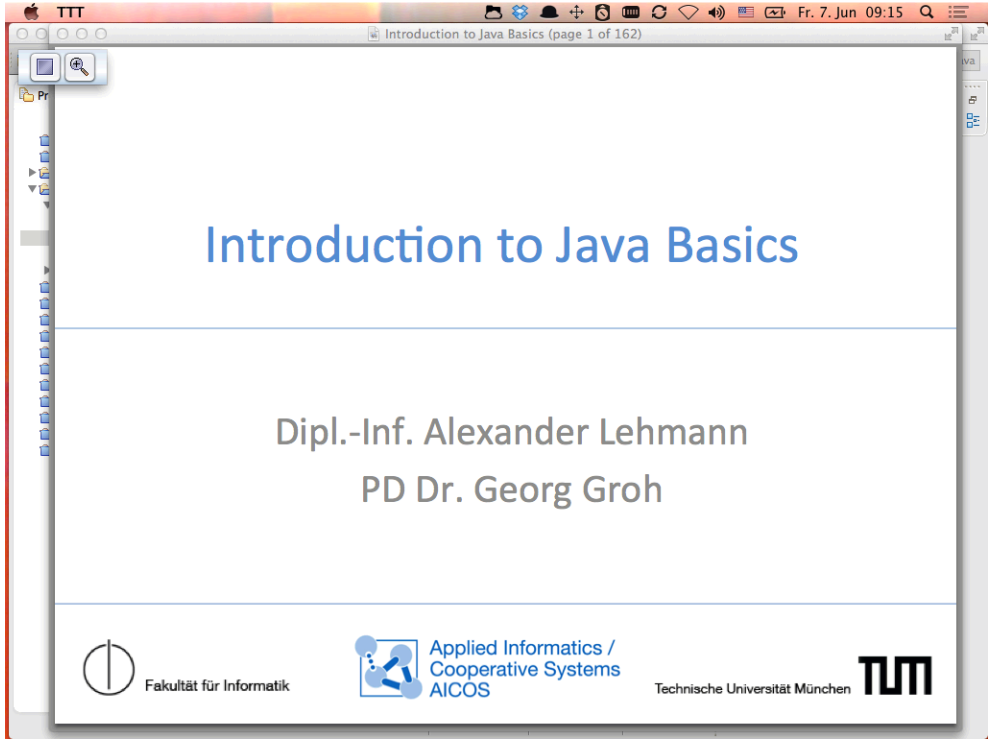
Script generated by TTT

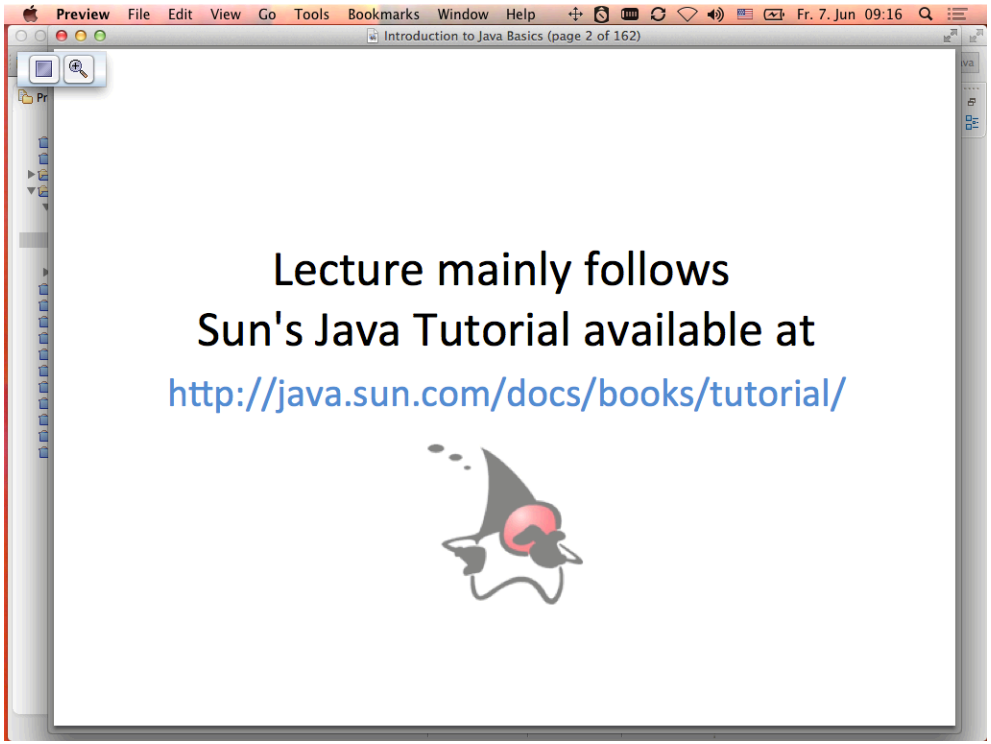
Title: Lehmann: Uebung_Einf_HF (07.06.2013)

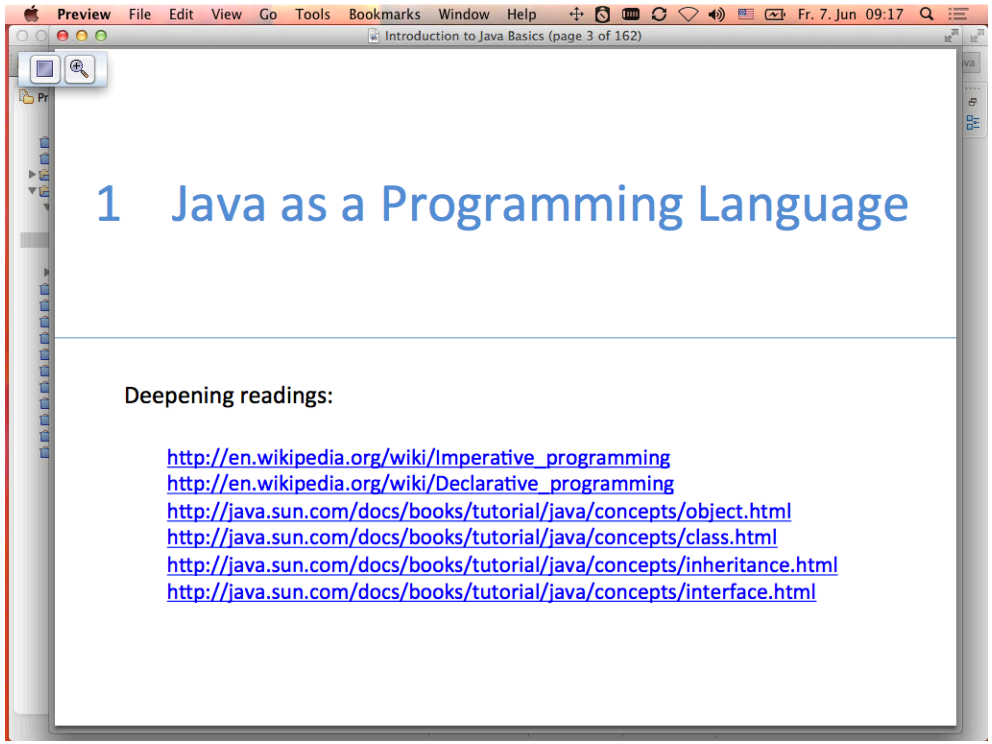
Date: Fri Jun 07 09:15:41 CEST 2013

Duration: 85:03 min

Pages: 57

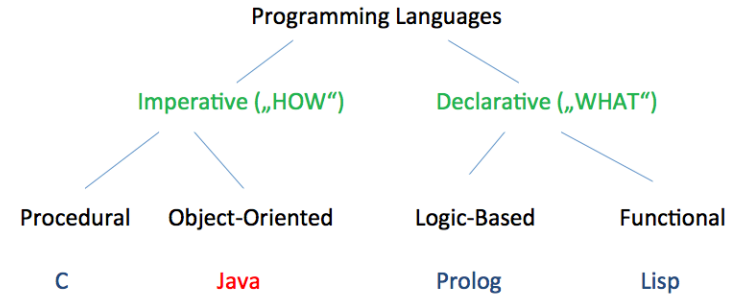




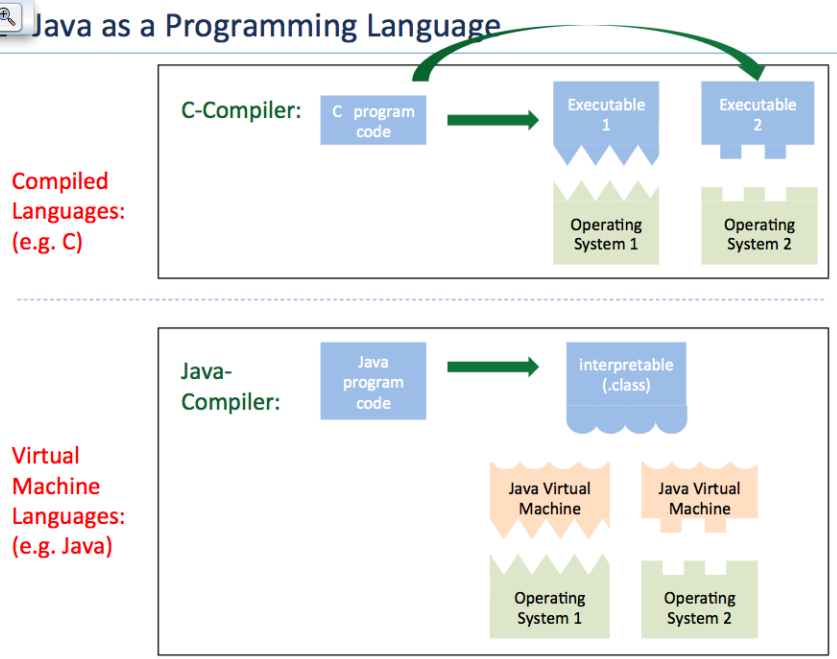


Java as a Programming Language

Programming Languages



Java as a Programming Language



Java as a Programming Language

Imperative Programming

- Imperative program:
  Sequence of statements
- Instructions change state (especially memory) of computer system

control flow

```
boolean plato;
int horst;
int heiner;
int fritz;
plato = false;
horst = 101;
heiner = 2;
fritz = horst + heiner;
horst = 2000;
```

memory (simplified model)

cell nr	cell name	cell content	
1123	plato	false	data
1124			
1125	horst	101	
1126	heiner	0	
1127			
1128	fritz	0	
4027		boolean plato;	instructions
4028		int horst;	
4029		int heiner;	
4030		int fritz;	
4029		plato = false;	
4030		horst = 101;	

Java as a Programming Language

Imperative Programming

- Imperative program:
  Sequence of statements
- Instructions change state (especially memory) of computer system

control flow

```
boolean plato;
int horst;
int heiner;
int fritz;
plato = false;
horst = 101;
heiner = 2;
fritz = horst + heiner;
horst = 2000;
```

memory (simplified model)

cell nr	cell name	cell content	
1123	plato	false	data
1124			
1125	horst	2000	
1126	heiner	2	
1127			
1128	fritz	103	
4027		boolean plato;	instructions
4028		int horst;	
4029		int heiner;	
4030		int fritz;	
4029		plato = false;	
4030		horst = 101;	

Java as a Programming Language

Procedural Programming

- Group sequences of instructions into named „procedures“ („functions“, „methods“, „sub-routines“ etc.)

```
int doSelfSumSquare(int someNumber){
        int a;
        a = someNumber + someNumber;
        a = a * a;
        return a;
}
```

$f(x) = (x + x)^2$

- Advantages
  - no copying of instruction sequences
  - better testing
  - modularity (e.g. code change inside procedure doesn't affect caller)
  - code re-use
  - etc.

Java as a Programming Language

```
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = horst + 2;
```

⋮

```
int doSelfSumSquare(int someNumber){
        int a;
        a = someNumber + someNumber;
        a = a * a;
        return a;
}
```

- In the example: Control flow is transferred to procedure, back to main program, back to procedure and back to main program

Java as a Programming Language

```
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = horst + 2;
```

⋮

```
int doSelfSumSquare(int someNumber){
        int a;
        a = someNumber + someNumber;
        a = a * a;
        return a;
}
```

- In the example: Control flow is transferred to procedure, back to main program, back to procedure and back to main program

Java as a Programming Language

```
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = horst + 2;
```

⋮

```
int doSelfSumSquare(int someNumber){
        int a;
        a = someNumber + someNumber;
        a = a * a;
        return a;
}
```

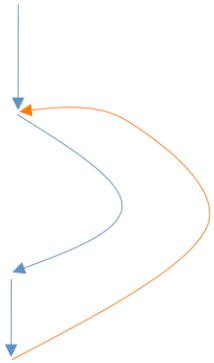
- In the example: Control flow is transferred to procedure, back to main program, back to procedure and back to main program

Java as a Programming Language

```
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = horst + 2;
```

⋮

```
int doSelfSumSquare(int someNumber){
        int a;
        a = someNumber + someNumber;
        a = a * a;
        return a;
}
```



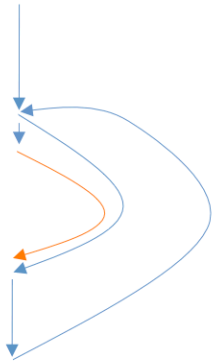
- In the example: Control flow is transferred to procedure, back to main program, back to procedure and back to main program

Java as a Programming Language

```
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = horst + 2;
```

⋮

```
int doSelfSumSquare(int someNumber){
        int a;
        a = someNumber + someNumber;
        a = a * a;
        return a;
}
```



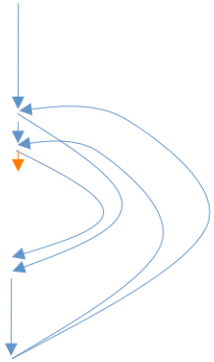
- In the example: Control flow is transferred to procedure, back to main program, back to procedure and back to main program

Java as a Programming Language

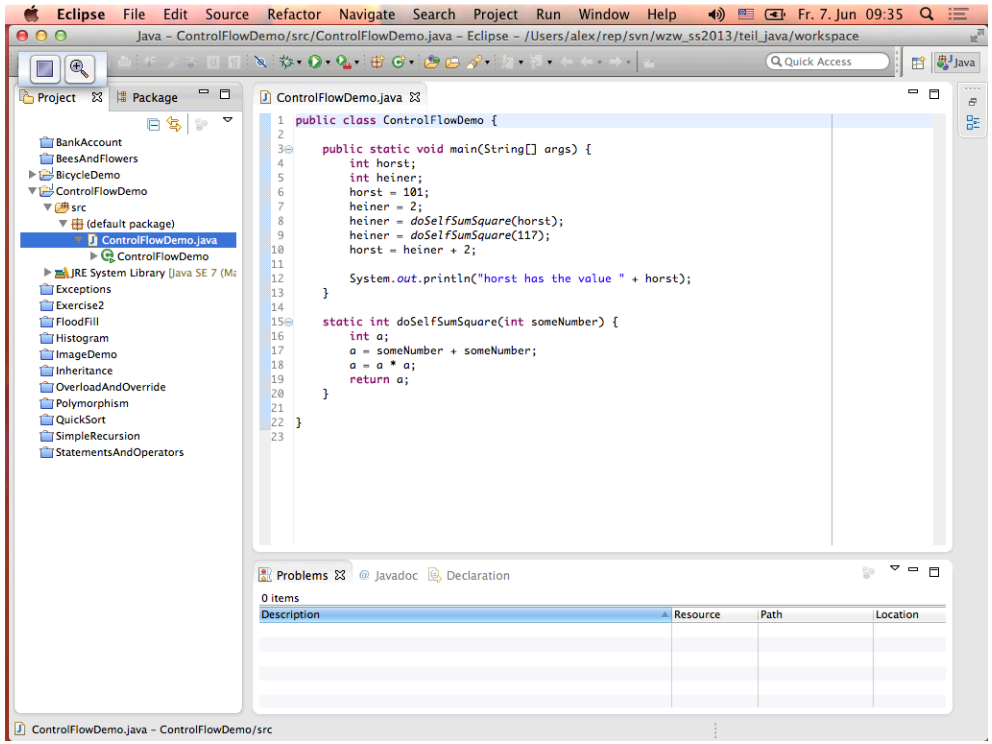
```
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = horst + 2;
```

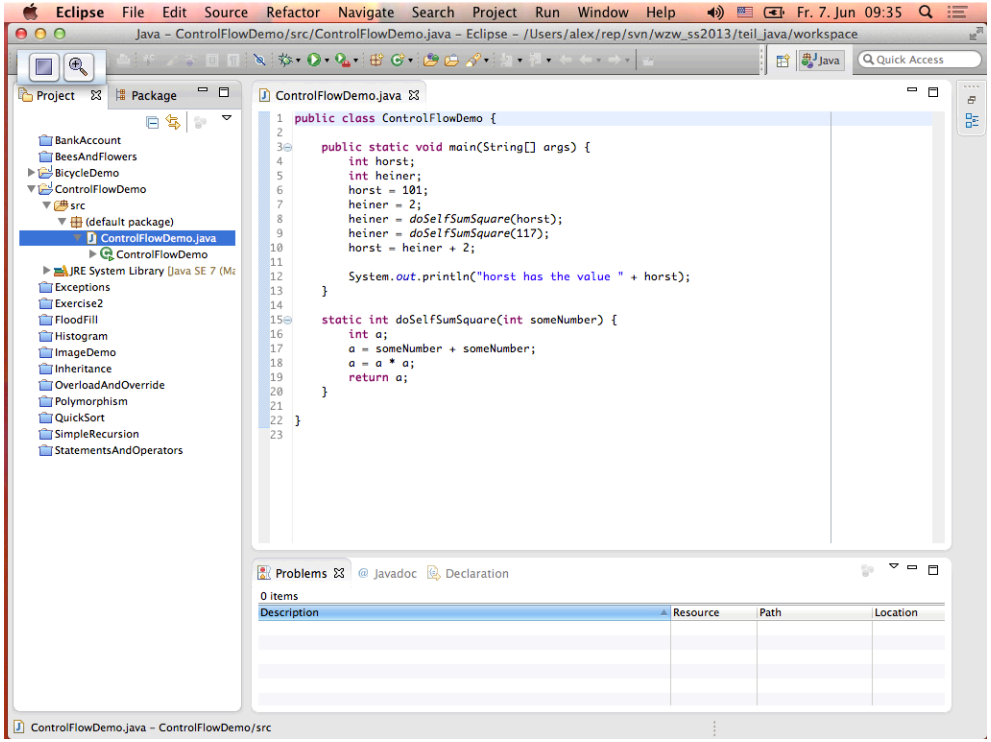
⋮

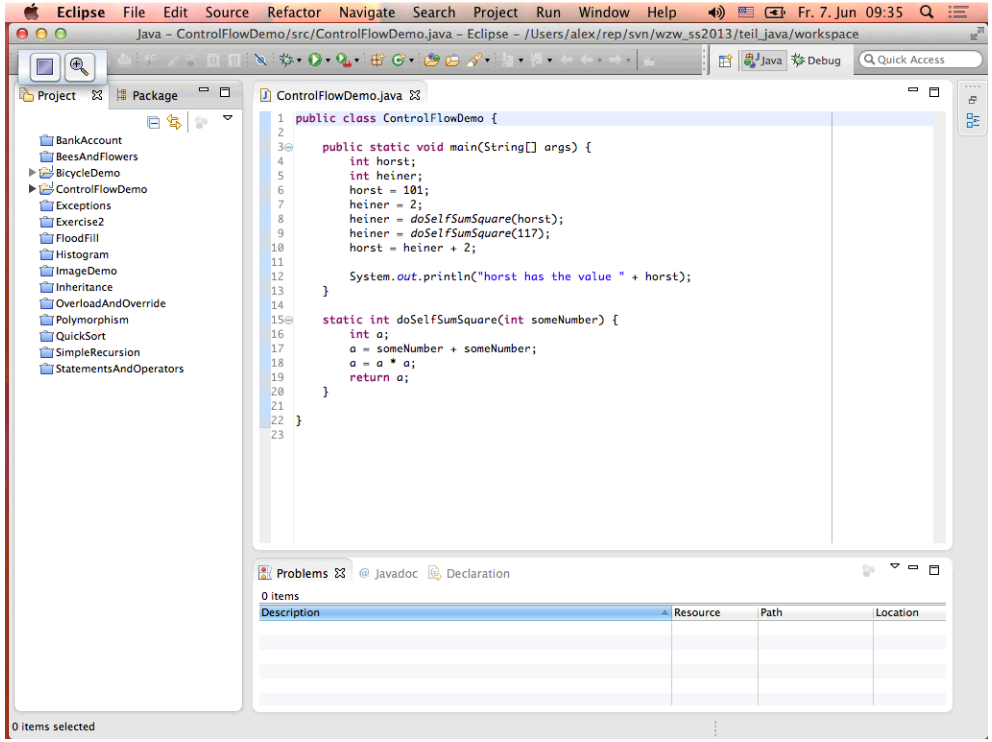
```
int doSelfSumSquare(int someNumber){
        int a;
        a = someNumber + someNumber;
        a = a * a;
        return a;
}
```

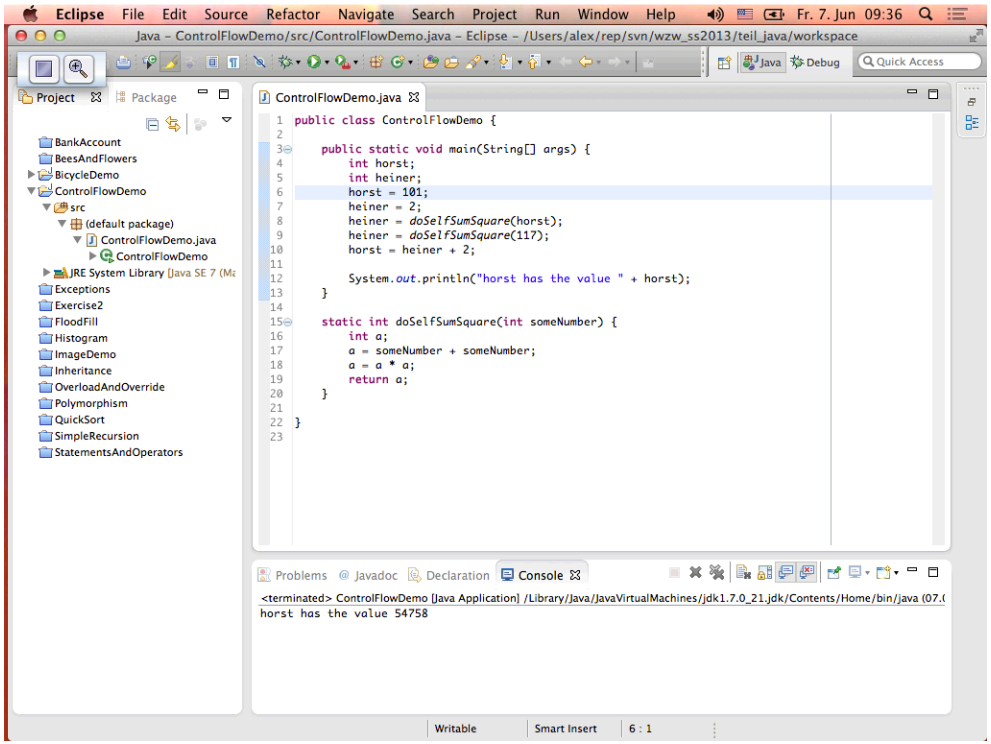


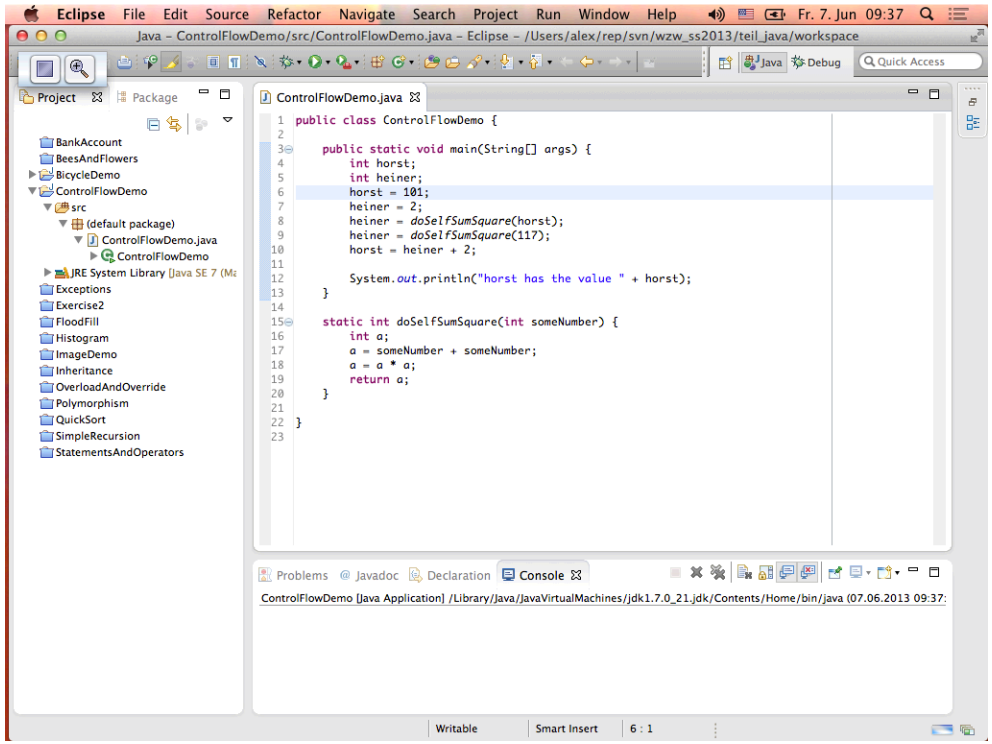
- In the example: Control flow is transferred to procedure, back to main program, back to procedure and back to main program

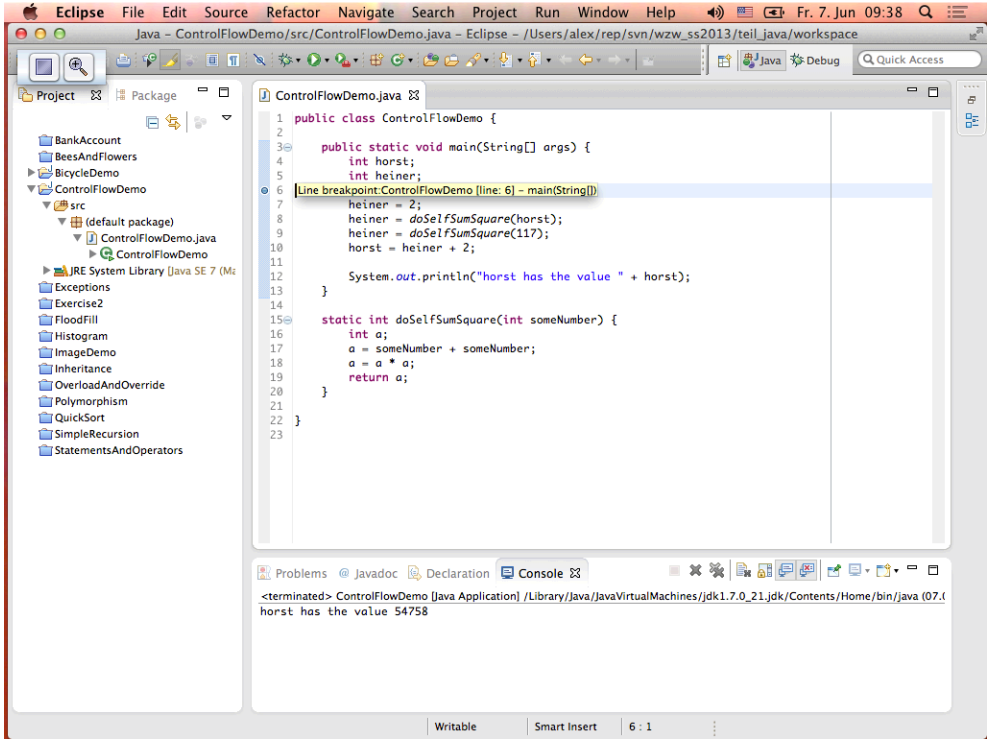


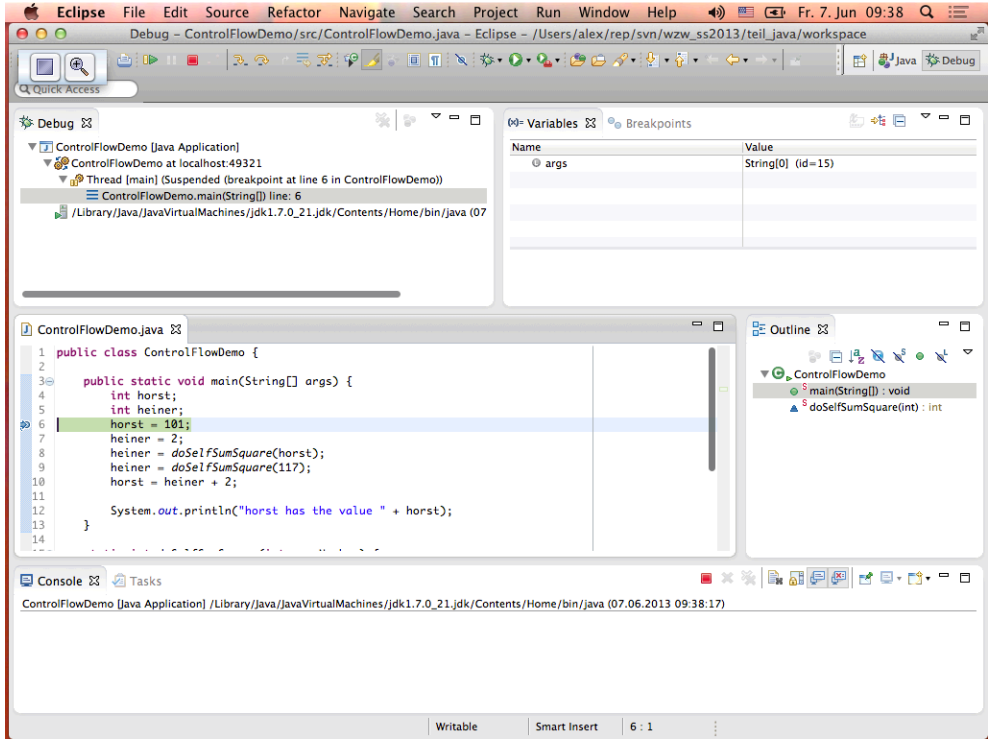

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:35
Java – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
src
(default package)
ControlFlowDemo.java
ControlFlowDemo
JRE System Library [Java SE 7 (Ma
Exceptions
Exercise2
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
ControlFlowDemo.java
public class ControlFlowDemo {
public static void main(String[] args) {
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = heiner + 2;
System.out.println("horst has the value " + horst);
}
static int doSelfSumSquare(int someNumber) {
int a;
a = someNumber + someNumber;
a = a * a;
return a;
}
}
Problems
Javadoc
Declaration
0 items
Description
Resource
Path
Location
ControlFlowDemo.java – ControlFlowDemo/src

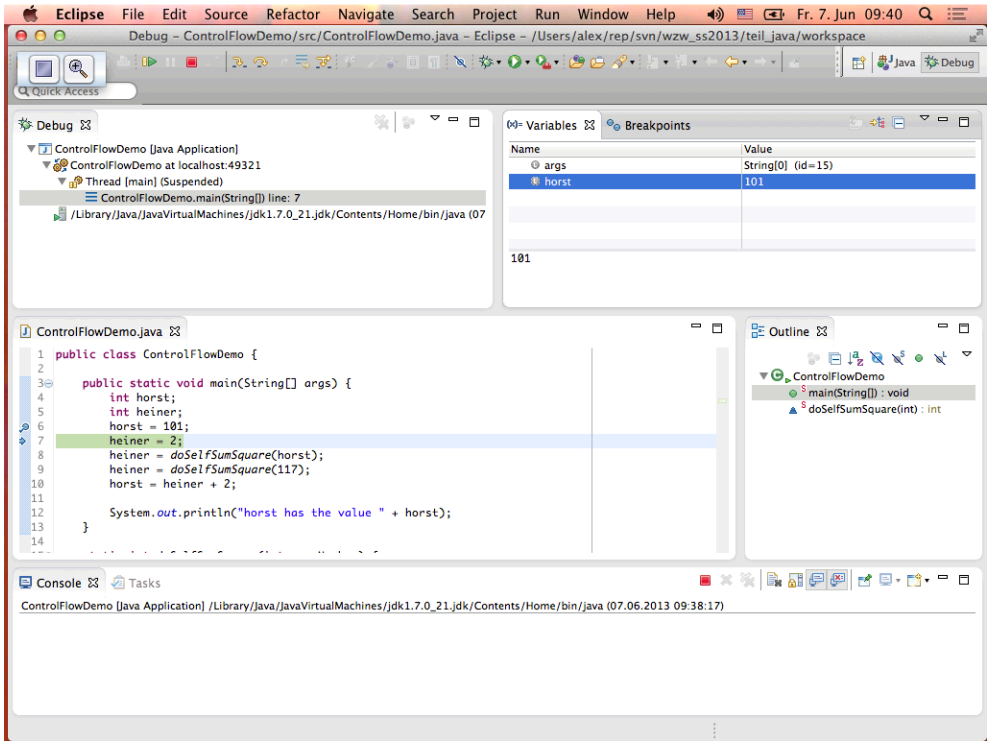

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:35
Java – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Java
Debug
Quick Access
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Exceptions
Exercise2
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
ControlFlowDemo.java
public class ControlFlowDemo {
public static void main(String[] args) {
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = heiner + 2;
System.out.println("horst has the value " + horst);
}
static int doSelfSumSquare(int someNumber) {
int a;
a = someNumber + someNumber;
a = a * a;
return a;
}
}
Problems
Javadoc
Declaration
0 items
Description
Resource
Path
Location
0 items selected

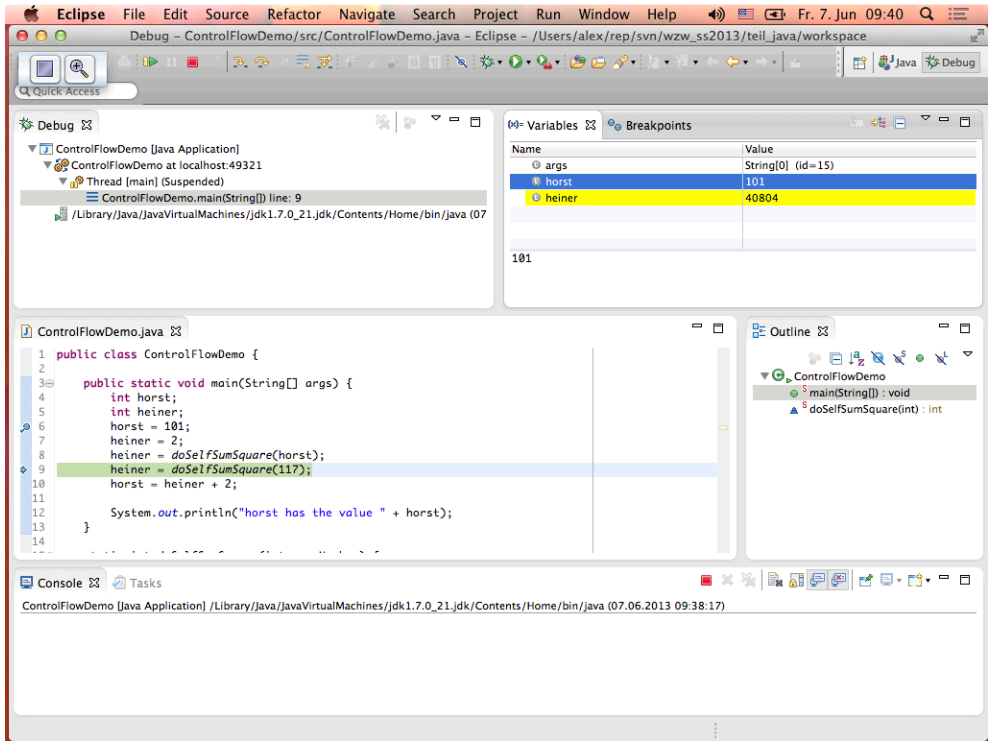

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:36
Java – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Java
Debug
Quick Access
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
src
(default package)
ControlFlowDemo.java
ControlFlowDemo
JRE System Library [Java SE 7 (Ma
Exceptions
Exercise2
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
ControlFlowDemo.java
public class ControlFlowDemo {
public static void main(String[] args) {
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = heiner + 2;
System.out.println("horst has the value " + horst);
}
static int doSelfSumSquare(int someNumber) {
int a;
a = someNumber + someNumber;
a = a * a;
return a;
}
}
Problems
Javadoc
Declaration
Console
<terminated> ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.
horst has the value 54758
Writable
Smart Insert
6 : 1

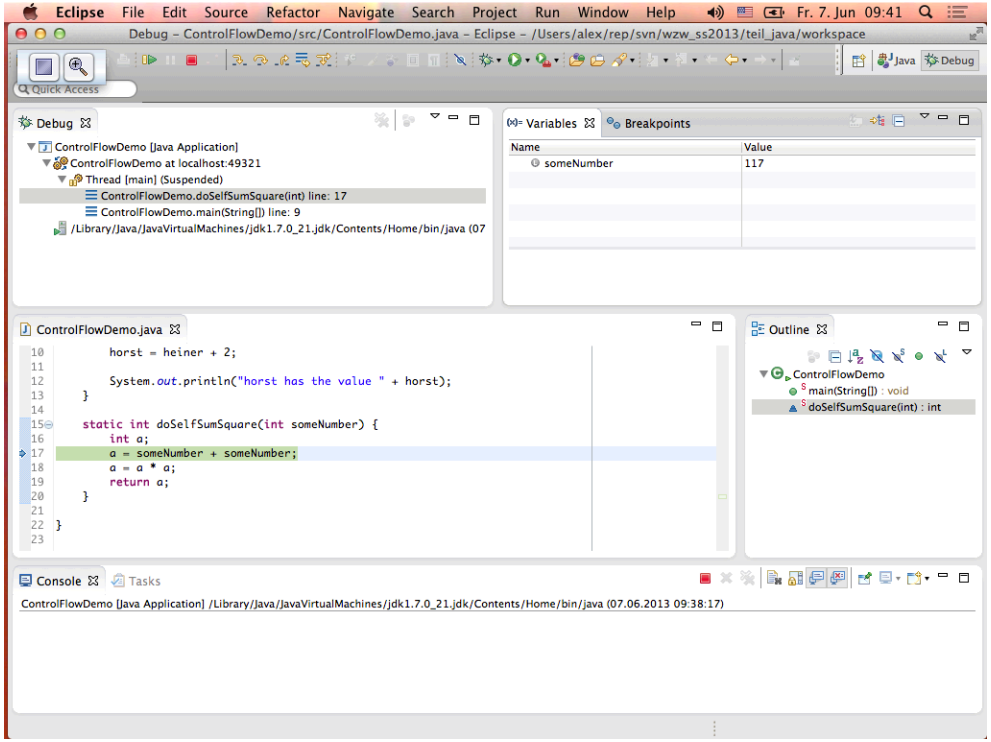

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:37
Java – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Java
Debug
Quick Access
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
src
(default package)
ControlFlowDemo.java
ControlFlowDemo
JRE System Library [Java SE 7 (Ma
Exceptions
Exercise2
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
ControlFlowDemo.java
public class ControlFlowDemo {
public static void main(String[] args) {
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = heiner + 2;
System.out.println("horst has the value " + horst);
}
static int doSelfSumSquare(int someNumber) {
int a;
a = someNumber + someNumber;
a = a * a;
return a;
}
}
Problems
Javadoc
Declaration
Console
ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:37:
Writable
Smart Insert
6 : 1

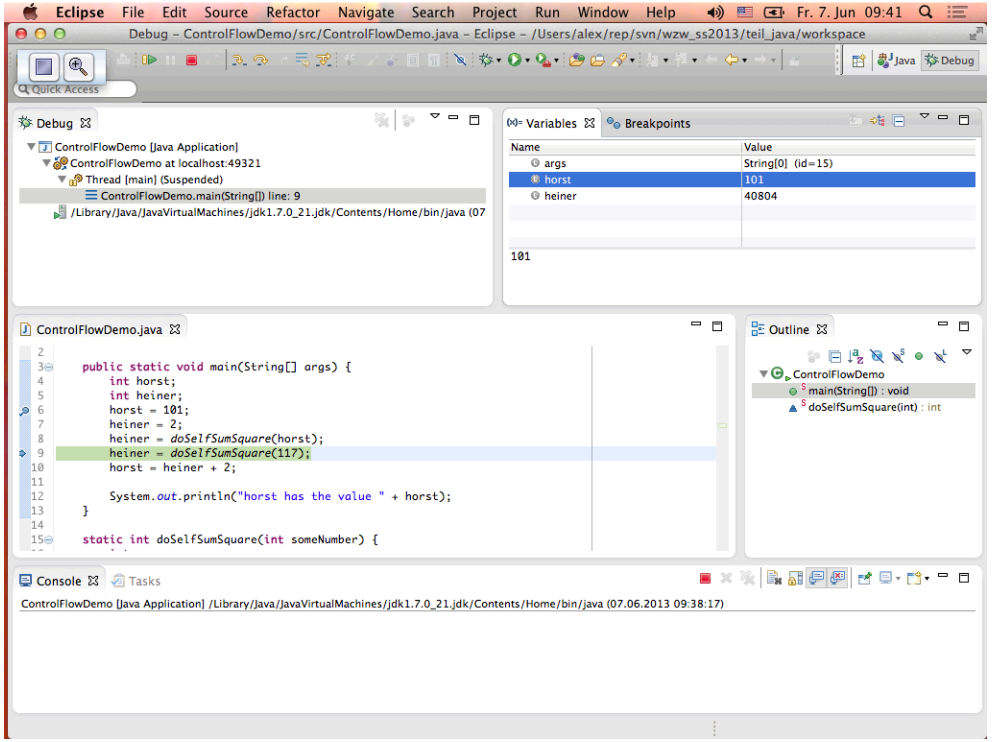

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:38
Java – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
src
(default package)
ControlFlowDemo.java
ControlFlowDemo
JRE System Library [Java SE 7 (Ma
Exceptions
Exercise2
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
ControlFlowDemo.java
1 public class ControlFlowDemo {
2
3 public static void main(String[] args) {
4 int horst;
5 int heiner;
6 Line breakpoint:ControlFlowDemo [line: 6] - main(String[])
7 heiner = 2;
8 heiner = doSelfSumSquare(horst);
9 heiner = doSelfSumSquare(117);
10 horst = heiner + 2;
11
12 System.out.println("horst has the value " + horst);
13 }
14
15 static int doSelfSumSquare(int someNumber) {
16 int a;
17 a = someNumber + someNumber;
18 a = a * a;
19 return a;
20 }
21
22 }
23
Problems Javadoc Declaration Console
<terminated> ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.
horst has the value 54758
Writable Smart Insert 6 : 1

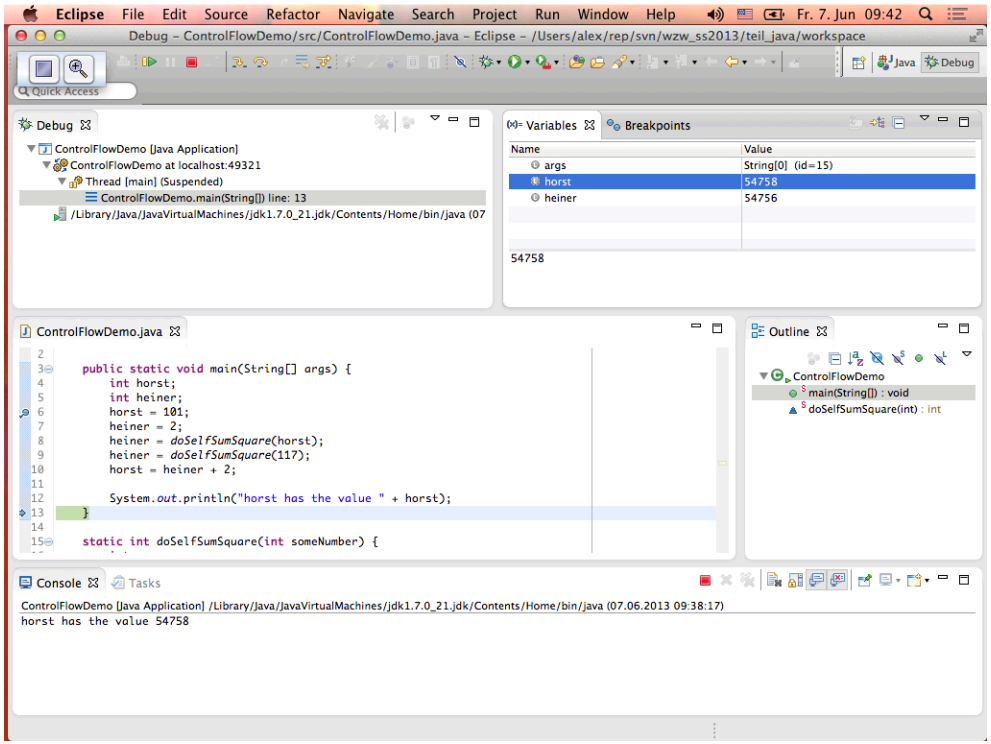

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:38
Debug – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
ControlFlowDemo [Java Application]
ControlFlowDemo at localhost:49321
Thread [main] (Suspended (breakpoint at line 6 in ControlFlowDemo))
ControlFlowDemo.main(String[]) line: 6
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07
Variables
Breakpoints
Name
Value
args
String[0] (id=15)
ControlFlowDemo.java
1 public class ControlFlowDemo {
2
3 public static void main(String[] args) {
4 int horst;
5 int heiner;
6 horst = 101;
7 heiner = 2;
8 heiner = doSelfSumSquare(horst);
9 heiner = doSelfSumSquare(117);
10 horst = heiner + 2;
11
12 System.out.println("horst has the value " + horst);
13 }
14
Outline
ControlFlowDemo
main(String[]) : void
doSelfSumSquare(int) : int
Console
Tasks
ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:38:17)
Writable Smart Insert 6 : 1

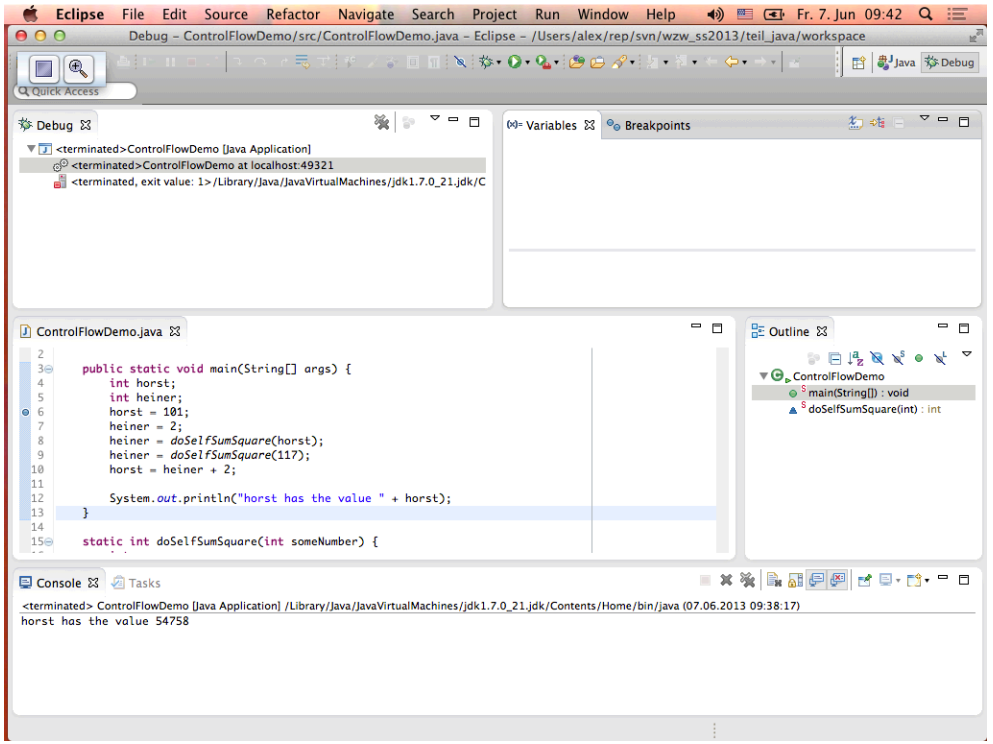

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:40
Debug – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
ControlFlowDemo [Java Application]
ControlFlowDemo at localhost:49321
Thread [main] (Suspended)
ControlFlowDemo.main(String[]) line: 7
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07
Variables
Breakpoints
Name
Value
args
String[0] (id=15)
horst
101
101
ControlFlowDemo.java
1 public class ControlFlowDemo {
2
3 public static void main(String[] args) {
4 int horst;
5 int heiner;
6 horst = 101;
7 heiner = 2;
8 heiner = doSelfSumSquare(horst);
9 heiner = doSelfSumSquare(117);
10 horst = heiner + 2;
11
12 System.out.println("horst has the value " + horst);
13 }
14
Outline
ControlFlowDemo
main(String[]) : void
doSelfSumSquare(int) : int
Console
Tasks
ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:38:17)


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:40
Debug – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
ControlFlowDemo [Java Application]
ControlFlowDemo at localhost:49321
Thread [main] (Suspended)
ControlFlowDemo.main(String[]) line: 9
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07
Variables
Breakpoints
Name
Value
args
String[0] (id=15)
horst
101
heiner
40804
101
ControlFlowDemo.java
1 public class ControlFlowDemo {
2
3 public static void main(String[] args) {
4 int horst;
5 int heiner;
6 horst = 101;
7 heiner = 2;
8 heiner = doSelfSumSquare(horst);
9 heiner = doSelfSumSquare(117);
10 horst = heiner + 2;
11
12 System.out.println("horst has the value " + horst);
13 }
14
Outline
ControlFlowDemo
main(String[]) : void
doSelfSumSquare(int) : int
Console
Tasks
ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:38:17)


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:41
Debug – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
ControlFlowDemo [Java Application]
ControlFlowDemo at localhost:49321
Thread [main] (Suspended)
ControlFlowDemo.doSelfSumSquare(int) line: 17
ControlFlowDemo.main(String[]) line: 9
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07
Variables
Breakpoints
Name
Value
someNumber
117
ControlFlowDemo.java
horst = heiner + 2;
System.out.println("horst has the value " + horst);
}
static int doSelfSumSquare(int someNumber) {
int a;
a = someNumber + someNumber;
a = a * a;
return a;
}
Outline
ControlFlowDemo
main(String[]) : void
doSelfSumSquare(int) : int
Console
Tasks
ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:38:17)


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:41
Debug – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
ControlFlowDemo [Java Application]
ControlFlowDemo at localhost:49321
Thread [main] (Suspended)
ControlFlowDemo.main(String[]) line: 9
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07
Variables
Breakpoints
Name
Value
args
String[0] (id=15)
horst
101
heiner
40804
101
ControlFlowDemo.java
public static void main(String[] args) {
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = heiner + 2;
System.out.println("horst has the value " + horst);
}
static int doSelfSumSquare(int someNumber) {
Outline
ControlFlowDemo
main(String[]) : void
doSelfSumSquare(int) : int
Console
Tasks
ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:38:17)


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:42
Debug – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
ControlFlowDemo [Java Application]
ControlFlowDemo at localhost:49321
Thread [main] (Suspended)
ControlFlowDemo.main(String[]) line: 13
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07
Variables
Breakpoints
Name
Value
args
String[0] (id=15)
horst
54758
heiner
54756
54758
ControlFlowDemo.java
public static void main(String[] args) {
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = heiner + 2;
System.out.println("horst has the value " + horst);
}
static int doSelfSumSquare(int someNumber) {
Outline
ControlFlowDemo
main(String[]) : void
doSelfSumSquare(int) : int
Console
Tasks
ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:38:17)
horst has the value 54758


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:42
Debug – ControlFlowDemo/src/ControlFlowDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
<terminated>ControlFlowDemo [Java Application]
<terminated>ControlFlowDemo at localhost:49321
<terminated, exit value: 1>/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/C
Variables
Breakpoints
ControlFlowDemo.java
public static void main(String[] args) {
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = heiner + 2;
System.out.println("horst has the value " + horst);
}
static int doSelfSumSquare(int someNumber) {
Outline
ControlFlowDemo
main(String[]) : void
doSelfSumSquare(int) : int
Console
Tasks
<terminated> ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:38:17)
horst has the value 54758

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 7. Jun 09:42

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

Project Package

- BankAccount
- BeesAndFlowers
- BicycleDemo
- ControlFlowDemo
  - src
    - (default package)
      - ControlFlowDemo.java
        - ControlFlowDemo
  - JRE System Library [Java SE 7 (Ma
- Exceptions
- Exercise2
- FloodFill
- Histogram
- ImageDemo
- Inheritance
- OverloadAndOverride
- Polymorphism
- QuickSort
- SimpleRecursion
- StatementsAndOperators

ControlFlowDemo.java

```
1  public class ControlFlowDemo {
2
3      public static void main(String[] args) {
4          int horst;
5          int heiner;
6          horst = 101;
7          heiner = 2;
8          heiner = doSelfSumSquare(horst);
9          heiner = doSelfSumSquare(117);
10         horst = heiner + 2;
11
12         System.out.println("horst has the value " + horst);
13     }
14
15     static int doSelfSumSquare(int someNumber) {
16         int a;
17         a = someNumber + someNumber;
18         a = a * a;
19         return a;
20     }
21
22 }
23
```

Problems Javadoc Declaration Console

<terminated> ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.(

horst has the value 54758

Writable Smart Insert 13 : 1

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Fr. 7. Jun 09:43

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

Project Package

- BankAccount
- BeesAndFlowers
- BicycleDemo
- ControlFlowDemo
  - src
    - (default package)
      - ControlFlowDemo.java
        - ControlFlowDemo
  - JRE System Library [Java SE 7 (Ma
- Exceptions
- Exercise2
- FloodFill
- Histogram
- ImageDemo
- Inheritance
- OverloadAndOverride
- Polymorphism
- QuickSort
- SimpleRecursion
- StatementsAndOperators

ControlFlowDemo.java

```
1  public class ControlFlowDemo {
2
3      public static void main(String[] args) {
4          int horst;
5          int heiner;
6          horst = 101;
7          heiner = 2;
8          heiner = doSelfSumSquare(horst);
9          heiner = doSelfSumSquare(117);
10         horst = heiner + 2;
11
12         System.out.println("horst has the value " + horst);
13     }
14
15     static int doSelfSumSquare(int someNumber) {
16         int a;
17         a = someNumber + someNumber;
18         a = a * a;
19         return a;
20     }
21
22 }
23
```

Problems Javadoc Declaration Console

<terminated> ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.(

horst has the value 54758

Writable Smart Insert 13 : 1

Preview File Edit View Go Tools Bookmarks Window Help Fr. 7. Jun 09:43

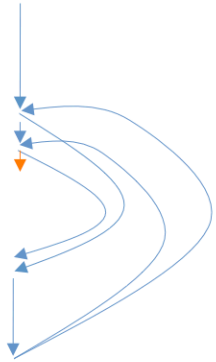
Introduction to Java Basics (page 17 of 162)

Java as a Programming Language

```
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = horst + 2;
```

⋮

```
int doSelfSumSquare(int someNumber){
        int a;
        a = someNumber + someNumber;
        a = a * a;
        return a;
```



- In the example: Control flow is transferred to procedure, back to main program, back to procedure and back to main program

Preview File Edit View Go Tools Bookmarks Window Help Fr. 7. Jun 09:43

Introduction to Java Basics (page 18 of 162)

Java as a Programming Language

```
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = horst + 2;
```

⋮

```
int doSelfSumSquareHeiner(int someNumber){

        int a;
        a = someNumber + someNumber;
        a = a * a;
        a = a * heiner;
        return a;
}
```

- Procedures often access global variables (bad style!)
- Goal: „Keep things local“ (better testing, code re-use etc.)

Introduction to Java Basics (page 19 of 162)

Java as a Programming Language

Object-oriented Programming

- Object-oriented programming:

  Group data and procedures into objects ←→
  Models of state and behaviour of **real world** objects
  state „fields“ ; behaviour „methods“

- Methods should mainly act on an object's fields
- Classes: Blueprints for objects → Objects: Instances of classes
- Advantages
  - Intuitive models
  - Information hiding
  - Increased modularity, locality etc.
  - Increased code re-use
  - etc.

Introduction to Java Basics (page 20 of 162)

Java as a Programming Language

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

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

        void changeGear(int newValue) {
             gear = newValue;
        }

        void speedUp(int increment) {
             speed = speed + increment;
        }

        void applyBrakes(int decrement) {
             speed = speed - decrement;
        }
}
```

fields (state)

methods (behaviour)

class

Source: [JTutorial]

Introduction to Java Basics (page 21 of 162)

Java as a Programming Language

```
class BicycleDemo {
     public static void main(String[] args) {
          // Create two different Bicycle objects
          Bicycle bike1 = new Bicycle();
          Bicycle bike2 = new Bicycle();

          // Invoke methods on these objects
          bike1.changeCadence(50);
          bike1.speedUp(10);
          bike1.changeGear(2);

          bike2.changeCadence(50);
          bike2.speedUp(10);
          bike2.changeGear(2);
          bike2.changeCadence(40);
          bike2.speedUp(10);
          bike2.changeGear(3);
        }
}
```

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

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

        void changeGear(int newValue) {
             gear = newValue;
        }

        void speedUp(int increment) {
             speed = speed + increment;
        }

        void applyBrakes(int decrement) {
             speed = speed - decrement;
        }
}
```

Source: [JTutorial]

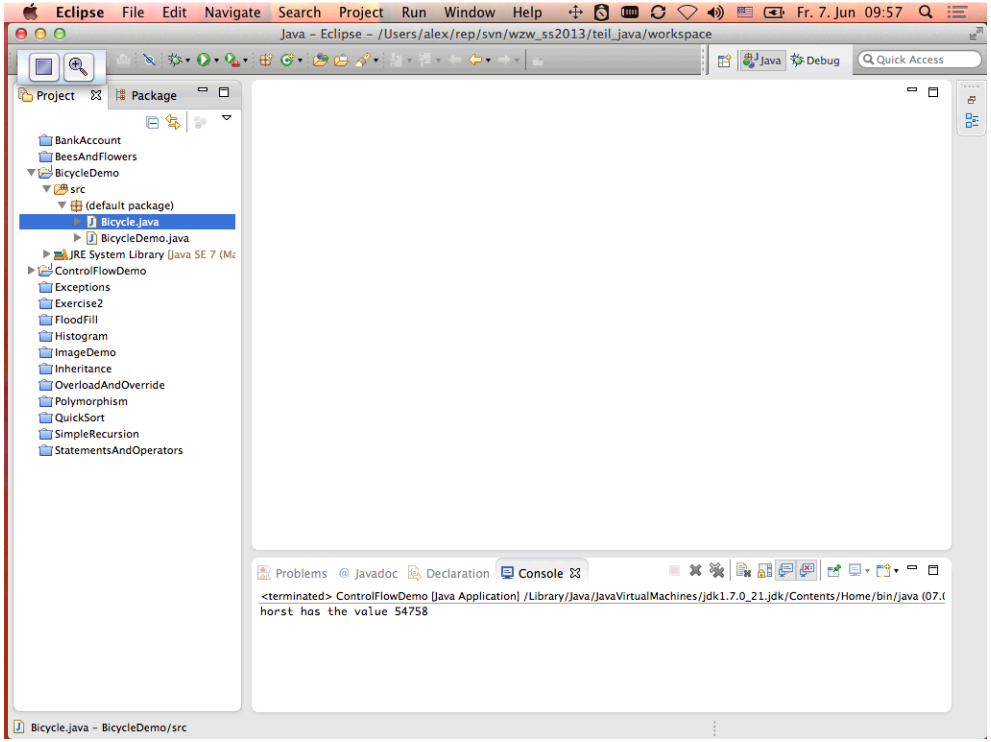
Java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace

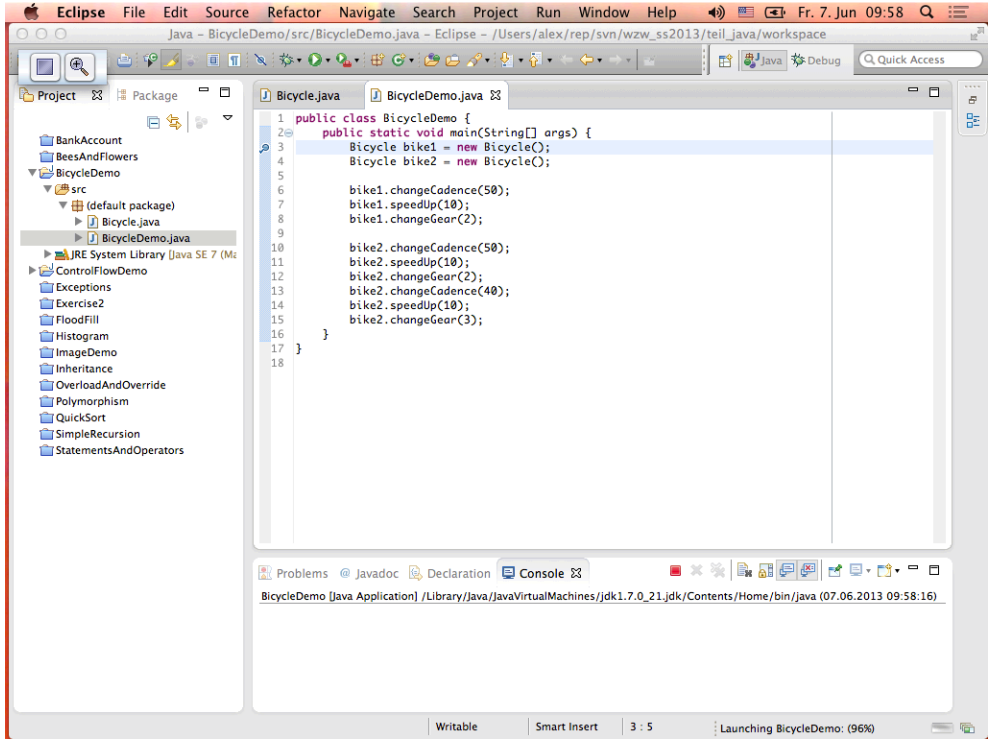
Project | Package

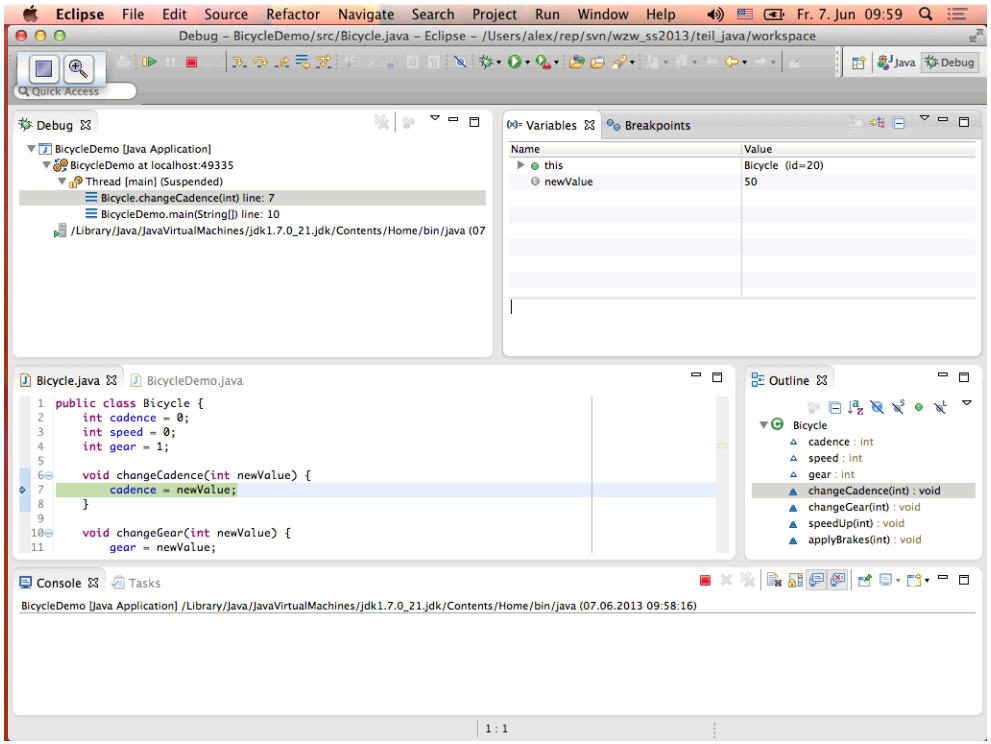
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
Exceptions
Exercise2
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators

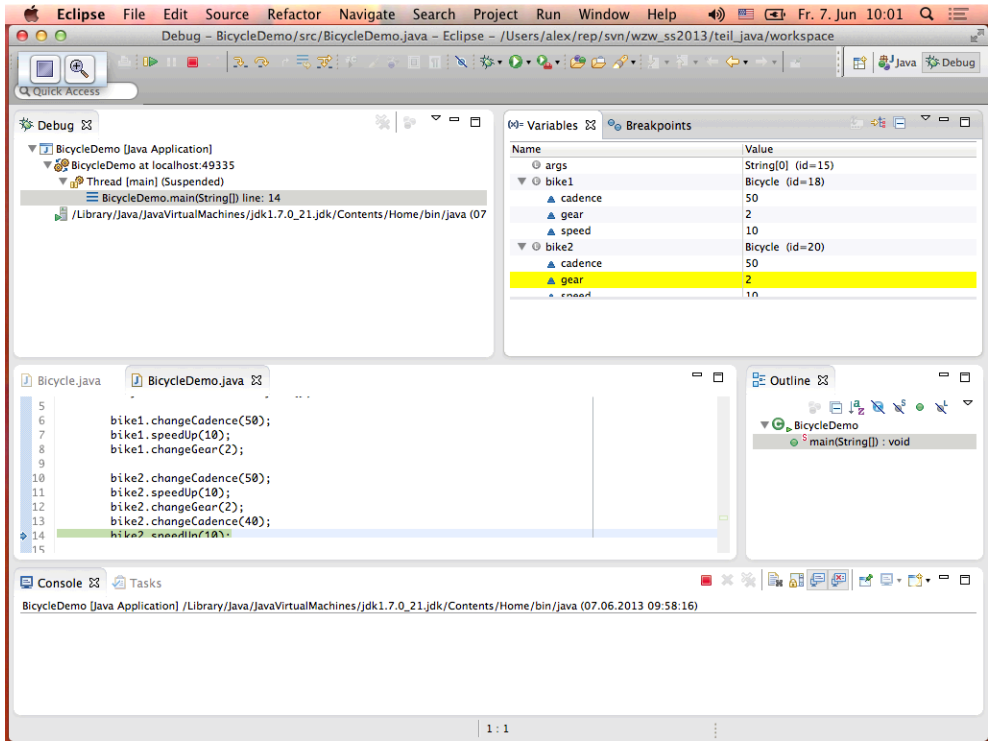
Problems | Javadoc | Declaration | Console

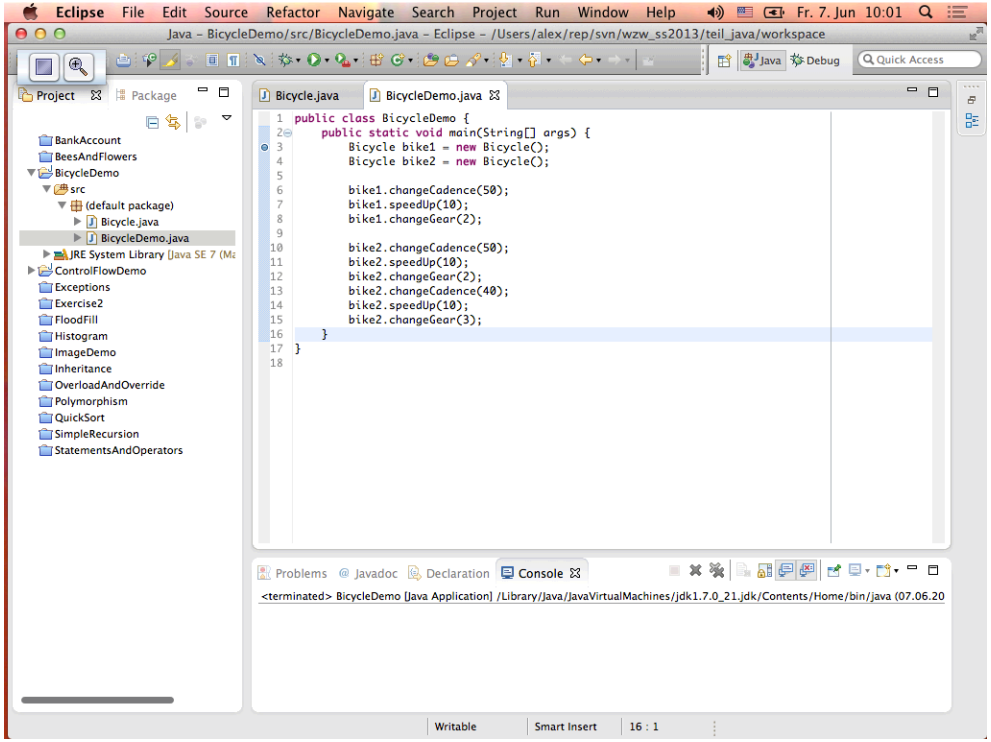
<terminated> ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.(
horst has the value 54758

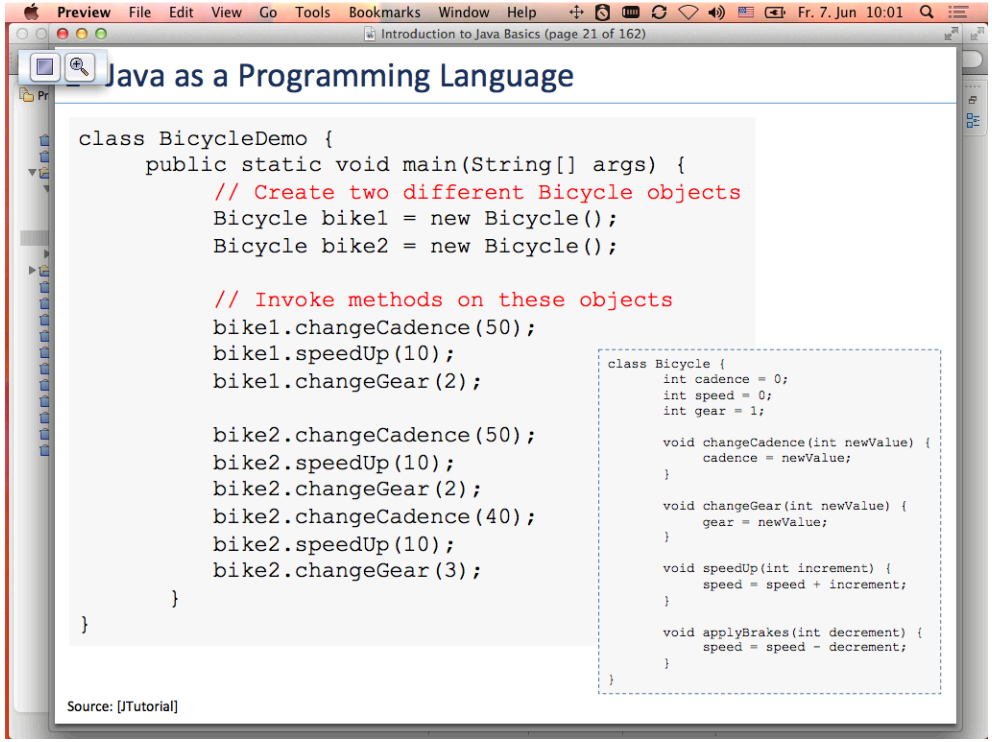

Eclipse File Edit Navigate Search Project Run Window Help
Fr. 7. Jun 09:57
Java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
src
(default package)
Bicycle.java
BicycleDemo.java
JRE System Library [Java SE 7 (Ma
ControlFlowDemo
Exceptions
Exercise2
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
Problems
Javadoc
Declaration
Console
<terminated> ControlFlowDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.(
horst has the value 54758
Bicycle.java – BicycleDemo/src

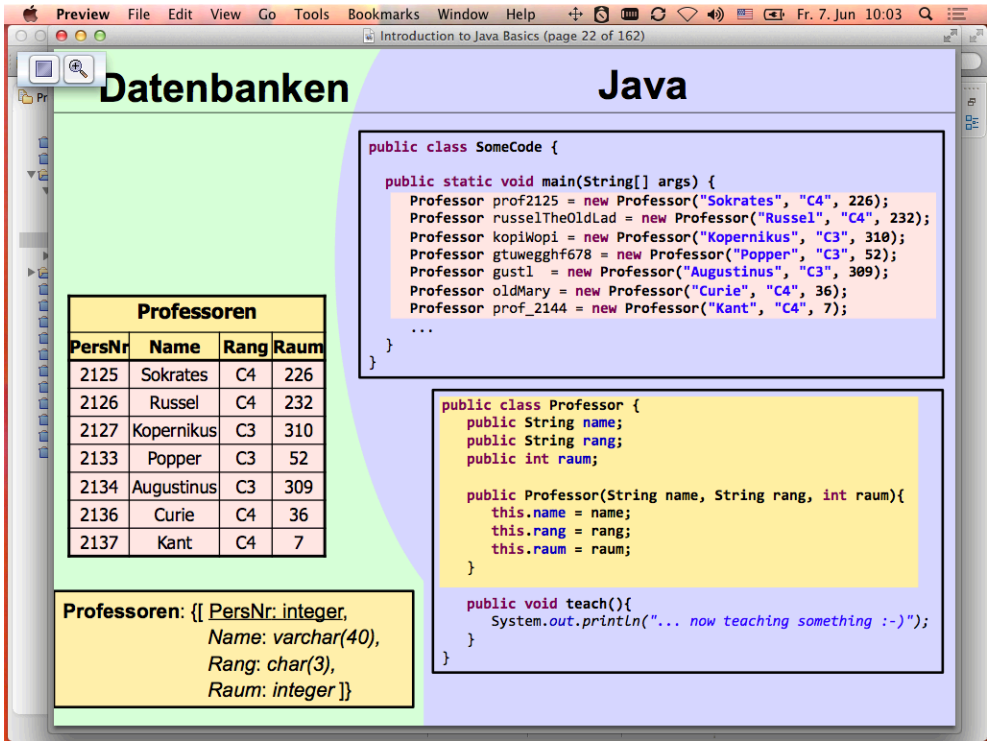

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:58
Java – BicycleDemo/src/BicycleDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Bicycle.java
BicycleDemo.java
public class BicycleDemo {
public static void main(String[] args) {
Bicycle bike1 = new Bicycle();
Bicycle bike2 = new Bicycle();
bike1.changeCadence(50);
bike1.speedUp(10);
bike1.changeGear(2);
bike2.changeCadence(50);
bike2.speedUp(10);
bike2.changeGear(2);
bike2.changeCadence(40);
bike2.speedUp(10);
bike2.changeGear(3);
}
}
BicycleDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:58:16)
Writable
Smart Insert
3 : 5
Launching BicycleDemo: (96%)

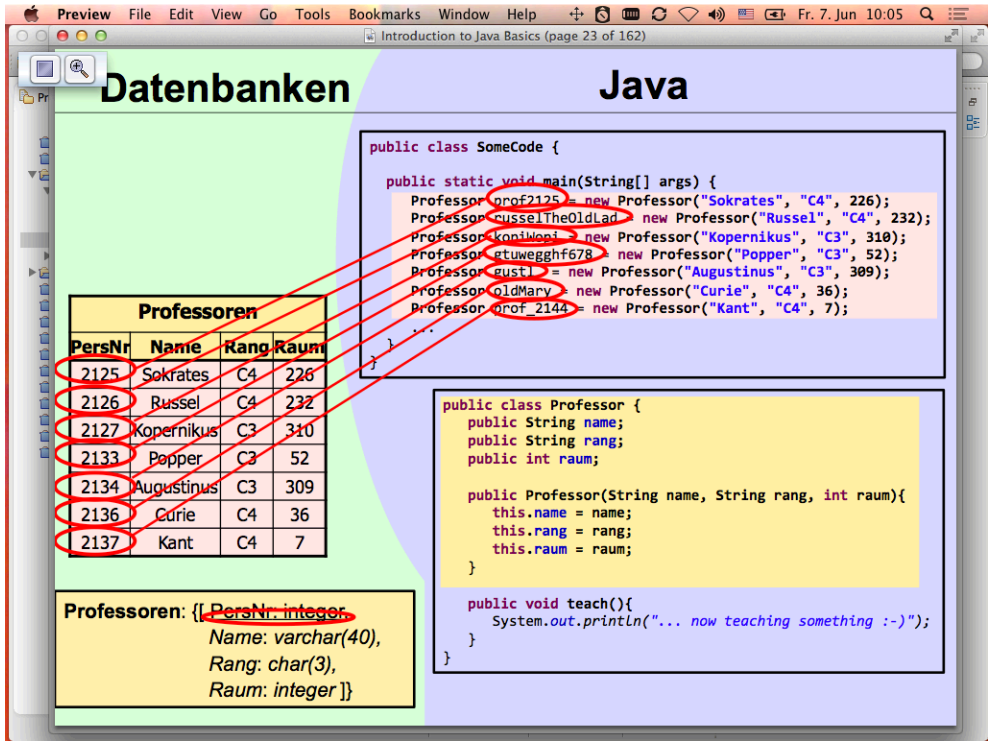

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 09:59
Debug – BicycleDemo/src/Bicycle.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
BicycleDemo [Java Application]
BicycleDemo at localhost:49335
Thread [main] (Suspended)
Bicycle.changeCadence(int) line: 7
BicycleDemo.main(String[]) line: 10
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07
Variables
Breakpoints
Name
Value
this
Bicycle (id=20)
newValue
50
Bicycle.java
BicycleDemo.java
public class Bicycle {
int cadence = 0;
int speed = 0;
int gear = 1;
void changeCadence(int newValue) {
cadence = newValue;
}
void changeGear(int newValue) {
gear = newValue;
Outline
Bicycle
cadence : int
speed : int
gear : int
changeCadence(int) : void
changeGear(int) : void
speedUp(int) : void
applyBrakes(int) : void
Console
Tasks
BicycleDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:58:16)
1 : 1


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fr. 7. Jun 10:01
Debug – BicycleDemo/src/BicycleDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Debug
BicycleDemo [Java Application]
BicycleDemo at localhost:49335
Thread [main] (Suspended)
BicycleDemo.main(String[]) line: 14
/Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07
Variables
Breakpoints
Name
Value
args
String[0] (id=15)
bike1
Bicycle (id=18)
cadence
50
gear
2
speed
10
bike2
Bicycle (id=20)
cadence
50
gear
2
Bicycle.java
BicycleDemo.java
bike1.changeCadence(50);
bike1.speedUp(10);
bike1.changeGear(2);
bike2.changeCadence(50);
bike2.speedUp(10);
bike2.changeGear(2);
bike2.changeCadence(40);
Outline
BicycleDemo
main(String[]) : void
Console
Tasks
BicycleDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.2013 09:58:16)
1 : 1


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Java – BicycleDemo/src/BicycleDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2013/teil_java/workspace
Project
Package
BankAccount
BeesAndFlowers
BicycleDemo
src
(default package)
Bicycle.java
BicycleDemo.java
JRE System Library [Java SE 7 (Ma
ControlFlowDemo
Exceptions
Exercise2
FloodFill
Histogram
ImageDemo
Inheritance
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
Bicycle.java
BicycleDemo.java
1 public class BicycleDemo {
2 public static void main(String[] args) {
3 Bicycle bike1 = new Bicycle();
4 Bicycle bike2 = new Bicycle();
5
6 bike1.changeCadence(50);
7 bike1.speedUp(10);
8 bike1.changeGear(2);
9
10 bike2.changeCadence(50);
11 bike2.speedUp(10);
12 bike2.changeGear(2);
13 bike2.changeCadence(40);
14 bike2.speedUp(10);
15 bike2.changeGear(3);
16 }
17 }
18
Problems Javadoc Declaration Console
<terminated> BicycleDemo [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_21.jdk/Contents/Home/bin/java (07.06.20
Writable
Smart Insert
16 : 1


Preview File Edit View Go Tools Bookmarks Window Help
Introduction to Java Basics (page 21 of 162)
Java as a Programming Language
class BicycleDemo {
public static void main(String[] args) {
// Create two different Bicycle objects
Bicycle bike1 = new Bicycle();
Bicycle bike2 = new Bicycle();
// Invoke methods on these objects
bike1.changeCadence(50);
bike1.speedUp(10);
bike1.changeGear(2);
bike2.changeCadence(50);
bike2.speedUp(10);
bike2.changeGear(2);
bike2.changeCadence(40);
bike2.speedUp(10);
bike2.changeGear(3);
}
}
class Bicycle {
int cadence = 0;
int speed = 0;
int gear = 1;
void changeCadence(int newValue) {
cadence = newValue;
}
void changeGear(int newValue) {
gear = newValue;
}
void speedUp(int increment) {
speed = speed + increment;
}
void applyBrakes(int decrement) {
speed = speed - decrement;
}
}
Source: [JTutorial]


Preview File Edit View Go Tools Bookmarks Window Help
Introduction to Java Basics (page 22 of 162)
Datenbanken
Java
Professoren
PersNr Name Rang Raum
2125 Sokrates C4 226
2126 Russel C4 232
2127 Kopernikus C3 310
2133 Popper C3 52
2134 Augustinus C3 309
2136 Curie C4 36
2137 Kant C4 7
Professoren: {[PersNr: integer,
Name: varchar(40),
Rang: char(3),
Raum: integer]}
public class SomeCode {
public static void main(String[] args) {
Professor prof2125 = new Professor("Sokrates", "C4", 226);
Professor russelTheOldLad = new Professor("Russel", "C4", 232);
Professor kopiWopi = new Professor("Kopernikus", "C3", 310);
Professor gtuwegghf678 = new Professor("Popper", "C3", 52);
Professor gustl = new Professor("Augustinus", "C3", 309);
Professor oldMary = new Professor("Curie", "C4", 36);
Professor prof_2144 = new Professor("Kant", "C4", 7);
...
}
}
public class Professor {
public String name;
public String rang;
public int raum;
public Professor(String name, String rang, int raum){
this.name = name;
this.rang = rang;
this.raum = raum;
}
public void teach(){
System.out.println("... now teaching something :-)");
}
}


Preview File Edit View Go Tools Bookmarks Window Help
Introduction to Java Basics (page 23 of 162)
Datenbanken
Java
Professoren
PersNr Name Rang Raum
2125 Sokrates C4 226
2126 Russel C4 232
2127 Kopernikus C3 310
2133 Popper C3 52
2134 Augustinus C3 309
2136 Curie C4 36
2137 Kant C4 7
Professoren: {[PersNr: integer,
Name: varchar(40),
Rang: char(3),
Raum: integer]}
public class SomeCode {
public static void main(String[] args) {
Professor prof2125 = new Professor("Sokrates", "C4", 226);
Professor russelTheOldLad = new Professor("Russel", "C4", 232);
Professor kopiWopi = new Professor("Kopernikus", "C3", 310);
Professor gtuwegghf678 = new Professor("Popper", "C3", 52);
Professor gustl = new Professor("Augustinus", "C3", 309);
Professor oldMary = new Professor("Curie", "C4", 36);
Professor prof_2144 = new Professor("Kant", "C4", 7);
...
}
}
public class Professor {
public String name;
public String rang;
public int raum;
public Professor(String name, String rang, int raum){
this.name = name;
this.rang = rang;
this.raum = raum;
}
public void teach(){
System.out.println("... now teaching something :-)");
}
}

Datenbanken

Professoren

PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Professoren: {[PersNr: integer,
Name: varchar(40),
Rang: char(3),
Raum: integer]}

Java

```
public class SomeCode {

  public static void main(String[] args) {
     Professor prof2125 = new Professor("Sokrates", "C4", 226);
     Professor russelTheOldLad = new Professor("Russel", "C4", 232);
     Professor kopiWopi = new Professor("Kopernikus", "C3", 310);
     Professor gtuwegghf678 = new Professor("Popper", "C3", 52);
     Professor gustl  = new Professor("Augustinus", "C3", 309);
     Professor oldMary = new Professor("Curie", "C4", 36);
     Professor prof_2144 = new Professor("Kant", "C4", 7);
     ...
  }
}
```

```
public class Professor {
   public String name;
   public String rang;
   public int raum;

   public Professor(String name, String rang, int raum){
      this.name = name;
      this.rang = rang;
      this.raum = raum;
   }

   public void teach(){
      System.out.println("... now teaching something :-)");
   }
}
```


Datenbanken

Java

???

Professoren

PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Professoren: {[PersNr: integer,
Name: varchar(40),
Rang: char(3),
Raum: integer]}

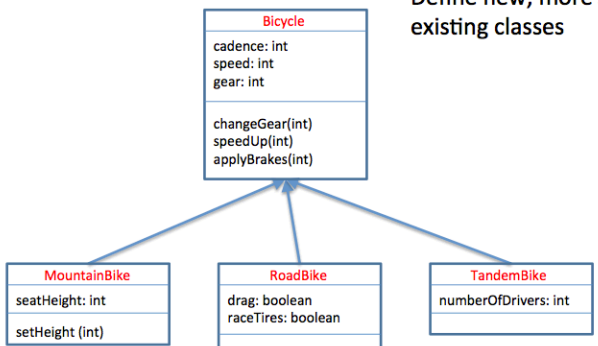
```
public class SomeCode {

  public static void main(String[] args) {
     Professor prof2125 = new Professor("Sokrates", "C4", 226);
     Professor russelTheOldLad = new Professor("Russel", "C4", 232);
     Professor kopiWopi = new Professor("Kopernikus", "C3", 310);
     Professor gtuwegghf678 = new Professor("Popper", "C3", 52);
     Professor gustl  = new Professor("Augustinus", "C3", 309);
     Professor oldMary = new Professor("Curie", "C4", 36);
     Professor prof_2144 = new Professor("Kant", "C4", 7);
     ...
  }
}
```

```
public class Professor {
   public String name;
   public String rang;
   public int raum;

   public Professor(String name, String rang, int raum){
      this.name = name;
      this.rang = rang;
      this.raum = raum;
   }

   public void teach(){
      System.out.println("... now teaching something :-)");
   }
}
```


Java as a Programming Language

Inheritance

Inheritance:
Define new, more *specialized* classes from existing classes




```
class RoadBike extends Bicycle {
    // additional fields and methods
    // that define a road bike
    // go here
}
```


Java as a Programming Language

Interfaces

Interface:
Specify in an abstract way what a class implementing that interface should exhibit as behaviours (create blueprint for blueprints)

```
interface IBicycle {
       void changeCadence(int newValue);

       void changeGear(int newValue);

       void speedUp(int increment);

       void applyBrakes(int decrement);
}
```

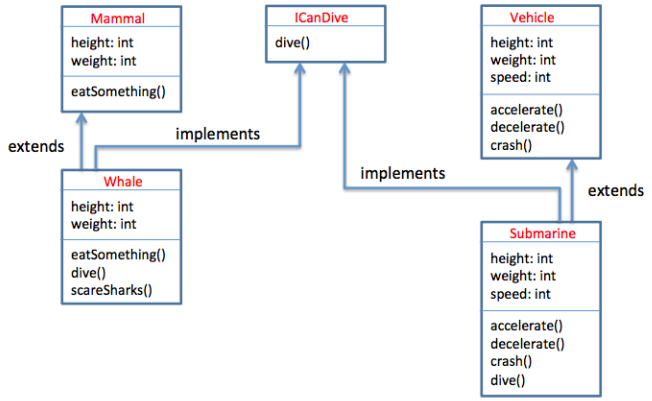
```
class Bicycle implements IBicycle {
   // remainder of this class implemented as before
   // except that above methods must be public
}
```

see: [JTutorial]

Java as a Programming Language

Interfaces

Example:



2 Language Basics

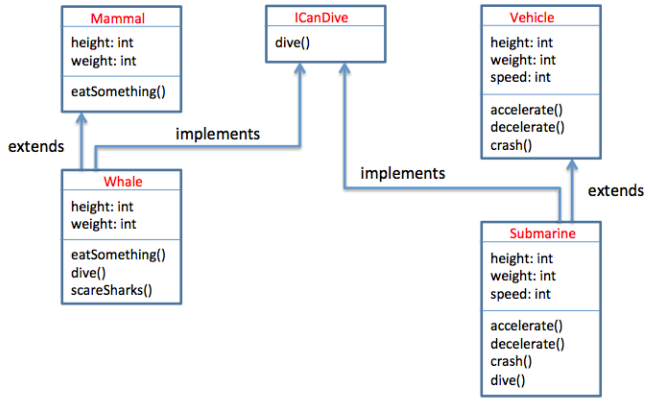
Deepening readings:

http://java.sun.com/docs/books/tutorial/java/nutsandbolts/variables.html
http://java.sun.com/docs/books/tutorial/java/nutsandbolts/datatypes.html
http://java.sun.com/docs/books/tutorial/java/nutsandbolts/arrays.html
http://java.sun.com/docs/books/tutorial/java/nutsandbolts/opsummary.html
http://java.sun.com/docs/books/tutorial/java/nutsandbolts/expressions.html
http://java.sun.com/docs/books/tutorial/java/nutsandbolts/if.html
http://java.sun.com/docs/books/tutorial/java/nutsandbolts/while.html
http://java.sun.com/docs/books/tutorial/java/nutsandbolts/for.html
http://java.sun.com/docs/books/tutorial/java/nutsandbolts/branch.html

Java as a Programming Language

Interfaces

Example:



Language Basics – Variables

Variables

- Variables have a type
  - Primitive type
  - Reference type

	Definition	Declaration	Instantiation	Manipulation	Equality
Primitive	predefined	int a;	a = 117;	a = b + 42;	a == b;
Reference	class Student { // Fields and // methods ... }	Student heiner;	heiner = new Student();	heiner.yawn();	heiner.equals(sabine);

Language Basics – Variables

- Variables have a type
  - Primitive type

```
int horst = 101;
long heiner;
heiner = 235638465837465845;
```

  - Reference type

```
Bicycle bike1 = new Bicycle();
bike1.gear = 3;

MountainBike bike2 =
    new MountainBike();
```

memory (simplified model)

cell nr.	cell name	cell content	
1123	horst	101	data
1124	heiner	235638465	
1125		837465845	
...	...	...	
1150	bike1.cadence	0	
1151	bike1.speed	0	
1152	bike1.gear	3	
...	...	...	
1330	bike2.cadence	0	
1331	bike2.speed	0	
1332	bike2.gear	1	
1333	bike2.seatHeight	15	
...	...	...	instructions
4027		void changeCadence(int newValue) {	
4028		cadence = newValue;	
4029		}	
...	...	...	
4035		int horst = 101;	

Language Basics – Variables

- Variables have a type
  - Primitive type

int horst = 101;
long heiner;
heiner = 235638465837465845;

  - Reference type

Bicycle bike1 = new Bicycle();
bike1.gear = 3;

MountainBike bike2 =
    new MountainBike();

memory (simplified model)

cell nr.	cell name	cell content	
1123	horst	101	data
1124	heiner	235638465	
1125		837465845	
...	...	...	
1150	bike1.cadence	0	
1151	bike1.speed	0	
1152	bike1.gear	3	
...	...	...	
1330	bike2.cadence	0	
1331	bike2.speed	0	
1332	bike2.gear	1	
1333	bike2.seatHeight	15	
...	...	...	instructions
4027		void changeCadence(int newValue) {	
4028		cadence = newValue;	
4029		}	
...	...	...	
4035		int horst = 101;	

Language Basics – Variables

Primitive Types

- Primitive types (numeric):

byte	short	int	long	float	double
8 bit	16 bit	32 bit	64 bit	32 bit	64 bit

- Examples:

```
byte flags = 63;
short bbb  = 10133;
int heiner = 234103234;
long dong  = -83628735682345;
float fff  = 5464.00345;
float ggg  = -345545.34534E-12f;
double sss = 3245343455.555E67;
```

= -345545.34534 * 10^{-12}
= 3245343455.555 * 10^{67}

Language Basics – Variables

- Primitive types (numeric, boolean, character):

$\in \mathbb{Z}$ (byte, short, int, long) $\in \mathbb{R}$ (float, double) Unicode (UTF-16) (char)

byte	short	int	long	float	double	boolean	char
8 bit	16 bit	32 bit	64 bit	32 bit	64 bit	1 bit	16 bit
$[-2^7, 2^7-1]$ = [-128,127]	$[-2^{15}, 2^{15}-1]$ = [-32768,32767]	$[-2^{31}, 2^{31}-1]$ = [-2147483648, 2147483647]	$[-2^{63}, 2^{63}-1]$ = [-9223372036854775808, 9223372036854775807]	[+/- ~ $1.4*10^{-45}$, +/- ~ $3.4*10^{38}$]	[+/- ~ $4.9*10^{-324}$, +/- ~ $1.8\ 10^{308}$]	{ true, false }	{ ... !, „, §, $, %, &, ..., a, b, c, ..., ∭,...,∰,...,≎, ガ,キ,ギ,...., ㋗,㋘,㋙,..., 樵,樶,蕿,... 액,앸, 앤...,ش,ش, .. .,→,↓,■, ...}

Language Basics – Variables

- More examples:

```
byte flags = 63;
short bbb  = 10133;
int heiner = 234103234;
long dng   = -83628735682345;
float fff  = 5464.00345f;
float ggg  = -345545.34534E-12f;
double sss = 3245343455.555E67d;

char ccc   = 'm';
char ccc2  = '\n';

boolean isCool = true;
```

byte typically used for bit-patterns

= -345545.34534 * 10^{-12} (float)

= 3245343455.555 * 10^{67} (double)

\n means "new line"

Language Basics – Variables

Reference Type Variables

- Reference type variables "point" to an object of the reference type

```
bike1 = new Bicycle();
bike2 = new Bicycle();

boolean c;
c = bike1.equals(bike2);
    // c == true
c = (bike1 == bike2);
    // c == false
```

memory (simplified model)

cell nr	cell name	cell content
...	...	...
1149	bike1	<1150>
1150	bike1.cadence	0
1151	bike1.speed	0
1152	bike1.gear	1
...	...	...
1327	bike2	<1405>
...	...	...
1405	bike2.cadence	0
1406	bike2.speed	0
1407	bike2.gear	1
...	...	...

data