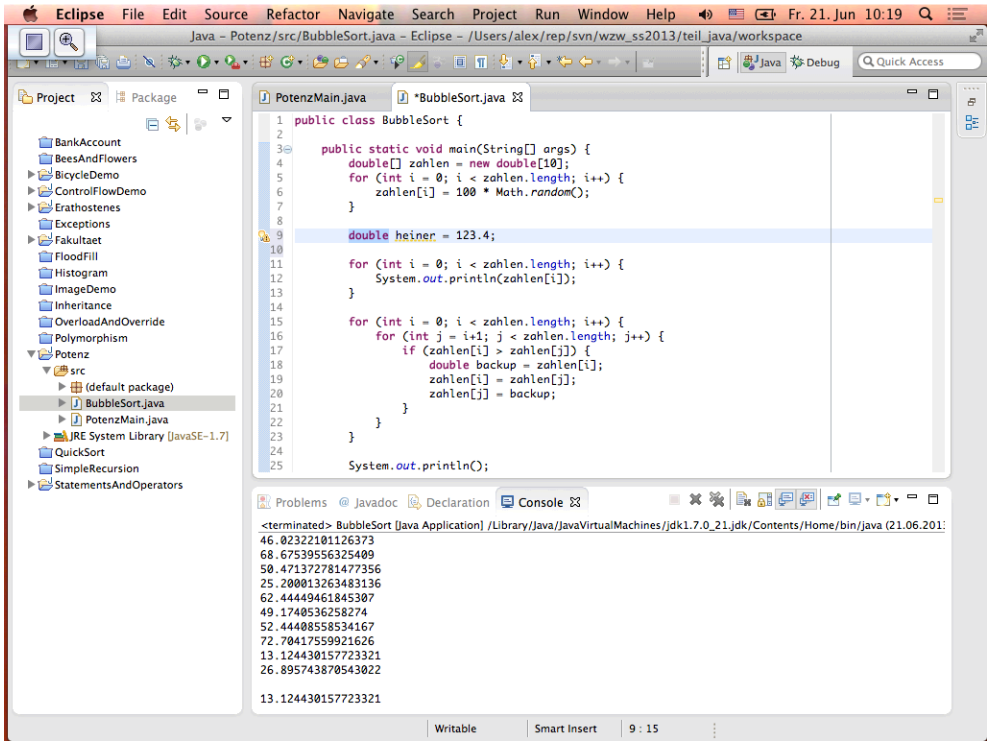

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:14
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > j) {
// tauschen
}
}
}
}
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
45.75008391625802
50.84768255156875
9.817754228868814
64.18620177244411
98.65382787122844
58.00303482064463
Writable Smart Insert 15 : 29

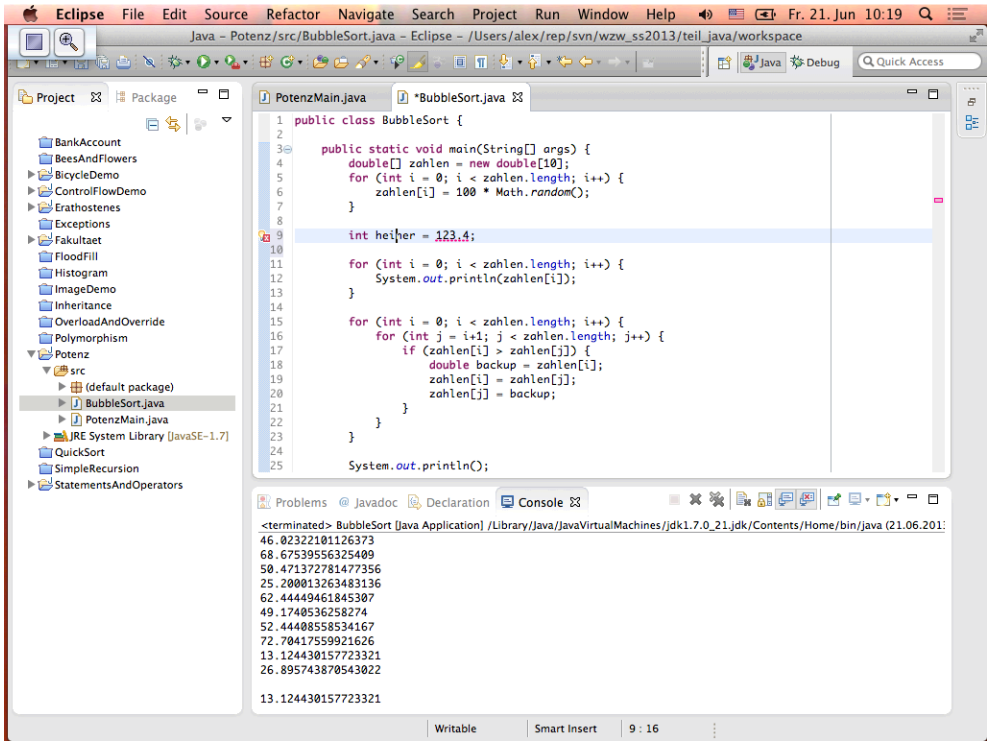

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:16
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
zahlen
}
}
}
}
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
45.75008391625802
50.84768255156875
9.817754228868814
64.18620177244411
98.65382787122844
58.00303482064463
Writable Smart Insert 16 : 27

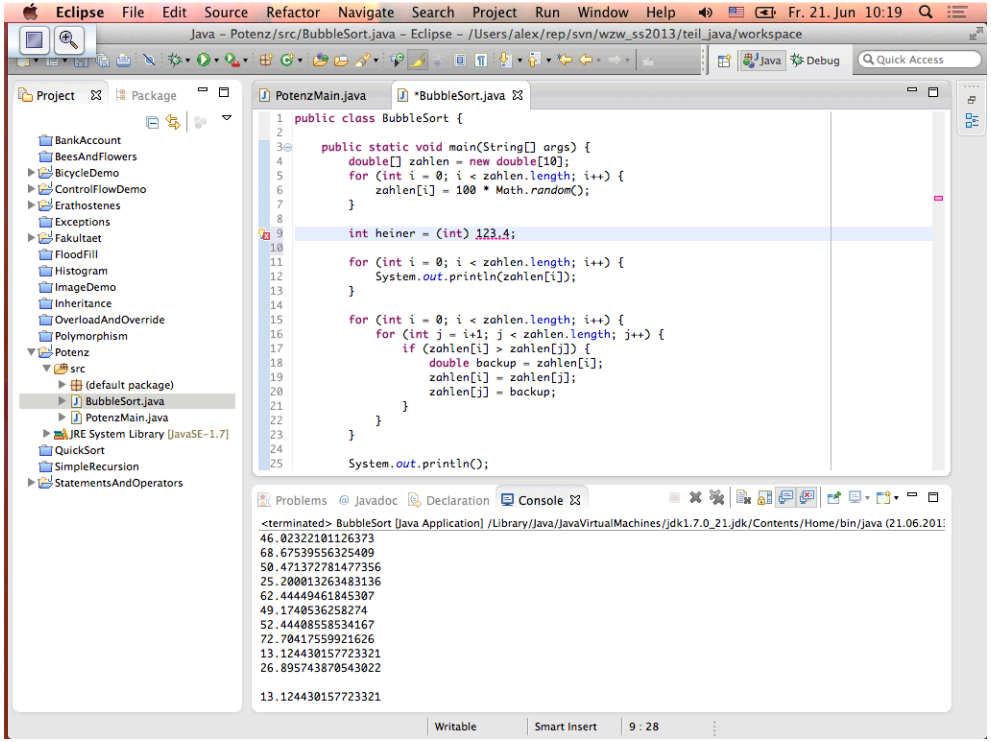

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:17
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
double backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
}
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
45.75008391625802
50.84768255156875
9.817754228868814
64.18620177244411
98.65382787122844
58.00303482064463
Writable Smart Insert 10 : 1

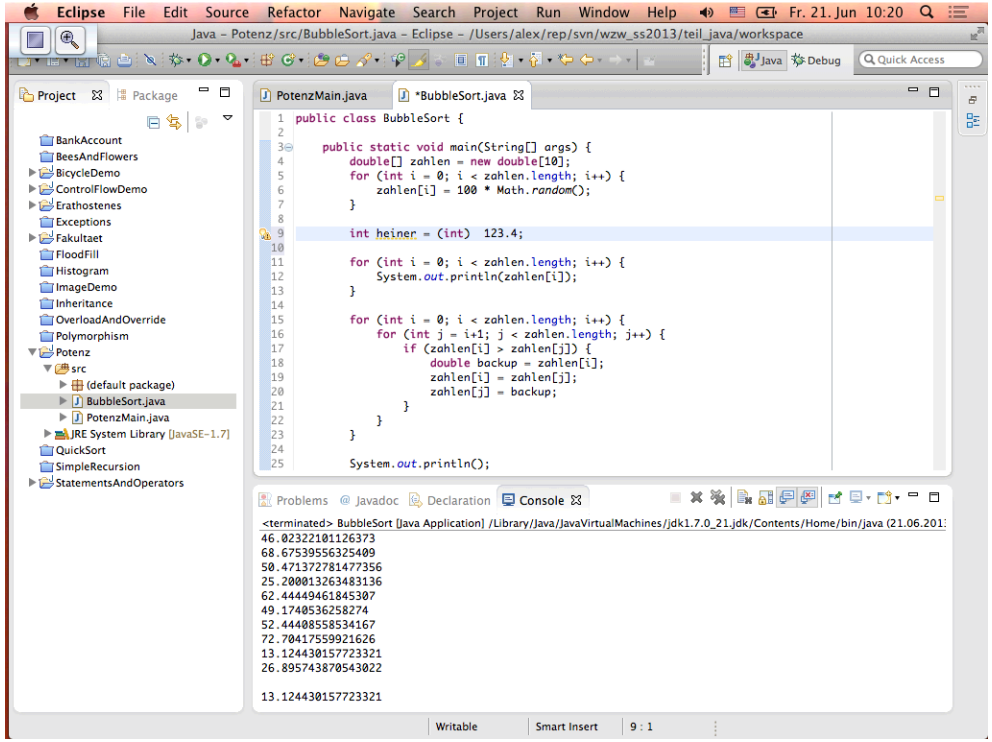

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:18
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
BubbleSort.java
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
PotenzMain.java
BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
double backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
46.02322101126373
68.67539556325409
50.471372781477356
25.200013263483136
62.44449461845307
49.1740536258274
52.44408558534167
72.70417559921626
13.124430157723321
26.895743870543022
13.124430157723321
25.200013263483136
26.895743870543022
46.02322101126373
49.1740536258274
Writable Smart Insert 23 : 30

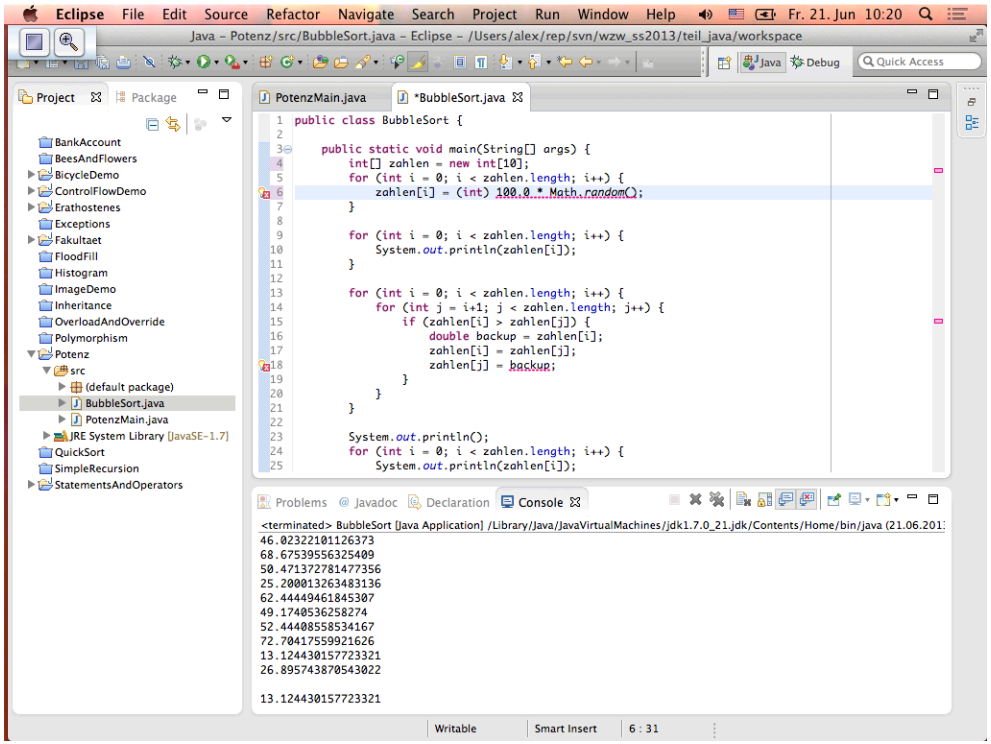

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:18
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
BubbleSort.java
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
PotenzMain.java
BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
double backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
System.out.println();
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
46.02322101126373
68.67539556325409
50.471372781477356
25.200013263483136
62.44449461845307
49.1740536258274
52.44408558534167
72.70417559921626
13.124430157723321
26.895743870543022
13.124430157723321
Writable Smart Insert 10 : 43

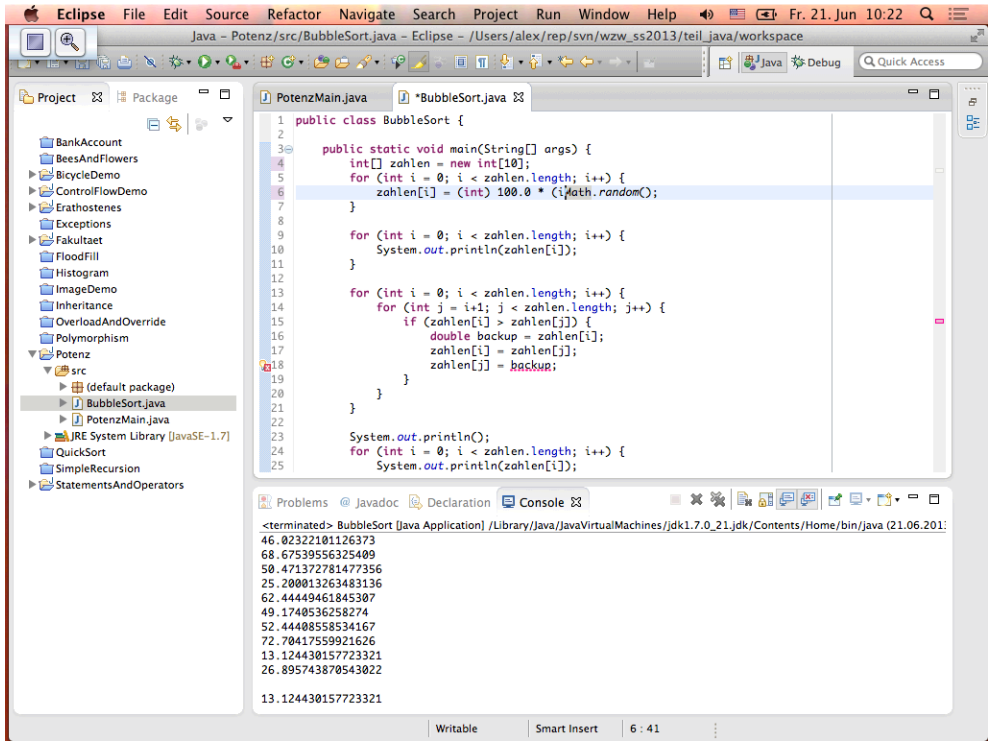

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:19
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
BubbleSort.java
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
double heiner = 123.4;
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
double backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
System.out.println();
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
46.02322101126373
68.67539556325409
50.471372781477356
25.200013263483136
62.44449461845307
49.1740536258274
52.44408558534167
72.70417559921626
13.124430157723321
26.895743870543022
13.124430157723321
Writable Smart Insert 9 : 15

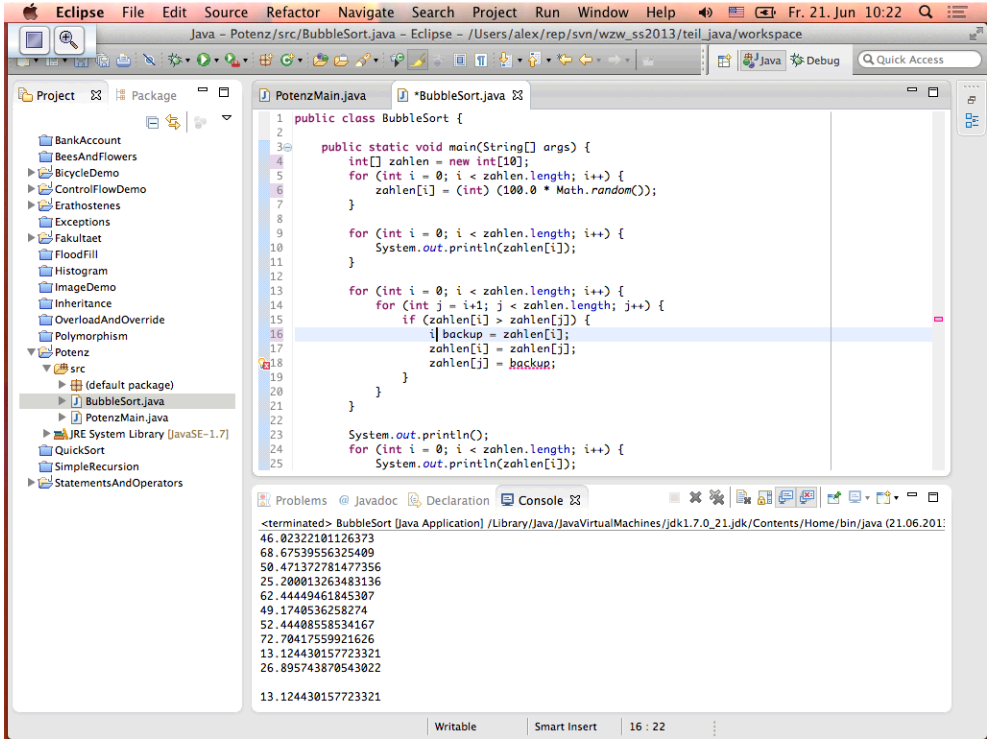

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:19
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
BubbleSort.java
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
int heiner = 123.4;
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
double backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
System.out.println();
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
46.02322101126373
68.67539556325409
50.471372781477356
25.200013263483136
62.44449461845307
49.1740536258274
52.44408558534167
72.70417559921626
13.124430157723321
26.895743870543022
13.124430157723321
Writable Smart Insert 9 : 16

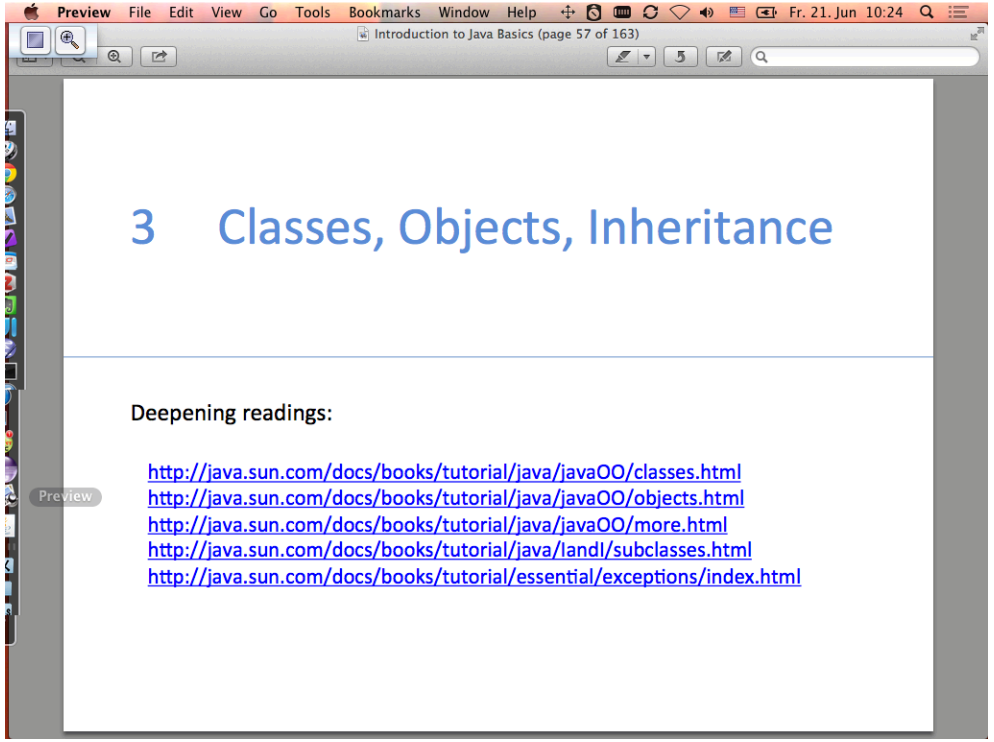

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:19
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
int heiner = (int) 123.4;
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
double backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
System.out.println();
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
BubbleSort.java
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
Problems
Javadoc
Declaration
Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
46.02322101126373
68.67539556325409
50.471372781477356
25.200013263483136
62.44449461845307
49.1740536258274
52.44408558534167
72.70417559921626
13.124430157723321
26.895743870543022
13.124430157723321
Writable
Smart Insert
9 : 28

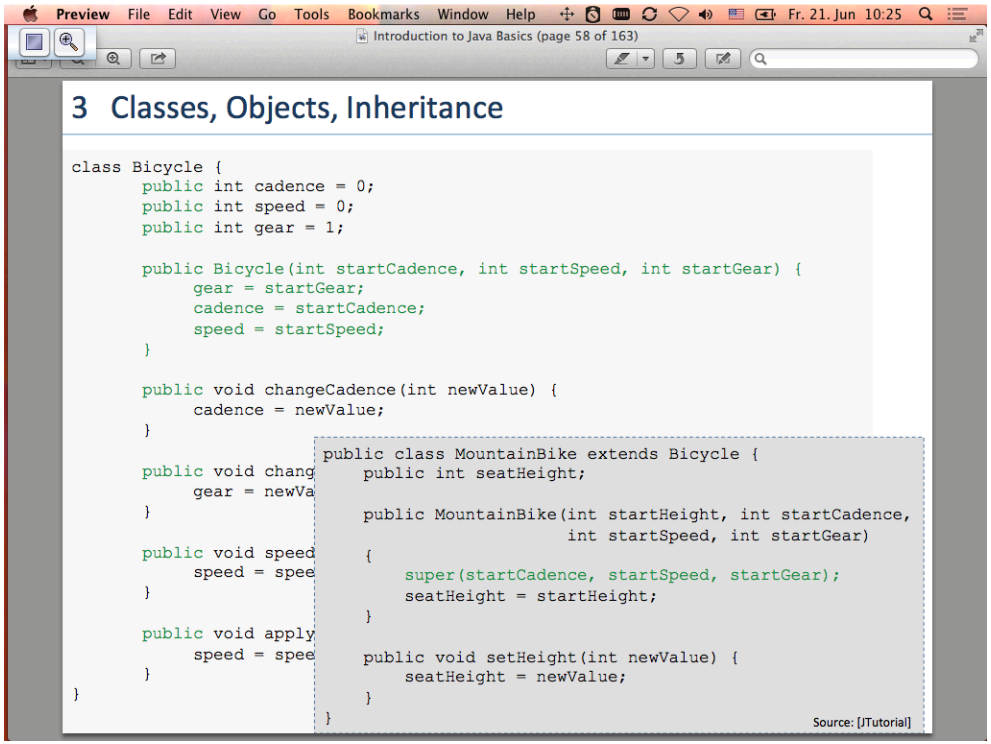

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:20
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
double[] zahlen = new double[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = 100 * Math.random();
}
int heiner = (int) 123.4;
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
double backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
System.out.println();
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
BubbleSort.java
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
Problems
Javadoc
Declaration
Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
46.02322101126373
68.67539556325409
50.471372781477356
25.200013263483136
62.44449461845307
49.1740536258274
52.44408558534167
72.70417559921626
13.124430157723321
26.895743870543022
13.124430157723321
Writable
Smart Insert
9 : 1

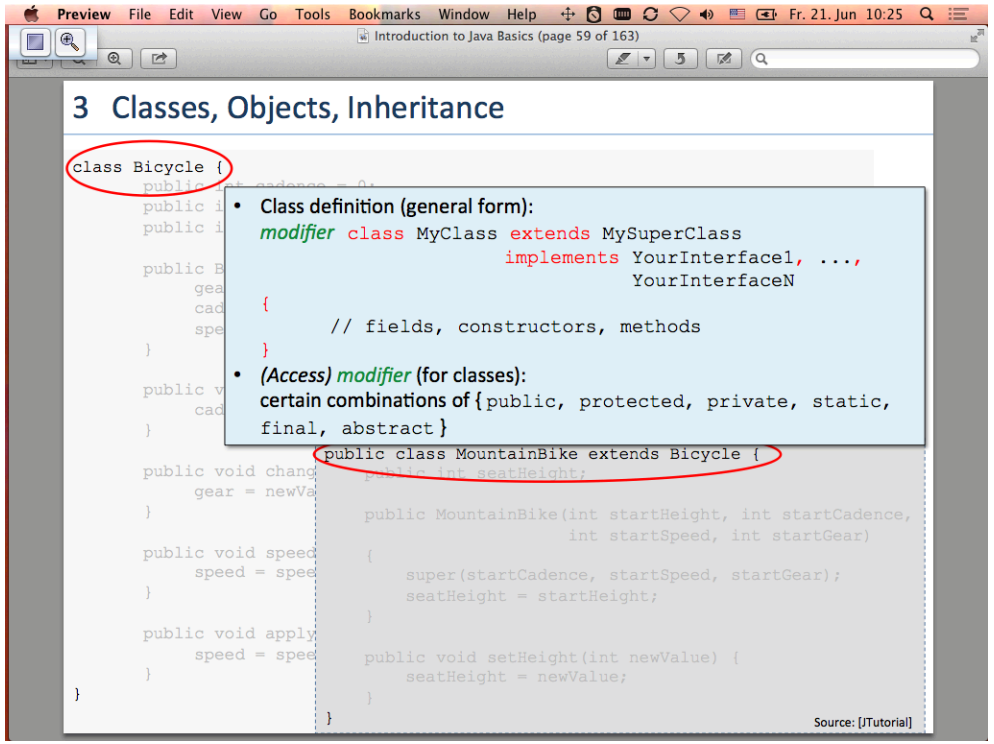

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:20
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
int[] zahlen = new int[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = (int) 100.0 * Math.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
double backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
System.out.println();
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
BubbleSort.java
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
Problems
Javadoc
Declaration
Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
46.02322101126373
68.67539556325409
50.471372781477356
25.200013263483136
62.44449461845307
49.1740536258274
52.44408558534167
72.70417559921626
13.124430157723321
26.895743870543022
13.124430157723321
Writable
Smart Insert
6 : 31


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:22
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
int[] zahlen = new int[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = (int) 100.0 * (iMath.random();
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
double backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
System.out.println();
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
BubbleSort.java
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
Problems
Javadoc
Declaration
Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
46.02322101126373
68.67539556325409
50.471372781477356
25.200013263483136
62.44449461845307
49.1740536258274
52.44408558534167
72.70417559921626
13.124430157723321
26.895743870543022
13.124430157723321
Writable
Smart Insert
6 : 41


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:22
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
BubbleSort.java
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
PotenzMain.java
*BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
int[] zahlen = new int[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = (int) (100.0 * Math.random());
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
int backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
System.out.println();
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
46.02322101126373
68.67539556325409
50.471372781477356
25.200013263483136
62.44449461845307
49.1740536258274
52.44408558534167
72.70417559921626
13.124430157723321
26.895743870543022
13.124430157723321
Writable Smart Insert 16 : 22


Preview File Edit View Go Tools Bookmarks Window Help Fr. 21. Jun 10:24
Introduction to Java Basics (page 57 of 163)
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
Preview


Preview File Edit View Go Tools Bookmarks Window Help Fr. 21. Jun 10:25
Introduction to Java Basics (page 58 of 163)
3 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]


Preview File Edit View Go Tools Bookmarks Window Help Fr. 21. Jun 10:25
Introduction to Java Basics (page 59 of 163)
3 Classes, Objects, Inheritance
class Bicycle {
• Class definition (general form):
modifier class MyClass extends MySuperClass
implements YourInterface1, ...,
YourInterfaceN
{
// fields, constructors, methods
}
• (Access) modifier (for classes):
certain combinations of { public, protected, private, static,
final, abstract }
public class MountainBike extends Bicycle {
Source: [JTutorial]

3 Classes, Objects, Inheritance

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

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

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

Source: [JTutorial]

3 Classes, Objects, Inheritance

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

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

Source: [JTutorial]

3 Classes, Objects, Inheritance

- Constructor declaration (general form):
  modifier `MyClass` (*parameter**) *throwsClause*
  {
  *statement**
  }
- *(Access) modifier*:
  certain combinations of { `public, protected, private` }
- *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]

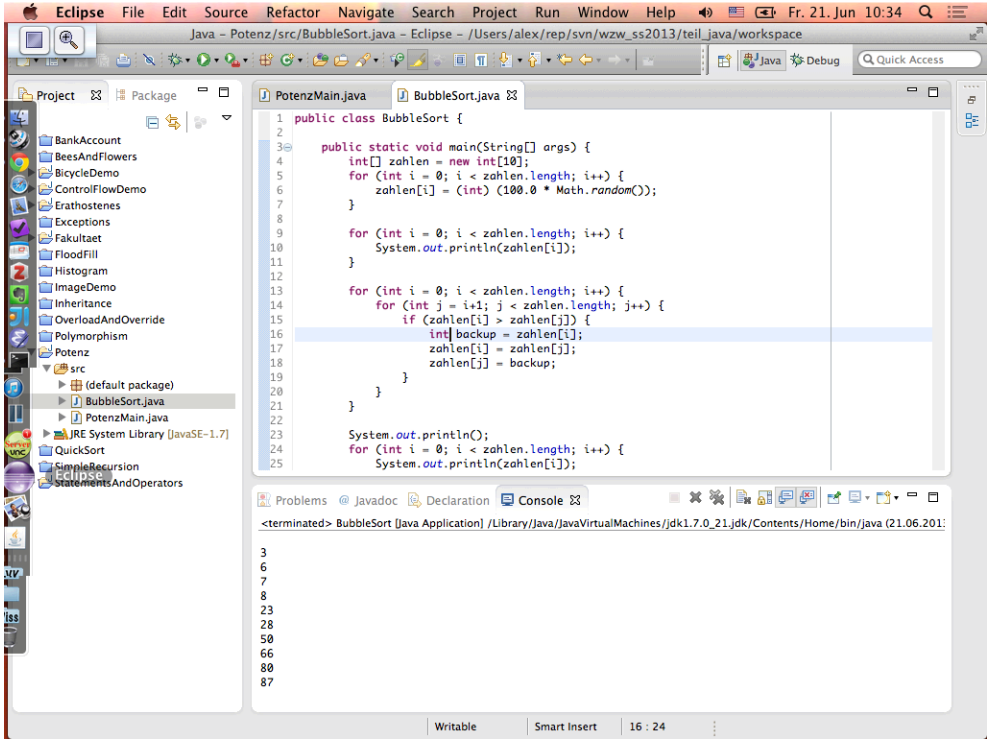
3 Classes, Objects, Inheritance

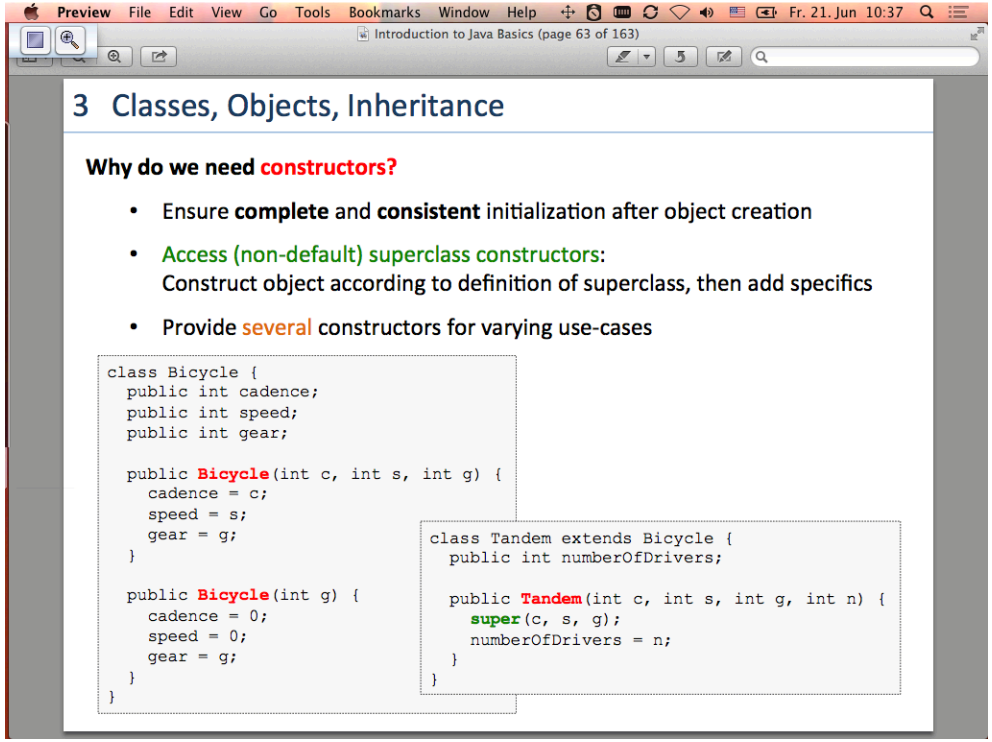
Why do we need constructors?

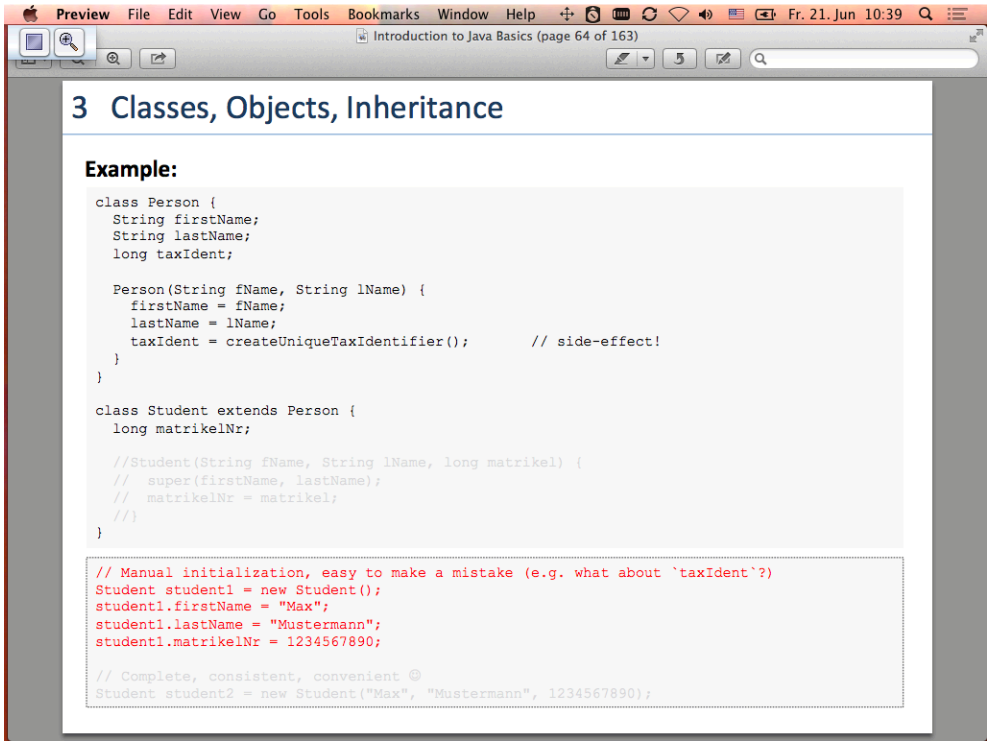
- Ensure **complete** and **consistent** initialization after object creation
- Access (non-default) superclass constructors:
  Construct object according to definition of superclass, then add specifics
- Provide several constructors for varying use-cases

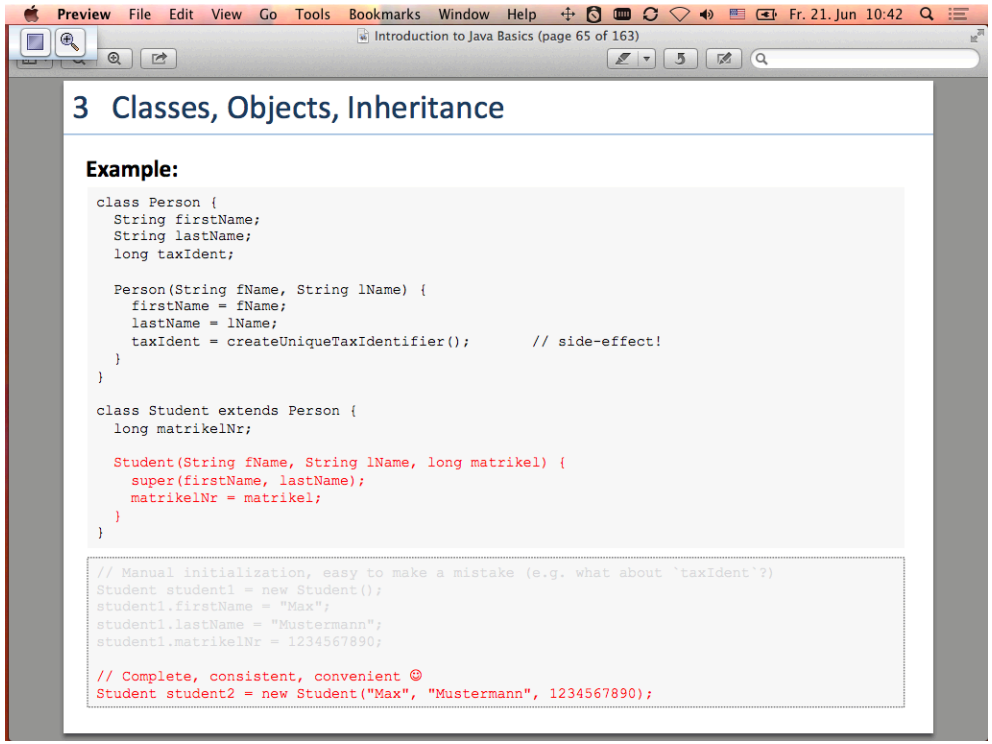
```
class Bicycle {
  public int cadence;
  public int speed;
  public int gear;

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

  public Bicycle(int g) {
    cadence = 0;
    speed = 0;
    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;
  }
}
```


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 21. Jun 10:34
Java – Potenz/src/BubbleSort.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
Potenz
src
(default package)
BubbleSort.java
PotenzMain.java
JRE System Library [JavaSE-1.7]
QuickSort
SimpleRecursion
StatementsAndOperators
PotenzMain.java
BubbleSort.java
public class BubbleSort {
public static void main(String[] args) {
int[] zahlen = new int[10];
for (int i = 0; i < zahlen.length; i++) {
zahlen[i] = (int) (100.0 * Math.random());
}
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
}
for (int i = 0; i < zahlen.length; i++) {
for (int j = i+1; j < zahlen.length; j++) {
if (zahlen[i] > zahlen[j]) {
int backup = zahlen[i];
zahlen[i] = zahlen[j];
zahlen[j] = backup;
}
}
}
System.out.println();
for (int i = 0; i < zahlen.length; i++) {
System.out.println(zahlen[i]);
Problems Javadoc Declaration Console
<terminated> BubbleSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (21.06.201
3
6
7
8
23
28
50
66
80
87
Writable Smart Insert 16 : 24


Preview File Edit View Go Tools Bookmarks Window Help Fr. 21. Jun 10:37
Introduction to Java Basics (page 63 of 163)
3 Classes, Objects, Inheritance
Why do we need constructors?
Ensure complete and consistent initialization after object creation
Access (non-default) superclass constructors:
Construct object according to definition of superclass, then add specifics
Provide several constructors for varying 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;
}
public Bicycle(int g) {
cadence = 0;
speed = 0;
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;
}
}


Preview File Edit View Go Tools Bookmarks Window Help Fr. 21. Jun 10:39
Introduction to Java Basics (page 64 of 163)
3 Classes, Objects, Inheritance
Example:
class Person {
String firstName;
String lastName;
long taxIdent;
Person(String fName, String lName) {
firstName = fName;
lastName = lName;
taxIdent = createUniqueTaxIdentifier(); // side-effect!
}
}
class Student extends Person {
long matrikelNr;
//Student(String fName, String lName, long matrikel) {
// super(firstName, lastName);
// matrikelNr = matrikel;
//}
}
// Manual initialization, easy to make a mistake (e.g. what about `taxIdent`?)
Student student1 = new Student();
student1.firstName = "Max";
student1.lastName = "Mustermann";
student1.matrikelNr = 1234567890;
// Complete, consistent, convenient ☺
Student student2 = new Student("Max", "Mustermann", 1234567890);


Preview File Edit View Go Tools Bookmarks Window Help Fr. 21. Jun 10:42
Introduction to Java Basics (page 65 of 163)
3 Classes, Objects, Inheritance
Example:
class Person {
String firstName;
String lastName;
long taxIdent;
Person(String fName, String lName) {
firstName = fName;
lastName = lName;
taxIdent = createUniqueTaxIdentifier(); // side-effect!
}
}
class Student extends Person {
long matrikelNr;
Student(String fName, String lName, long matrikel) {
super(firstName, lastName);
matrikelNr = matrikel;
}
}
// Manual initialization, easy to make a mistake (e.g. what about `taxIdent`?)
Student student1 = new Student();
student1.firstName = "Max";
student1.lastName = "Mustermann";
student1.matrikelNr = 1234567890;
// Complete, consistent, convenient ☺
Student student2 = new Student("Max", "Mustermann", 1234567890);