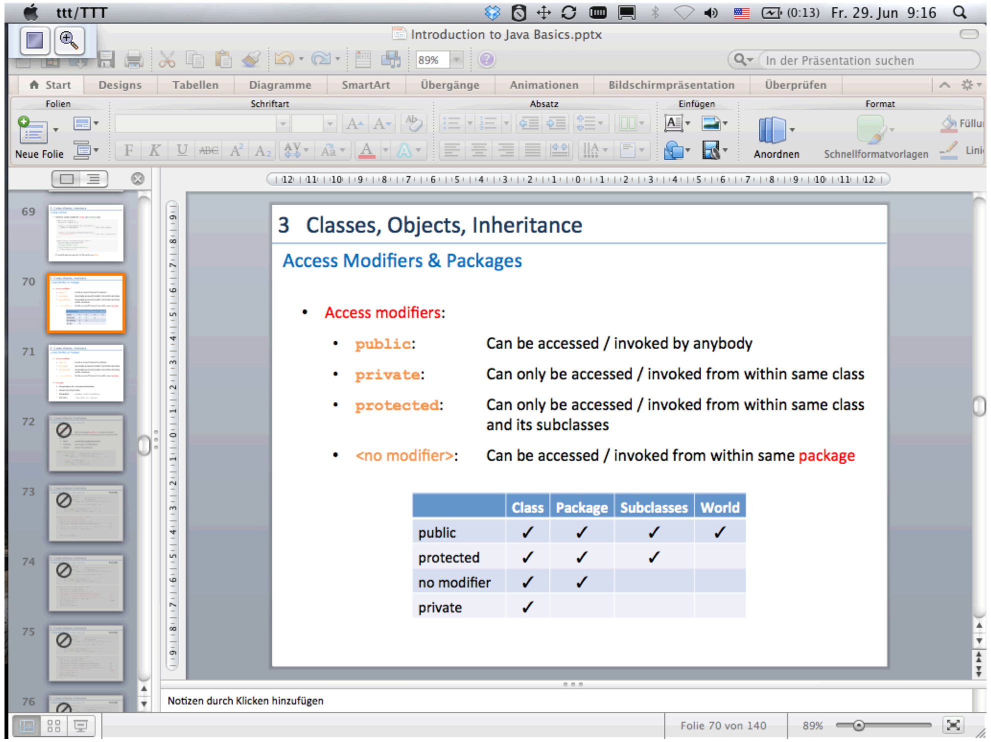
Script generated by TTT

Title: Lehmann: Uebung_Einf_HF (29.06.2012)

Date: Fri Jun 29 09:16:28 CEST 2012

Duration: 91:43 min

Pages: 95



3 Classes, Objects, Inheritance

Access Modifiers & Packages

- Access modifiers:
  - public: Can be accessed / invoked by anybody
  - private: Can only be accessed / invoked from within same class
  - protected: Can only be accessed / invoked from within same class and its subclasses
  - <no modifier>: Can be accessed / invoked from within same package

	Class	Package	Subclasses	World
public	✓	✓	✓	✓
protected	✓	✓	✓	
no modifier	✓	✓		
private	✓			

Classes, Objects, Inheritance

Access Modifiers & Packages

- Access modifiers:
  - public: Can be accessed / invoked by anybody
  - private: Can only be accessed / invoked from within same class
  - protected: Can only be accessed / invoked from within same class and its subclasses
  - <no modifier>: Can be accessed / invoked from within same package

	Class	Package	Subclasses	World
public	✓	✓	✓	✓
protected	✓	✓	✓	
no modifier	✓	✓		
private	✓			

Classes, Objects, Inheritance

Access Modifiers & Packages

- Access modifiers:
  - public: Can be accessed / invoked by anybody
  - private: Can only be accessed / invoked from within same class
  - protected: Can only be accessed / invoked from within same class and its subclasses
  - <no modifier>: Can be accessed / invoked from within same package
- Packages:
  - Encapsulate a set of classes and interfaces
  - Hierarchical organization
  - Declaration: `package myfirstpackage;`
  - Examples: `java.math, de.tum.wzw`

Classes, Objects, Inheritance

- Designated class-method `main` with fixed signature

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

  is called once at program start

- "Computer has to know where to start":

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

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

see: [JTutorial]

Classes, Objects, Inheritance

Overloading

- Overloading: Methods with same name but different parameters (types)

```
class OverloadingDemoClass {
      public int doSomething() {
           return 1 + 1;
      }

      public int doSomething(int param) {
           return param + 2;
      }
}
```

```
public static void main(String[] args) {
        OverloadingDemoClass odc = new OverloadingDemoClass();
        int result1 = odc.doSomething();
        int result2 = odc.doSomething(33);
}
```

- Method signature comprised of name and parameter types

Classes, Objects, Inheritance

Overriding, Hiding

- Overriding methods
  - Why?
    Let subclasses provide a more specialized version of an instance-method
  - How?
    Subclass defines an instance-method with same signature (name plus number and types of parameters) as defined by super-class

Classes, Objects, Inheritance

Overriding, Hiding

- Overriding methods
  - Let subclasses provide a more specialized version of an instance-method
  - Subclass defines an instance-method with same signature (name plus *number* and *types* of parameters) as defined by superclass

```
class Bicycle {
    public void speedUp(int increment) {
        speed = speed + increment;
        System.out.println("superclass instance-method");
    }
}
```

```
class MountainBike extends Bicycle {
    public void speedUp(int increment) {
        speed = speed + 2 * increment;
        System.out.println("subclass instance-method");
    }
}
```

```
MountainBike mountainBike = new MountainBike();
mountainBike.speedUp(10);
```

⇨ output will be: `subclass instance-method`

Classes, Objects, Inheritance

Overriding, Hiding

- Hiding methods
  - Why?
    Let subclasses provide a more specialized version of a class-method
  - How?
    Subclass defines a class-method with same signature (name plus number and types of parameters) as defined by superclass

Classes, Objects, Inheritance

Overriding, Hiding

- Hiding methods
  - Let subclasses provide a more specialized version of a class-method
  - Subclass defines a class-method with same signature (name plus number and types of parameters) as defined by superclass

```
class Bicycle {
    public static void myClassMethod(int someInt) {
        System.out.println("superclass class-method");
    }
    public void myInstanceMethod(int someInt) {
        System.out.println("superclass instance-method");
    }
}
```

```
class MountainBike extends Bicycle {
    public static void myClassMethod(int someInt) {
        System.out.println("subclass class-method");
    }
    public void myInstanceMethod(int someInt) {
        System.out.println("subclass instance-method");
    }
}
```

```
Bicycle.myClassMethod(10);          // "superclass class-method"
MountainBike.myClassMethod(10);     // "subclass class-method"
```

Classes, Objects, Inheritance

Polymorphism

- Polymorphism: subclass objects may be assigned to superclass variables

```
MountainBike mountainBike = new MountainBike();
Bicycle bicycle = mountainBike;
```

→ Essential feature of object oriented software

- Only methods and fields defined by the the superclass "portion" of the object may be accessed; and the overridden ("right") methods are called

```
Bicycle bike = new MountainBike();
bicycle.gear = 3;           // Ok, gear defined in class Bicycle

bicycle.speedUp(10);        // Overridden method in subclass MountainBike is used

bicycle.seatHeight = 20;    // ERROR! seatHeight is not a field in class Bicycle
```

Classes, Objects, Inheritance

Overriding, Hiding

- Hiding methods
  - Let subclasses provide a more specialized version of a class-method
  - Subclass defines a class-method with same signature (name plus number and types of parameters) as defined by superclass

```
class Bicycle {
    public static void myClassMethod(int someInt) {
        System.out.println("superclass class-method");
    }
    public void myInstanceMethod(int someInt) {
        System.out.println("superclass instance-method");
    }
}
```

```
class MountainBike extends Bicycle {
    public static void myClassMethod(int someInt) {
        System.out.println("subclass class-method");
    }
    public void myInstanceMethod(int someInt) {
        System.out.println("subclass instance-method");
    }
}
```

```
Bicycle.myClassMethod(10);          // "superclass class-method"
MountainBike.myClassMethod(10);     // "subclass class-method"
```

Classes, Objects, Inheritance

Polymorphism

- Polymorphism: subclass objects may be assigned to superclass variables

```
MountainBike mountainBike = new MountainBike();
Bicycle bicycle = mountainBike;
```

→ Essential feature of object oriented software

- Only methods and fields defined by the the superclass "portion" of the object may be accessed; and the overridden ("right") methods are called

```
Bicycle bike = new MountainBike();
bicycle.gear = 3;           // Ok, gear defined in class Bicycle

bicycle.speedUp(10);        // Overridden method in subclass MountainBike is used

bicycle.seatHeight = 20;    // ERROR! seatHeight is not a field in class Bicycle
```

Classes, Objects, Inheritance

Polymorphism

- Polymorphism: subclass objects may be assigned to superclass variables

```
MountainBike mountainBike = new MountainBike();
Bicycle bicycle = mountainBike;
```

→ Essential feature of object oriented software

- Only methods and fields defined by the the superclass "portion" of the object may be accessed; and the overridden ("right") methods are called

```
Bicycle bike = new MountainBike();
bicycle.gear = 3;           // Ok, gear defined in class Bicycle

bicycle.speedUp(10);        // Overridden method in subclass MountainBike is used

bicycle.seatHeight = 20;    // ERROR! seatHeight is not a field in class Bicycle
```

Classes, Objects, Inheritance

Polymorphism

- Polymorphism: subclass objects may be assigned to superclass variables

```
MountainBike mountainBike = new MountainBike();
Bicycle bicycle = mountainBike;
```

→ Essential feature of object oriented software

- Only methods and fields defined by the the superclass "portion" of the object may be accessed; and the overridden ("right") methods are called

```
Bicycle bike = new MountainBike();
bicycle.gear = 3;           // Ok, gear defined in class Bicycle

bicycle.speedUp(10);        // Overridden method in subclass MountainBike is used

bicycle.seatHeight = 20;    // ERROR! seatHeight is not a field in class Bicycle
```

Classes, Objects, Inheritance

- **Purpose of polymorphism:**
  General superclass state and behaviour may be used on all subclass objects
  → Good software design

- **Somewhat similar:**
  Interfaces provide a blueprint of blueprints, and may be used as *type* in variable declarations.

  Different classes may implement the same interface.
  → The methods which are guaranteed by the interface may be called on objects of all corresponding classes

Classes, Objects, Inheritance

Example with Interface

```
interface SubOrderApocrita {
      public void sting();
}
```

```
class LittleBee implements SubOrderApocrita {
      public void sting() {
            System.out.println("*pieks*");
      }
}
```

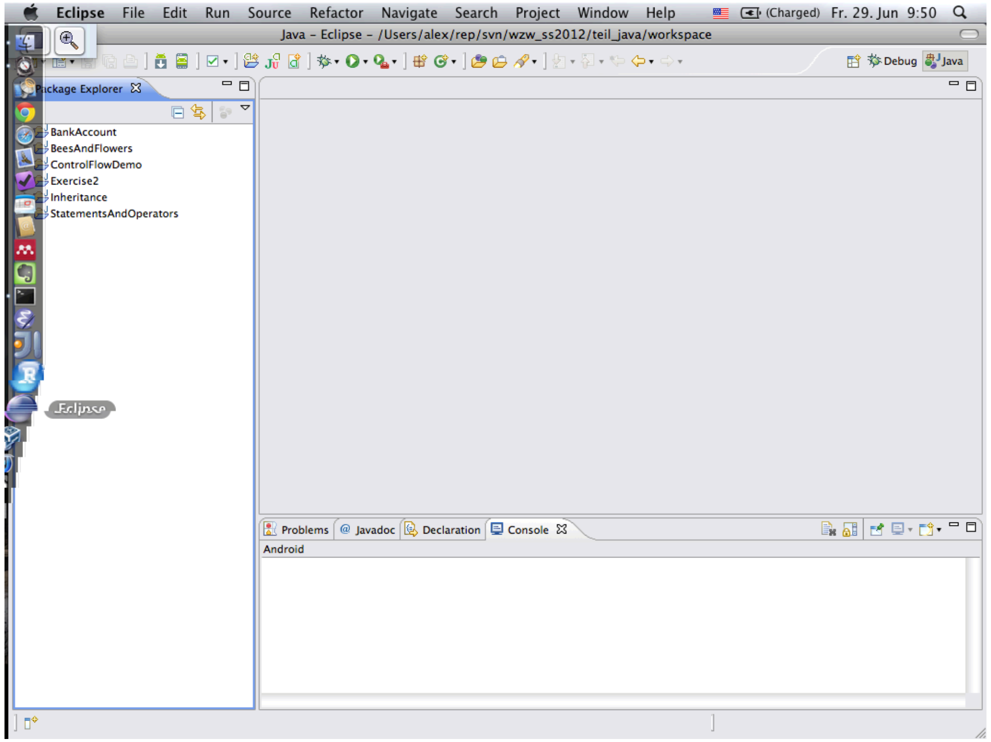
```
class AngryHornet implements SubOrderApocrita {
      public void sting() {
            System.out.println("*MEGAPIEKS*");
      }
}
```

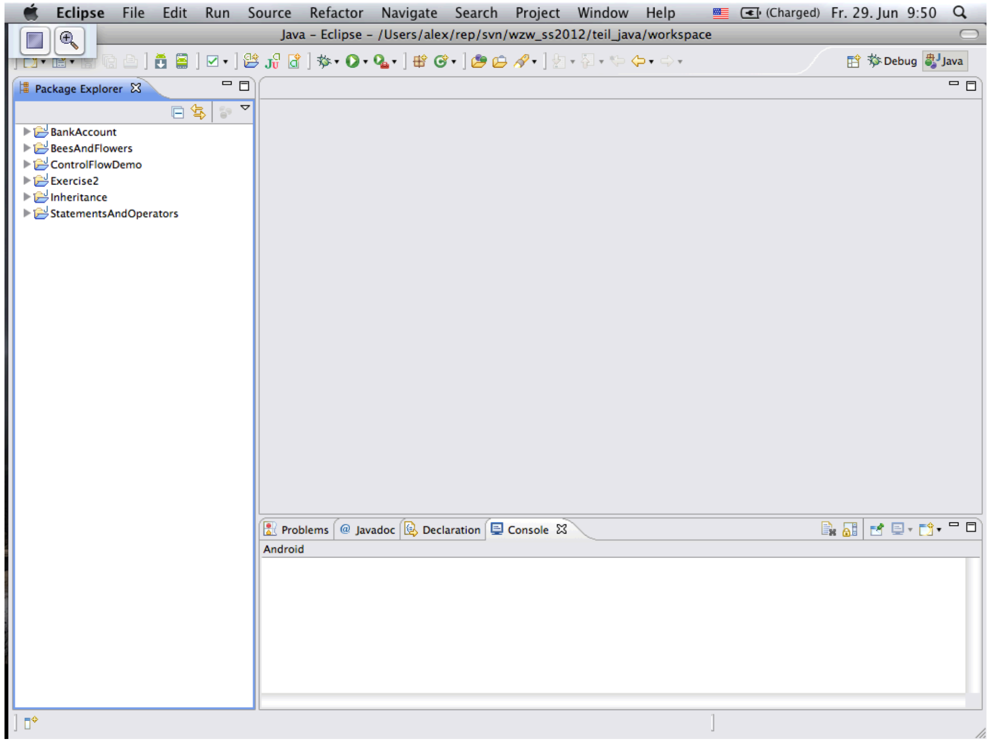
```
LittleBee maya = new LittleBee();
AngryHornet horst = new AngryHornet();
SuborderApocrita someStinger;
someStinger = maya;
someStinger.sting();          // *pieks*
someStinger = horst;
someStinger.sting();          // *MEGAPIEKS*
```

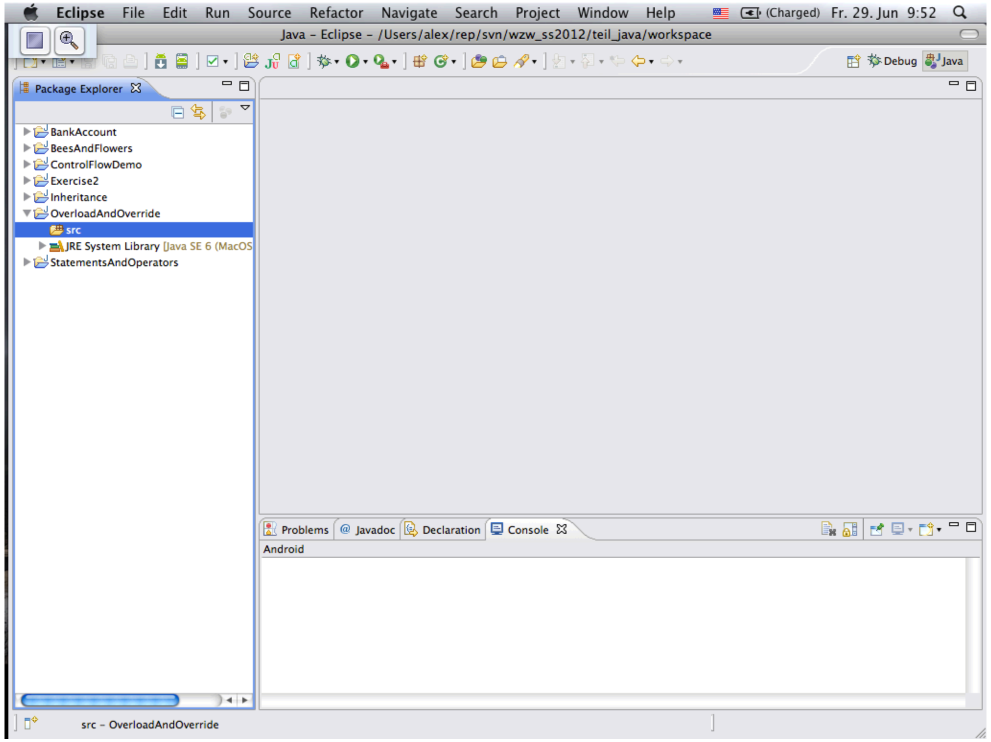
4 Recursion

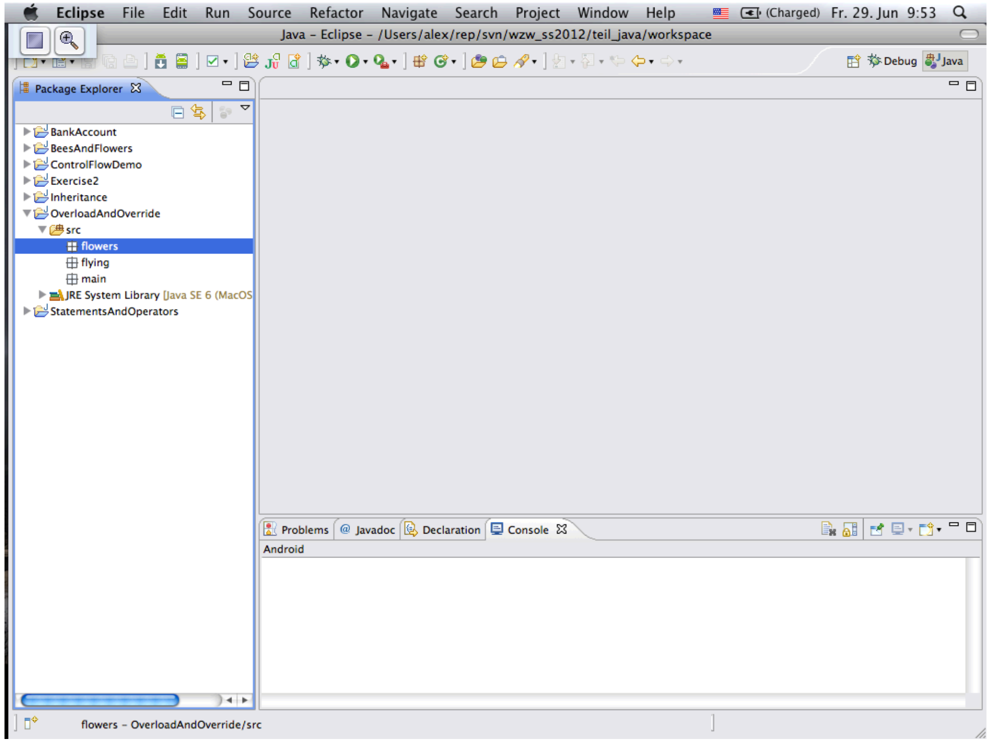
Deepening readings:

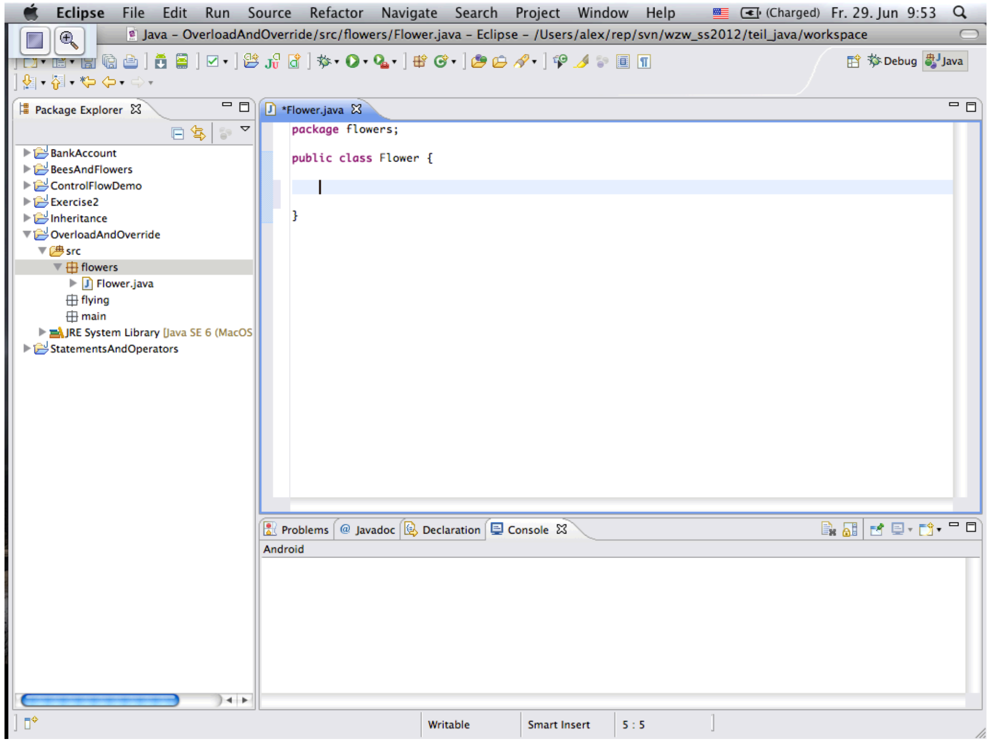
http://en.wikipedia.org/wiki/Recursion
http://en.wikipedia.org/wiki/Factorial
http://en.wikipedia.org/wiki/Tower_of_Hanoi


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 9:50
Java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Package Explorer
BankAccount
BeesAndFlowers
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
Eclipse
Problems
Javadoc
Declaration
Console
Android

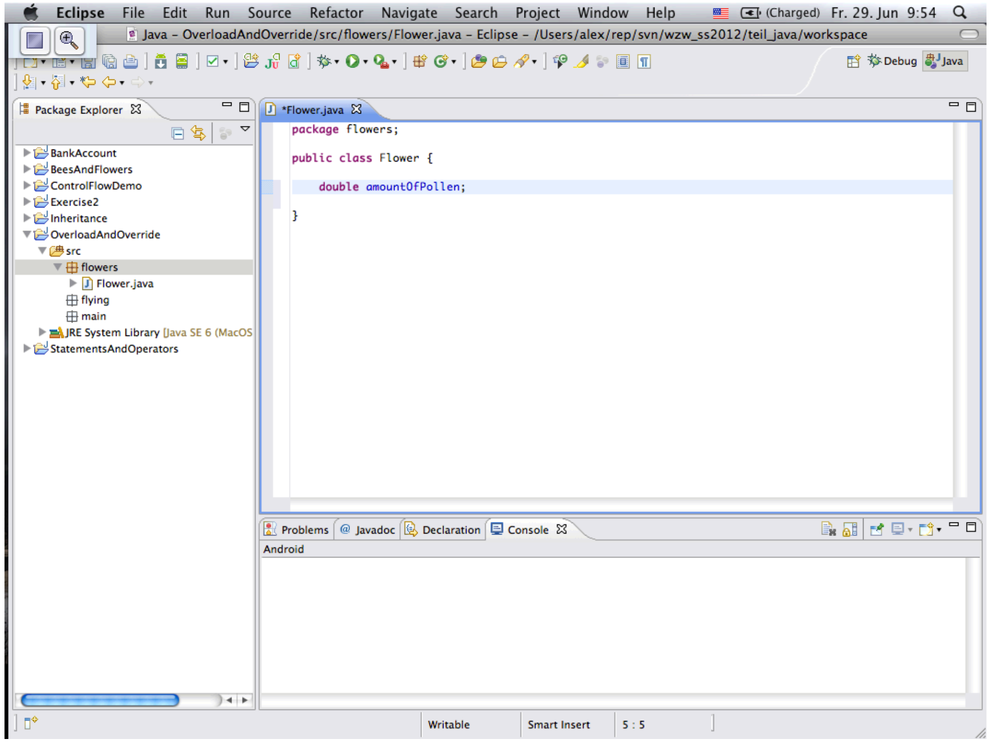

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 9:50
Java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Package Explorer
BankAccount
BeesAndFlowers
ControlFlowDemo
Exercise2
Inheritance
StatementsAndOperators
Problems
Javadoc
Declaration
Console
Android


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 9:52
Java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Package Explorer
BankAccount
BeesAndFlowers
ControlFlowDemo
Exercise2
Inheritance
OverloadAndOverride
src
JRE System Library [Java SE 6 (MacOS
StatementsAndOperators
Problems
Javadoc
Declaration
Console
Android
src – OverloadAndOverride

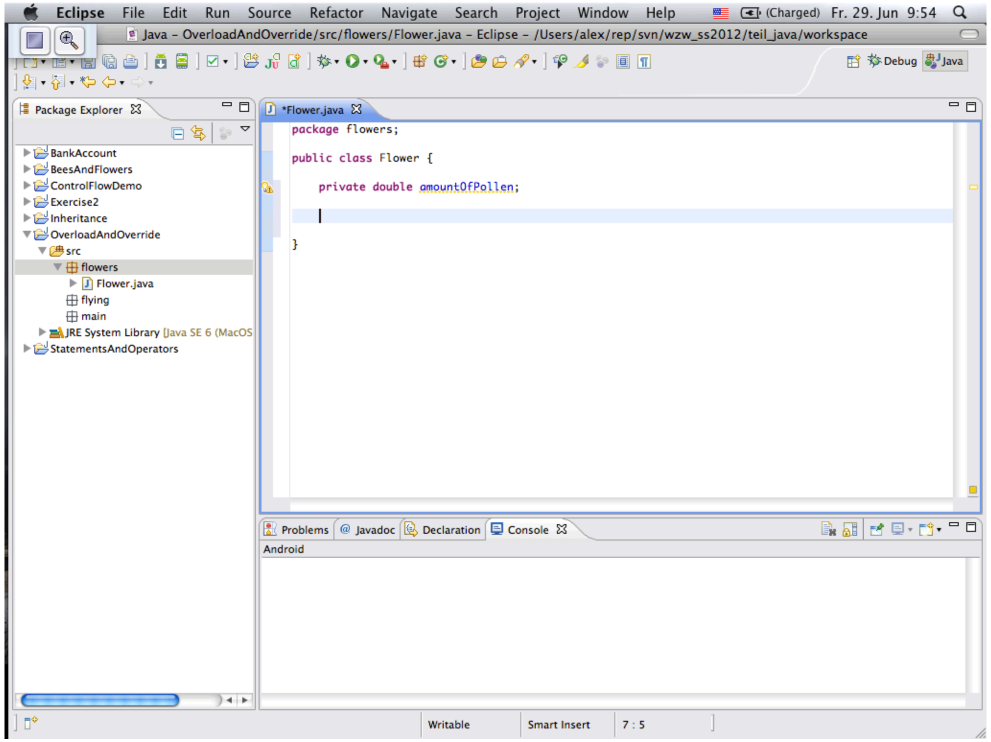

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 9:53
Java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Package Explorer
BankAccount
BeesAndFlowers
ControlFlowDemo
Exercise2
Inheritance
OverloadAndOverride
src
flowers
flying
main
JRE System Library [Java SE 6 (MacOS
StatementsAndOperators
Problems
Javadoc
Declaration
Console
Android
flowers – OverloadAndOverride/src


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 9:53
Java – OverloadAndOverride/src/flowers/Flower.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Package Explorer
*Flower.java
BankAccount
BeesAndFlowers
ControlFlowDemo
Exercise2
Inheritance
OverloadAndOverride
src
flowers
Flower.java
flying
main
JRE System Library [Java SE 6 (MacOS
StatementsAndOperators
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
5 : 5

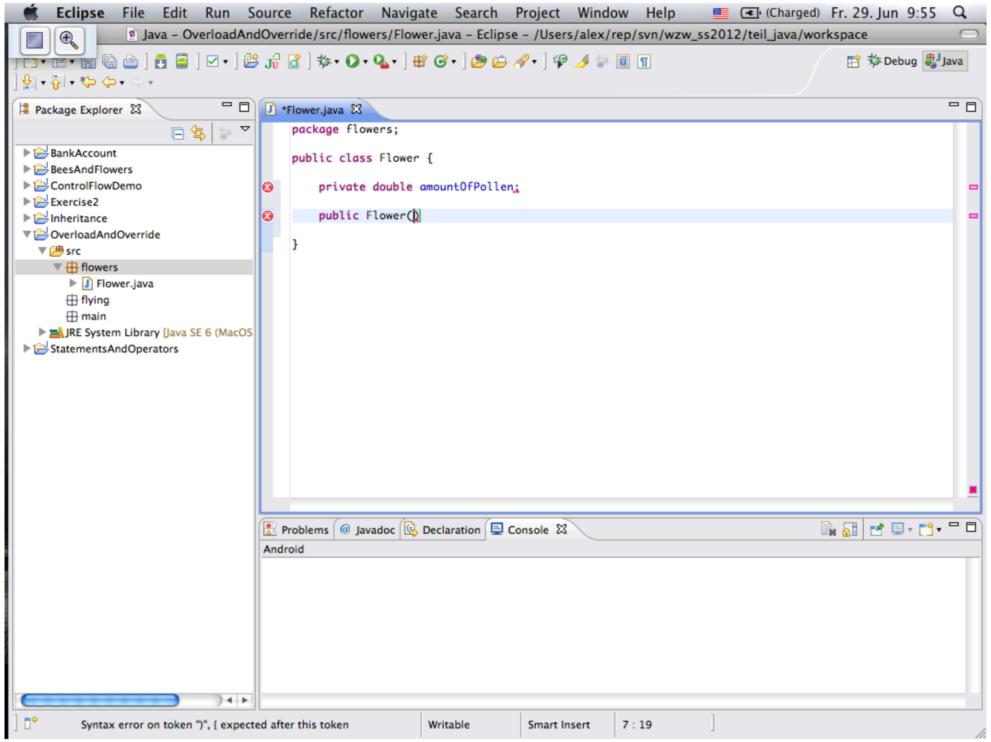
```java
package flowers;

public class Flower {

}
```


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 9:54
Java – OverloadAndOverride/src/flowers/Flower.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Package Explorer
*Flower.java
BankAccount
BeesAndFlowers
ControlFlowDemo
Exercise2
Inheritance
OverloadAndOverride
src
flowers
Flower.java
flying
main
JRE System Library [Java SE 6 (MacOS
StatementsAndOperators
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
5 : 5

```java
package flowers;

public class Flower {

    double amountOfPollen;
}
```

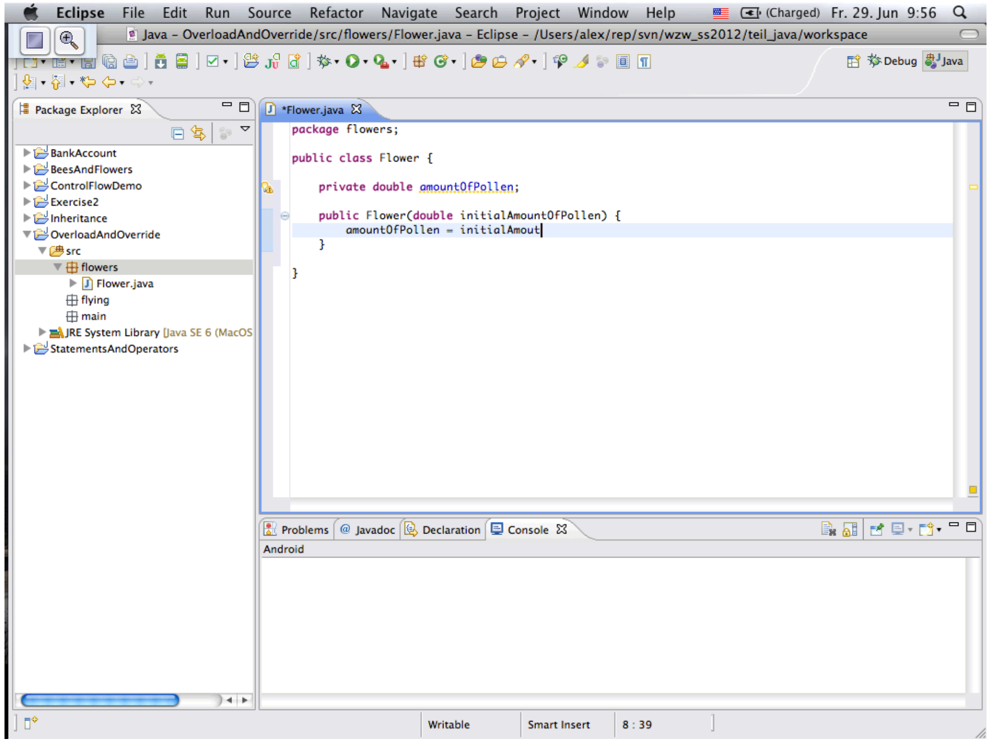

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 9:54
Java – OverloadAndOverride/src/flowers/Flower.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
BankAccount
BeesAndFlowers
ControlFlowDemo
Exercise2
Inheritance
OverloadAndOverride
src
flowers
Flower.java
flying
main
JRE System Library [Java SE 6 (MacOS
StatementsAndOperators
*Flower.java
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
7 : 5

```
package flowers;

public class Flower {

    private double amountOfPollen;

    

}
```

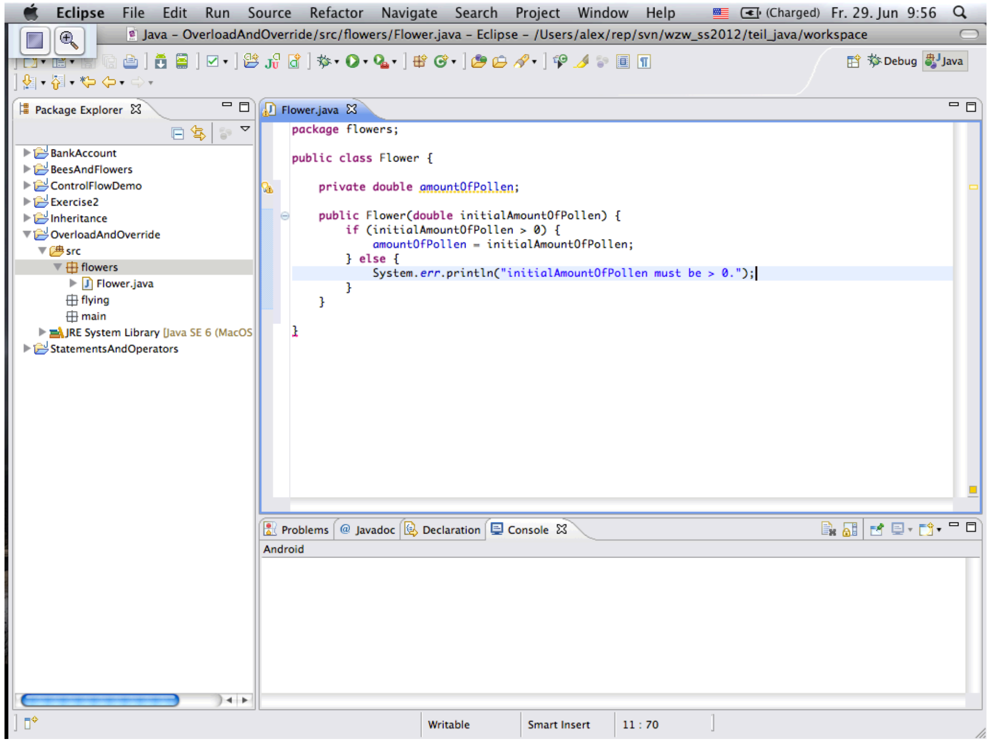

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 9:55
Java – OverloadAndOverride/src/flowers/Flower.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
*Flower.java
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
7 : 19

```
package flowers;

public class Flower {

    private double amountOfPollen;

    public Flower()

}
```

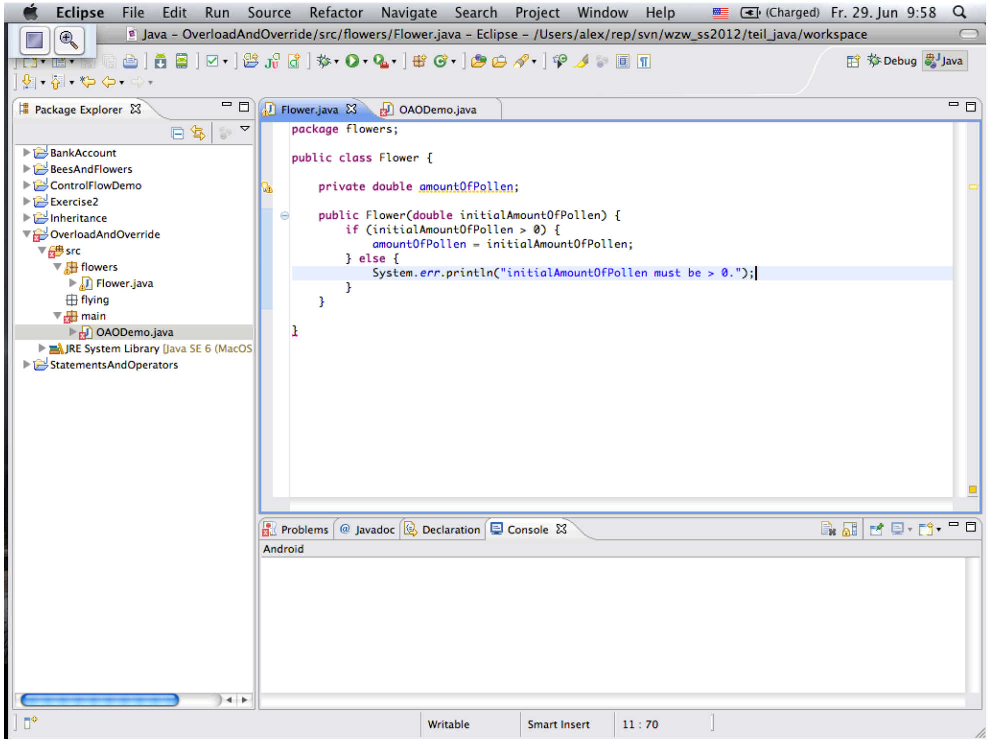
Syntax error on token ")", { expected after this token


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 9:56
Java – OverloadAndOverride/src/flowers/Flower.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
*Flower.java
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
8 : 39

```
package flowers;

public class Flower {

    private double amountOfPollen;

    public Flower(double initialAmountOfPollen) {
        amountOfPollen = initialAmout
    }

}
```

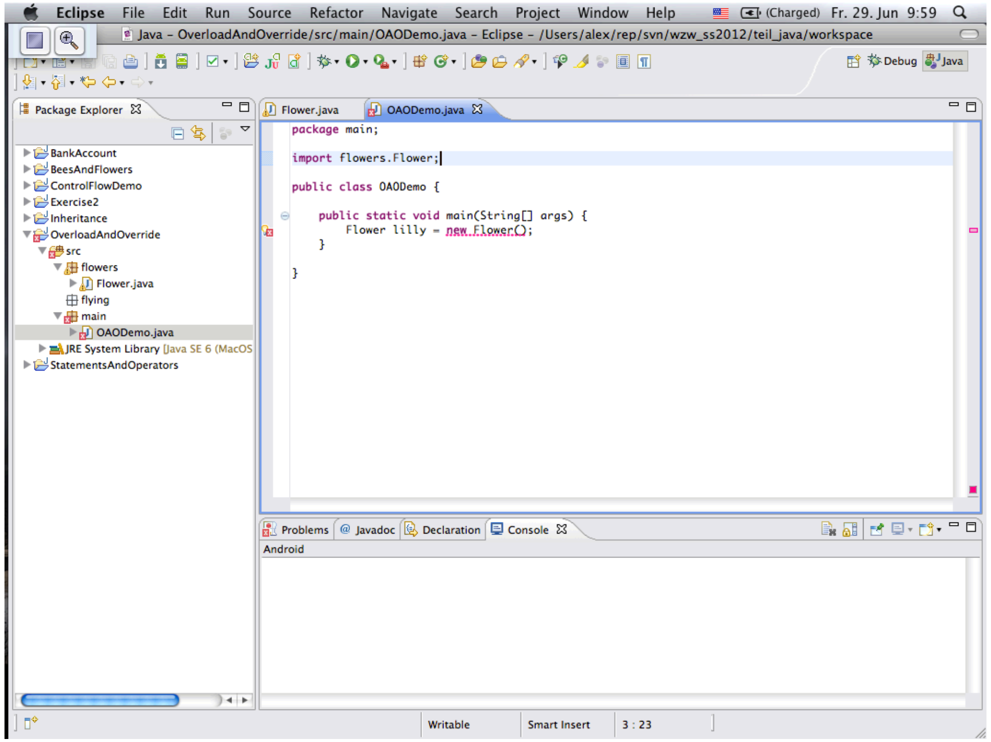

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 9:56
Java – OverloadAndOverride/src/flowers/Flower.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
Flower.java
Problems
Javadoc
Declaration
Console
Android
Writable
Smart Insert
11 : 70

```
package flowers;

public class Flower {

    private double amountOfPollen;

    public Flower(double initialAmountOfPollen) {
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

}
```



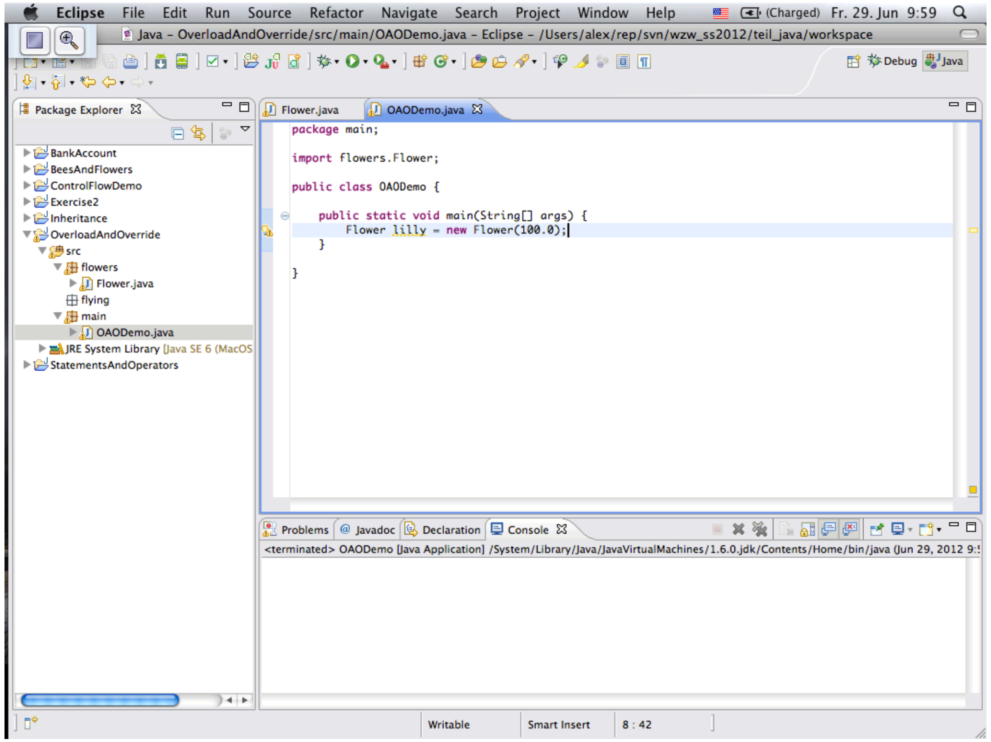
```
package flowers;

public class Flower {

    private double amountOfPollen;

    public Flower(double initialAmountOfPollen) {
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

}
```

```
Android
```



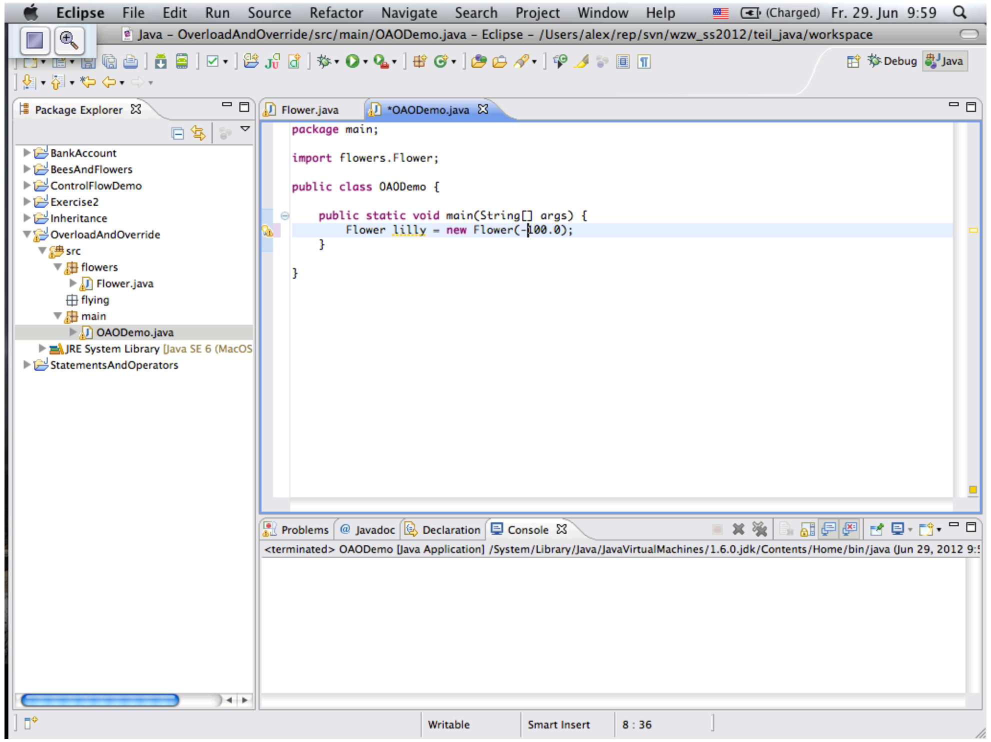
```
package main;

import flowers.Flower;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower();
    }

}
```

```
Android
```



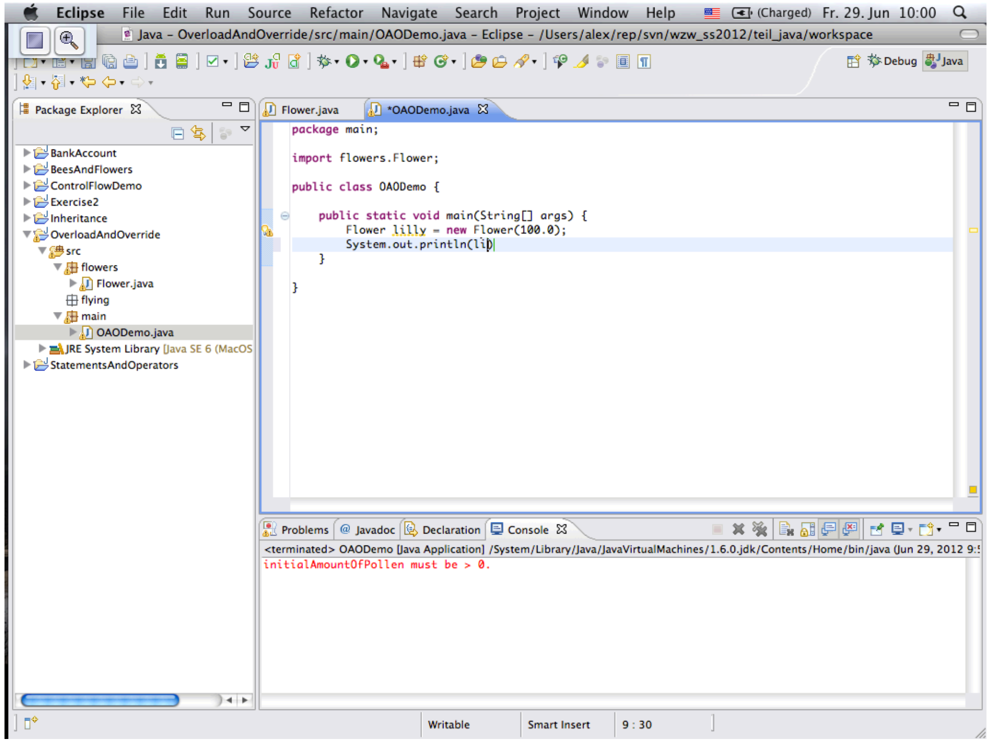
```
package main;

import flowers.Flower;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
    }

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 9:
```



```
package main;

import flowers.Flower;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(-100.0);
    }

}
```

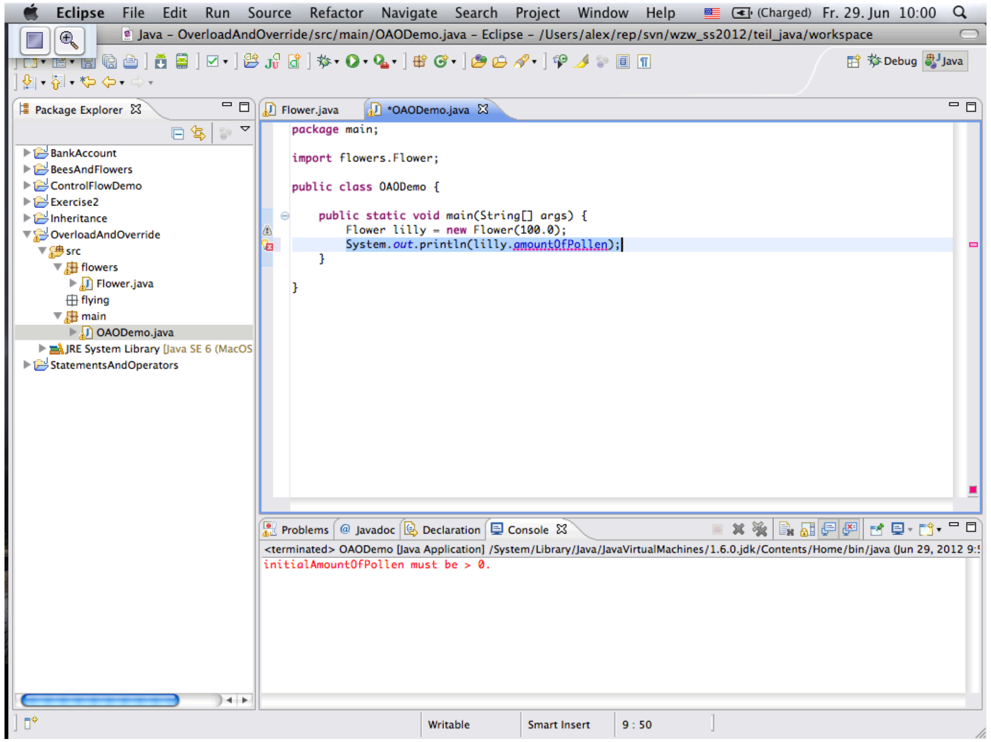
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 9:
```



Java – OverloadAndOverride/src/main/OAODemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package main;

import flowers.Flower;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        System.out.println(li)
    }

}
```

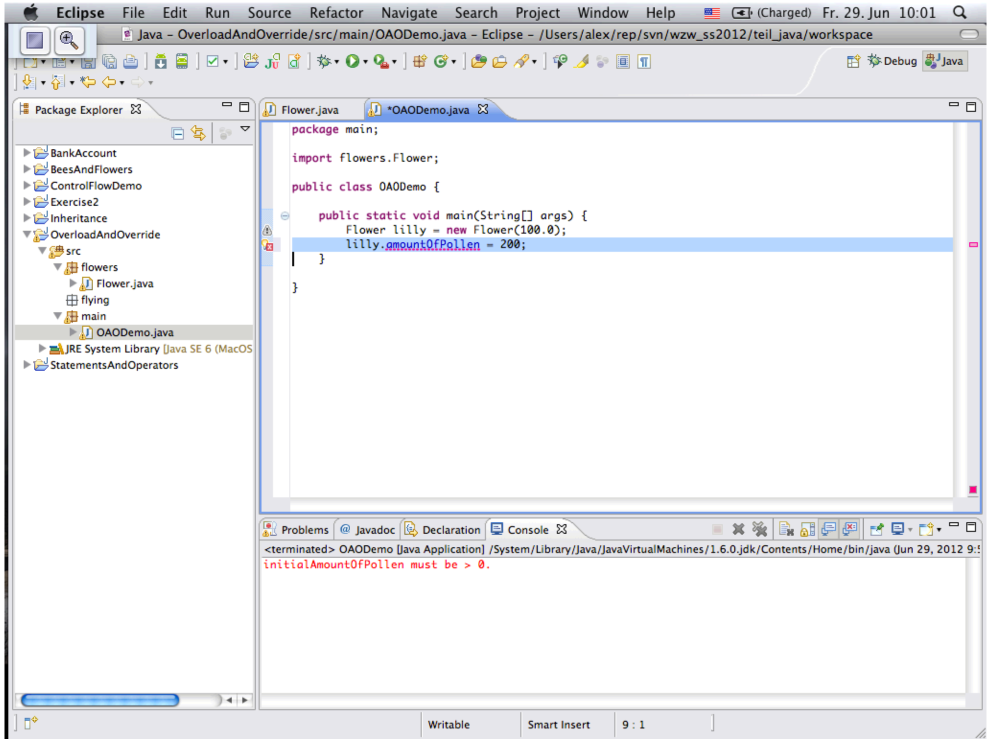
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 9:
initialAmountOfPollen must be > 0.
```



Java – OverloadAndOverride/src/main/OAODemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package main;

import flowers.Flower;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        System.out.println(lilly.amountOfPollen);
    }

}
```

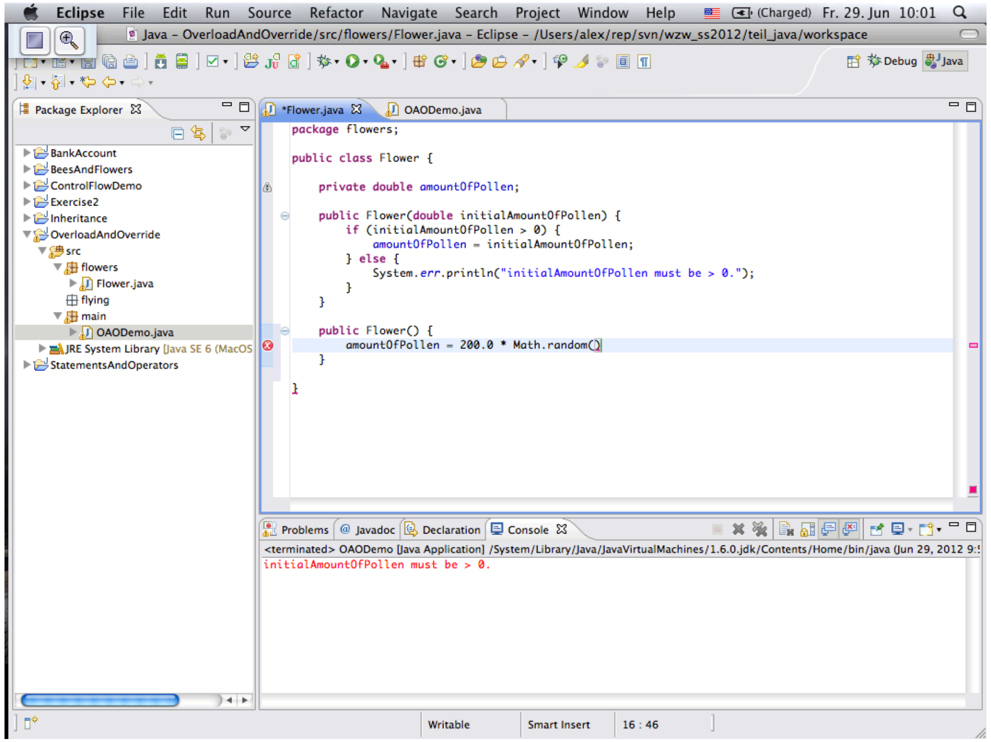
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 9:
initialAmountOfPollen must be > 0.
```



Java – OverloadAndOverride/src/main/OAODemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package main;

import flowers.Flower;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        lilly.amountOfPollen = 200;
    }

}
```

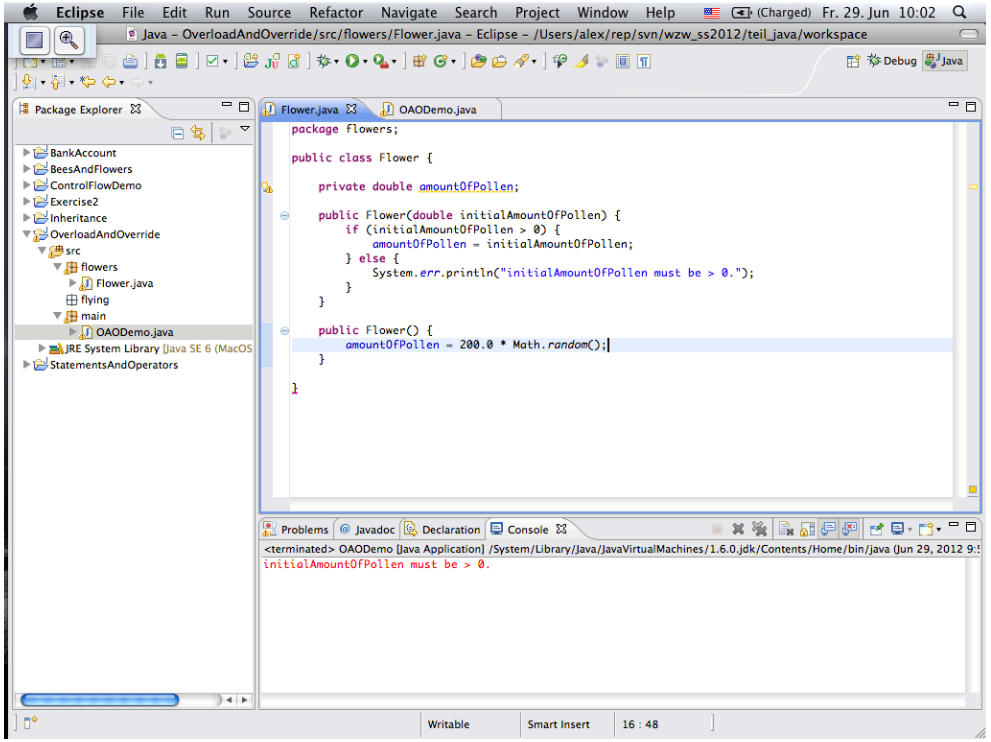
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 9:
initialAmountOfPollen must be > 0.
```



Java – OverloadAndOverride/src/flowers/Flower.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package flowers;

public class Flower {

    private double amountOfPollen;

    public Flower(double initialAmountOfPollen) {
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

    public Flower() {
        amountOfPollen = 200.0 * Math.random()
    }

}
```

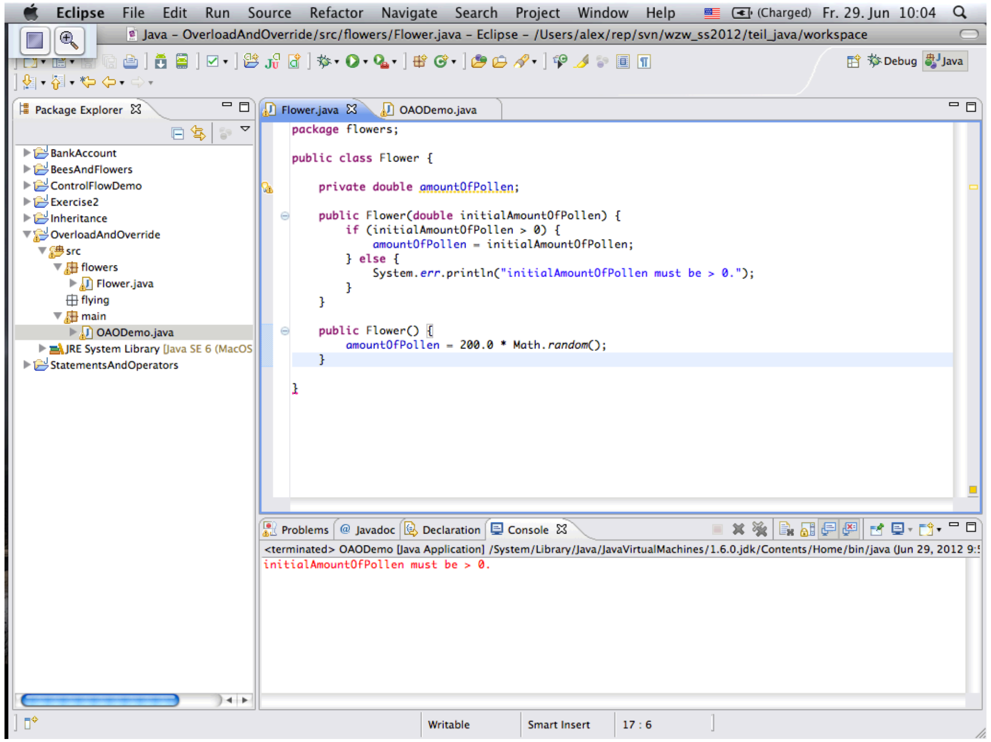
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 9:
initialAmountOfPollen must be > 0.
```



Java - OverloadAndOverride/src/flowers/Flower.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package flowers;

public class Flower {

    private double amountOfPollen;

    public Flower(double initialAmountOfPollen) {
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

    public Flower() {
        amountOfPollen = 200.0 * Math.random();
    }

}
```

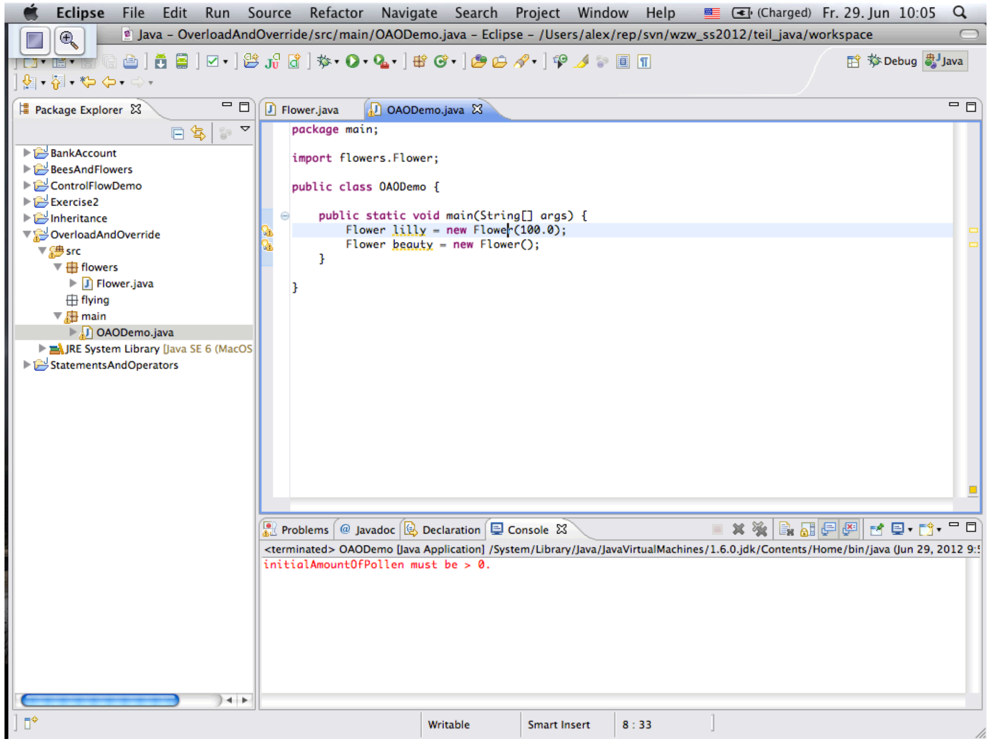
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 9:!
initialAmountOfPollen must be > 0.
```



Java - OverloadAndOverride/src/flowers/Flower.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package flowers;

public class Flower {

    private double amountOfPollen;

    public Flower(double initialAmountOfPollen) {
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

    public Flower() {
        amountOfPollen = 200.0 * Math.random();
    }

}
```

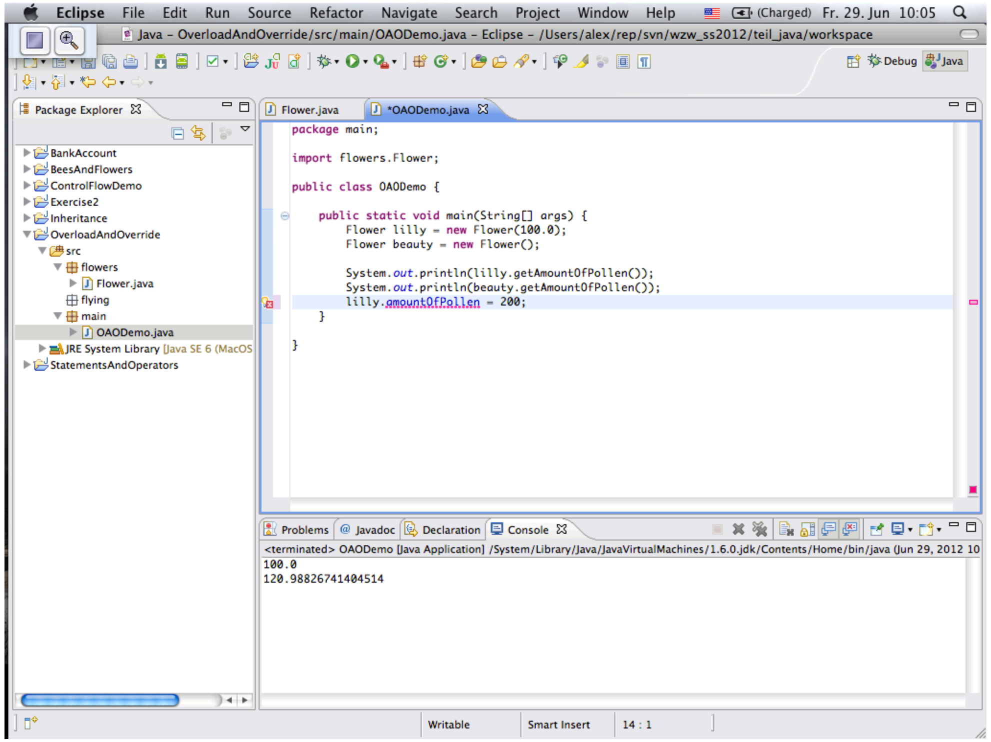
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 9:!
initialAmountOfPollen must be > 0.
```



Java - OverloadAndOverride/src/main/OAODemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package main;

import flowers.Flower;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();
    }

}
```

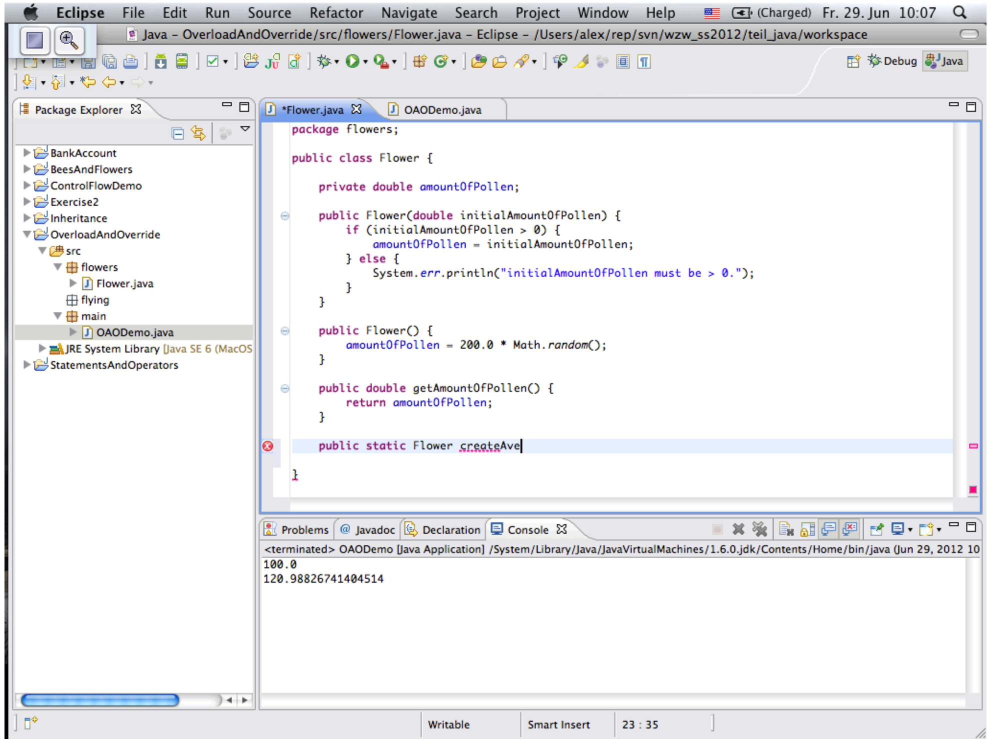
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 9:!
initialAmountOfPollen must be > 0.
```



Java - OverloadAndOverride/src/main/OAODemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package main;

import flowers.Flower;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());
        lilly.amountOfPollen = 200;
    }

}
```

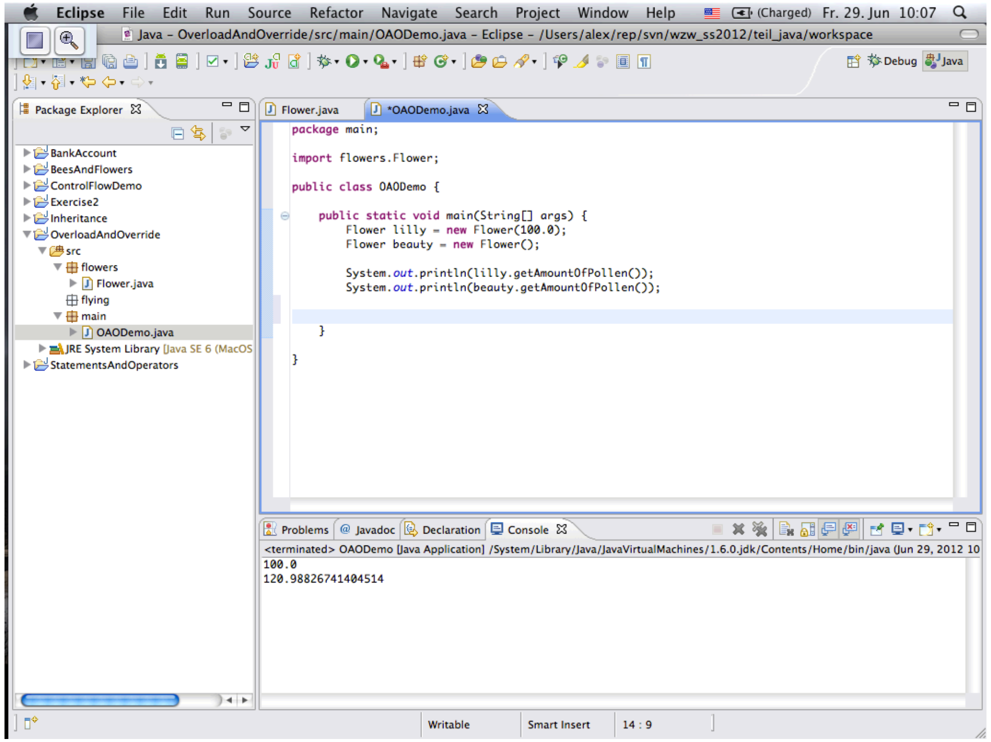
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
120.98826741404514
```



Java - OverloadAndOverride/src/flowers/Flower.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package flowers;

public class Flower {

    private double amountOfPollen;

    public Flower(double initialAmountOfPollen) {
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

    public Flower() {
        amountOfPollen = 200.0 * Math.random();
    }

    public double getAmountOfPollen() {
        return amountOfPollen;
    }

    public static Flower createAve

}
```

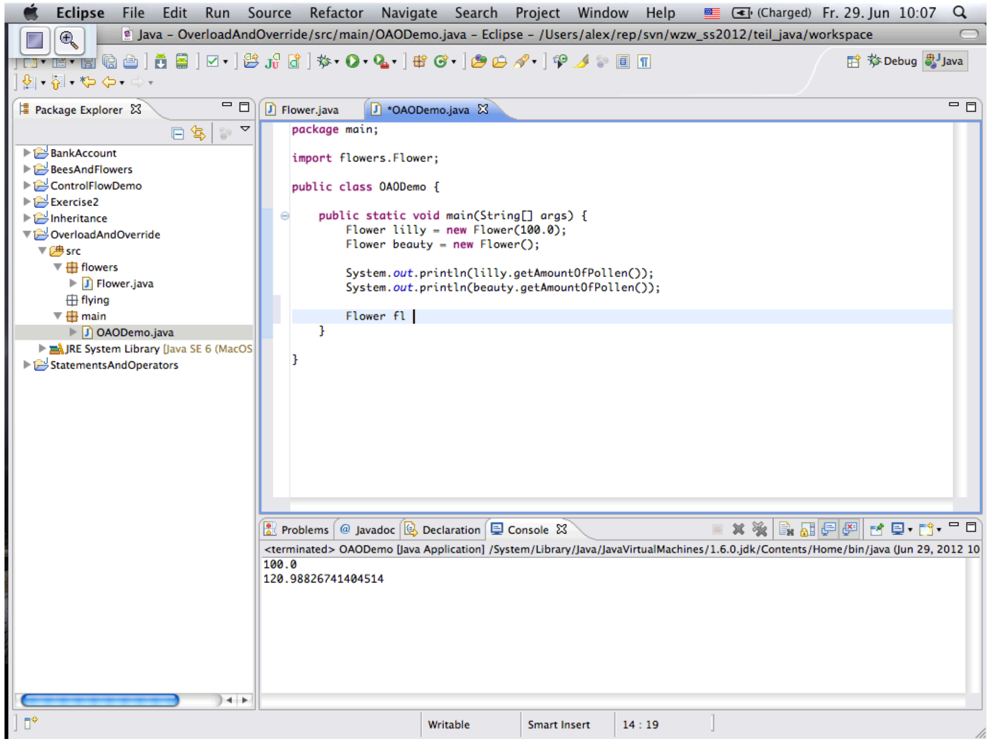
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
120.98826741404514
```



Java - OverloadAndOverride/src/main/OAODemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package main;

import flowers.Flower;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());


    }

}
```

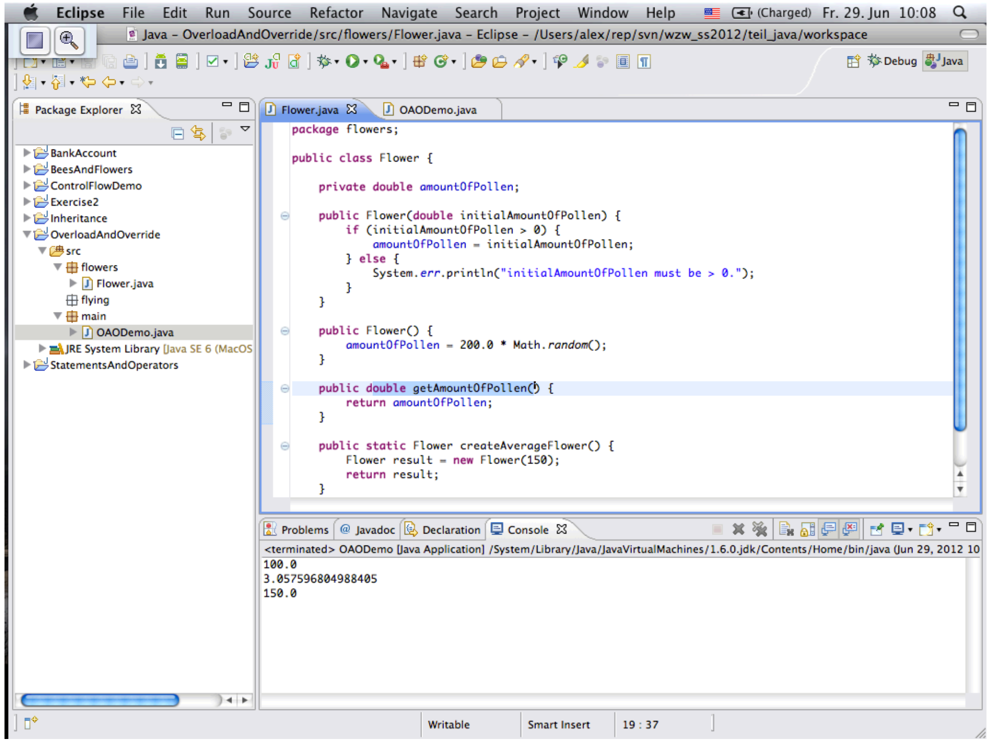
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
120.98826741404514
```



Java - OverloadAndOverride/src/main/OAODemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package main;

import flowers.Flower;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl 
    }

}
```

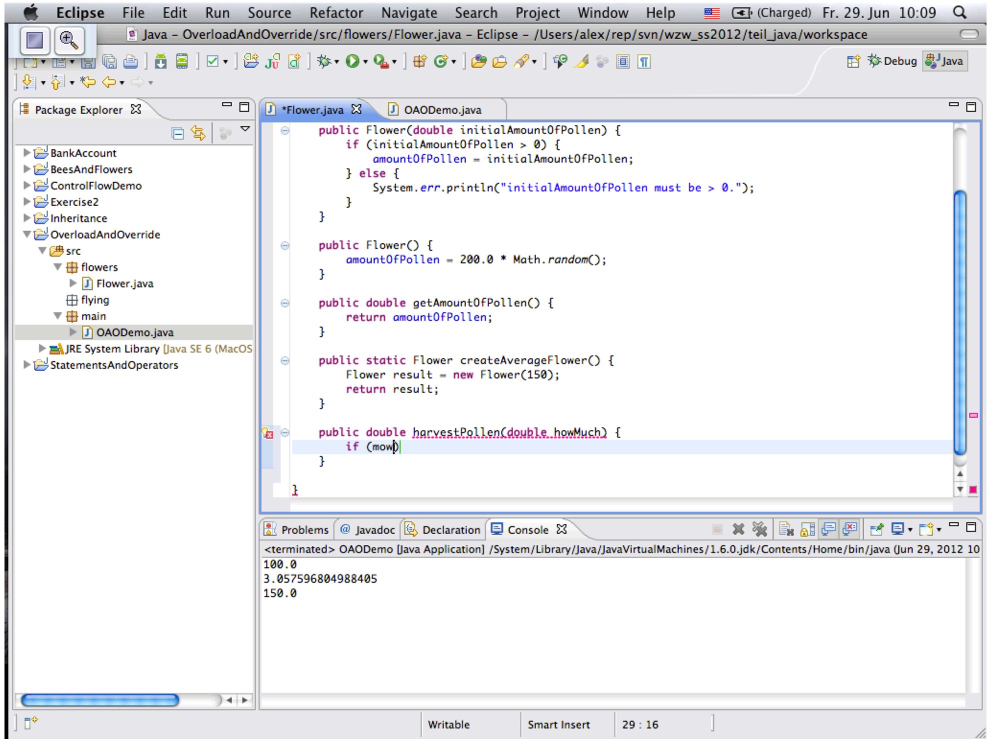
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
120.98826741404514
```



Java - OverloadAndOverride/src/flowers/Flower.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

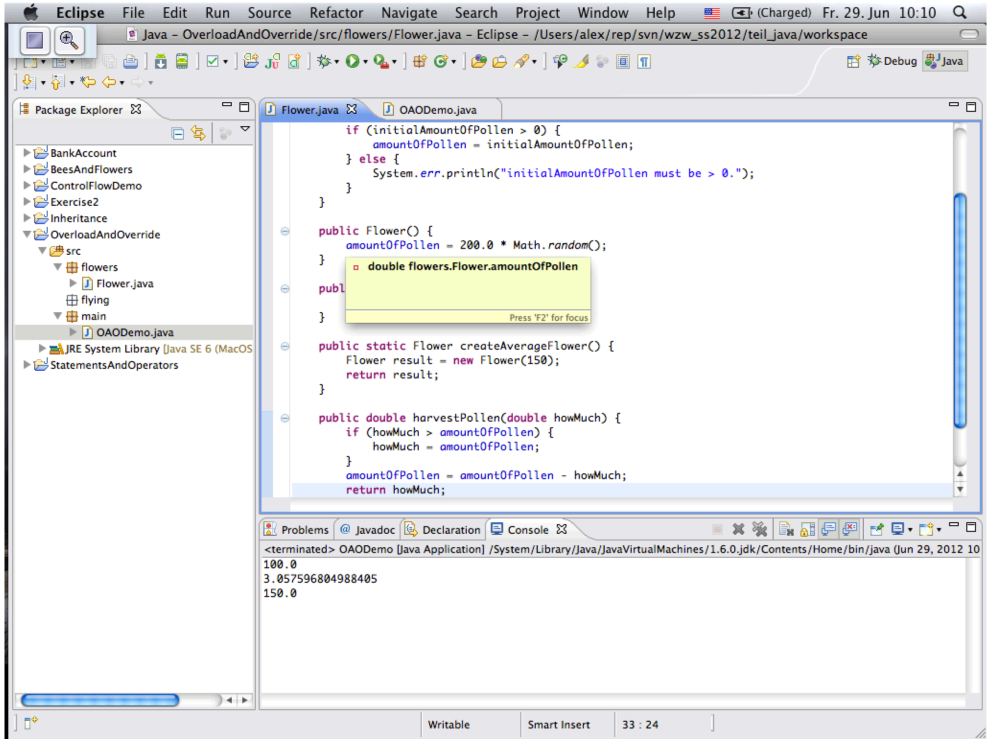
```java
package flowers;

public class Flower {

    private double amountOfPollen;

    public Flower(double initialAmountOfPollen) {
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

    public Flower() {
        amountOfPollen = 200.0 * Math.random();
    }

    public double getAmountOfPollen() {
        return amountOfPollen;
    }

    public static Flower createAverageFlower() {
        Flower result = new Flower(150);
        return result;
    }
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
3.057596804988405
150.0
```


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 29. Jun 10:09
Java – OverloadAndOverride/src/flowers/Flower.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug Java
Package Explorer
BankAccount
BeesAndFlowers
ControlFlowDemo
Exercise2
Inheritance
OverloadAndOverride
src
flowers
Flower.java
flying
main
OAODemo.java
JRE System Library [Java SE 6 (MacOS
StatementsAndOperators
*Flower.java
OAODemo.java
Problems Javadoc Declaration Console
Writable Smart Insert 29 : 16

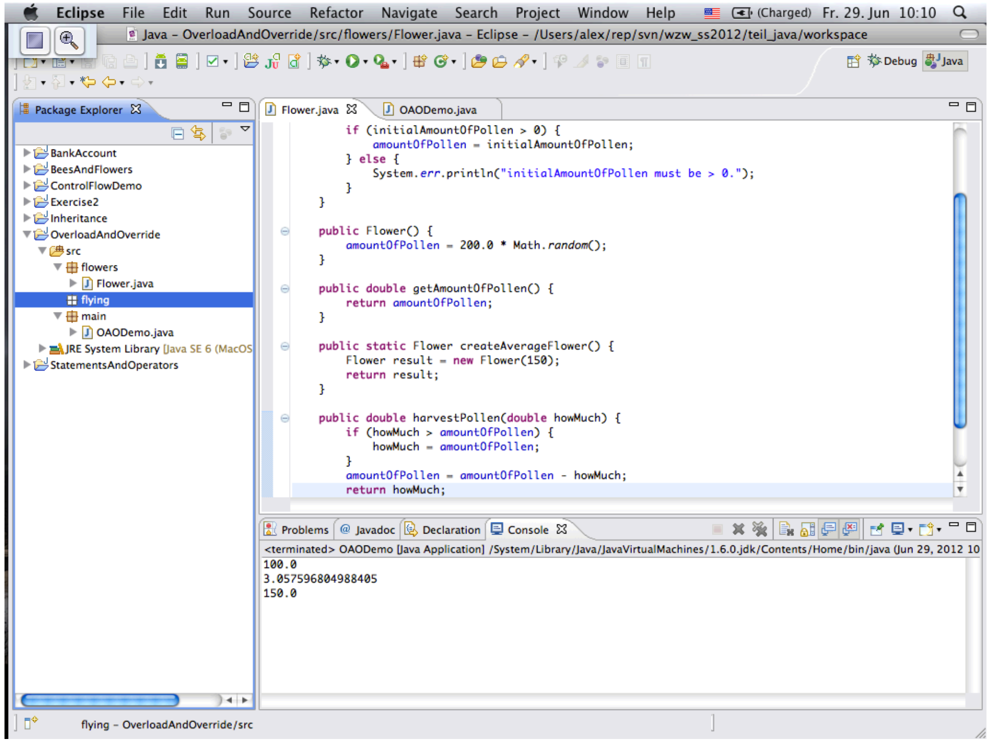
```
    public Flower(double initialAmountOfPollen) {
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

    public Flower() {
        amountOfPollen = 200.0 * Math.random();
    }

    public double getAmountOfPollen() {
        return amountOfPollen;
    }

    public static Flower createAverageFlower() {
        Flower result = new Flower(150);
        return result;
    }

    public double harvestPollen(double howMuch) {
        if (mow)
    }

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
3.057596804988405
150.0
```


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 29. Jun 10:10
Java – OverloadAndOverride/src/flowers/Flower.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug Java
Package Explorer
BankAccount
BeesAndFlowers
ControlFlowDemo
Exercise2
Inheritance
OverloadAndOverride
src
flowers
Flower.java
flying
main
OAODemo.java
JRE System Library [Java SE 6 (MacOS
StatementsAndOperators
Flower.java
OAODemo.java
double flowers.Flower.amountOfPollen
Press 'F2' for focus
Problems Javadoc Declaration Console
Writable Smart Insert 33 : 24

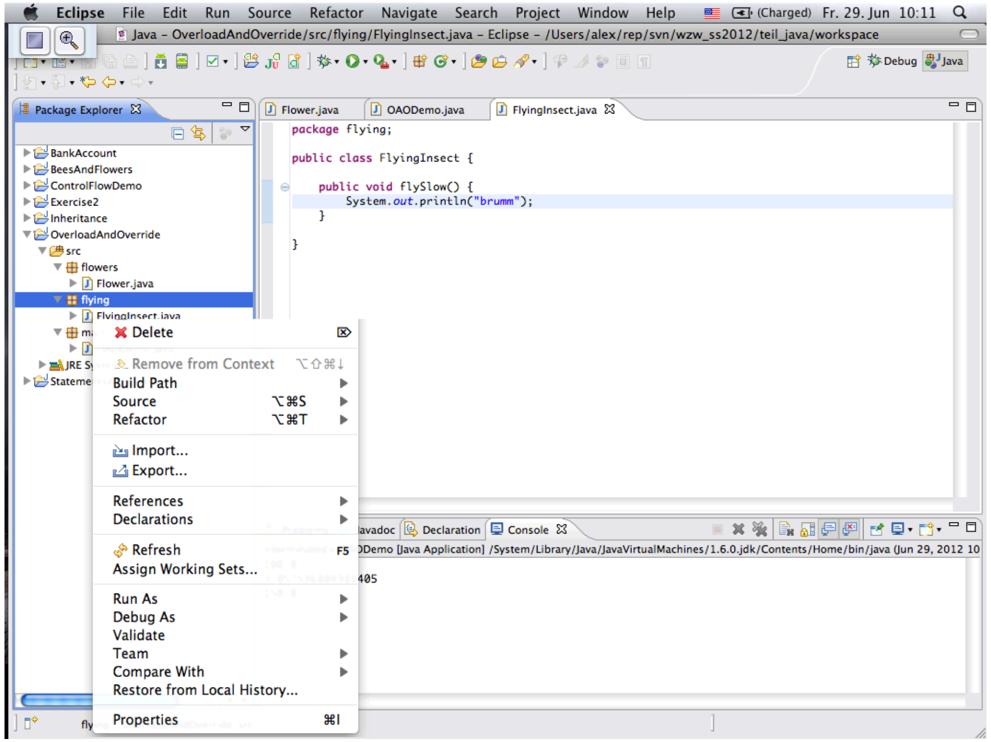
```
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

    public Flower() {
        amountOfPollen = 200.0 * Math.random();
    }

    publ

    }

    public static Flower createAverageFlower() {
        Flower result = new Flower(150);
        return result;
    }

    public double harvestPollen(double howMuch) {
        if (howMuch > amountOfPollen) {
            howMuch = amountOfPollen;
        }
        amountOfPollen = amountOfPollen - howMuch;
        return howMuch;
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
3.057596804988405
150.0
```


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 29. Jun 10:10
Java – OverloadAndOverride/src/flowers/Flower.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug Java
Package Explorer
BankAccount
BeesAndFlowers
ControlFlowDemo
Exercise2
Inheritance
OverloadAndOverride
src
flowers
Flower.java
flying
main
OAODemo.java
JRE System Library [Java SE 6 (MacOS
StatementsAndOperators
Flower.java
OAODemo.java
Problems Javadoc Declaration Console
flying – OverloadAndOverride/src

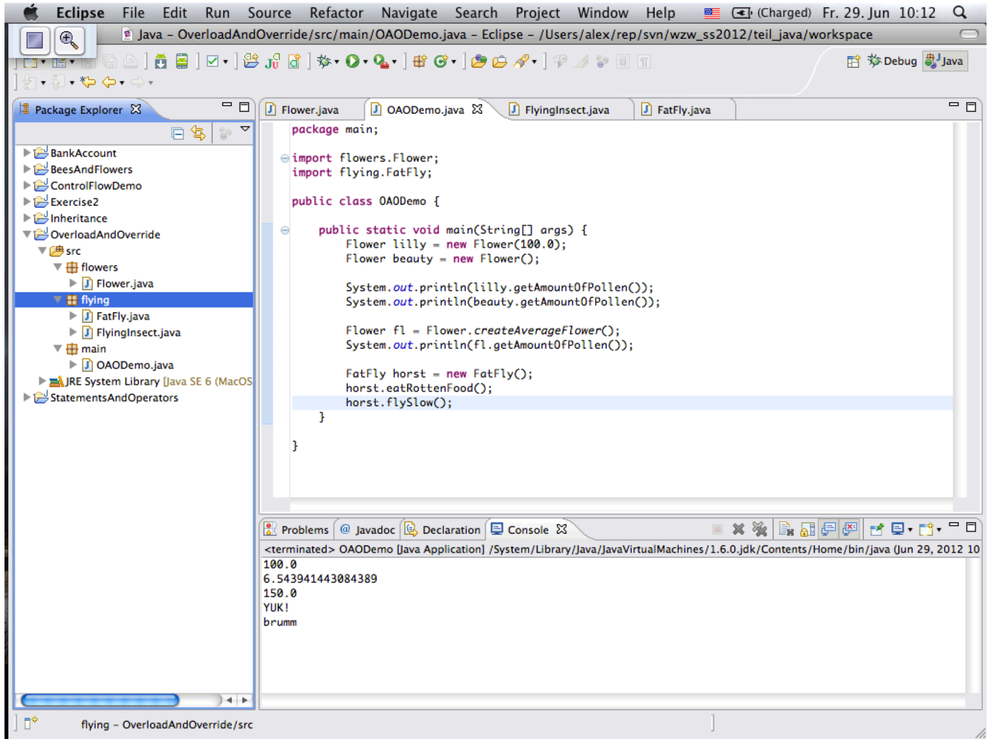
```
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

    public Flower() {
        amountOfPollen = 200.0 * Math.random();
    }

    public double getAmountOfPollen() {
        return amountOfPollen;
    }

    public static Flower createAverageFlower() {
        Flower result = new Flower(150);
        return result;
    }

    public double harvestPollen(double howMuch) {
        if (howMuch > amountOfPollen) {
            howMuch = amountOfPollen;
        }
        amountOfPollen = amountOfPollen - howMuch;
        return howMuch;
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
3.057596804988405
150.0
```



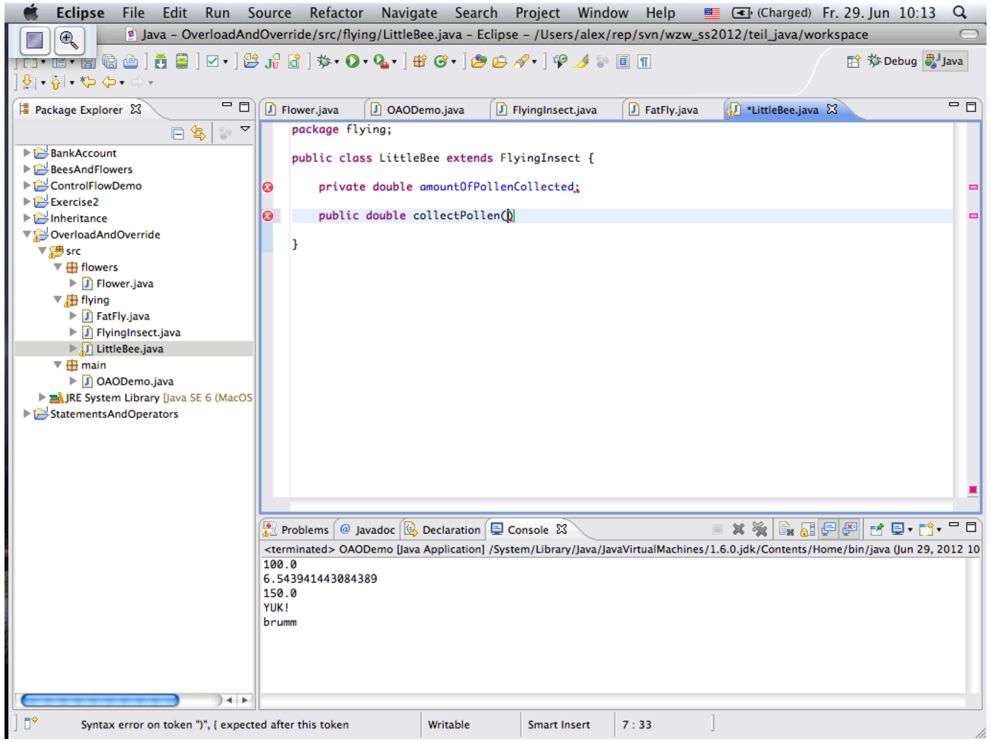
```
package flying;

public class FlyingInsect {

    public void flySlow() {
        System.out.println("brumm");
    }

}
```

```
Demo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
405
```



```
package main;

import flowers.Flower;
import flying.FatFly;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();
    }

}
```

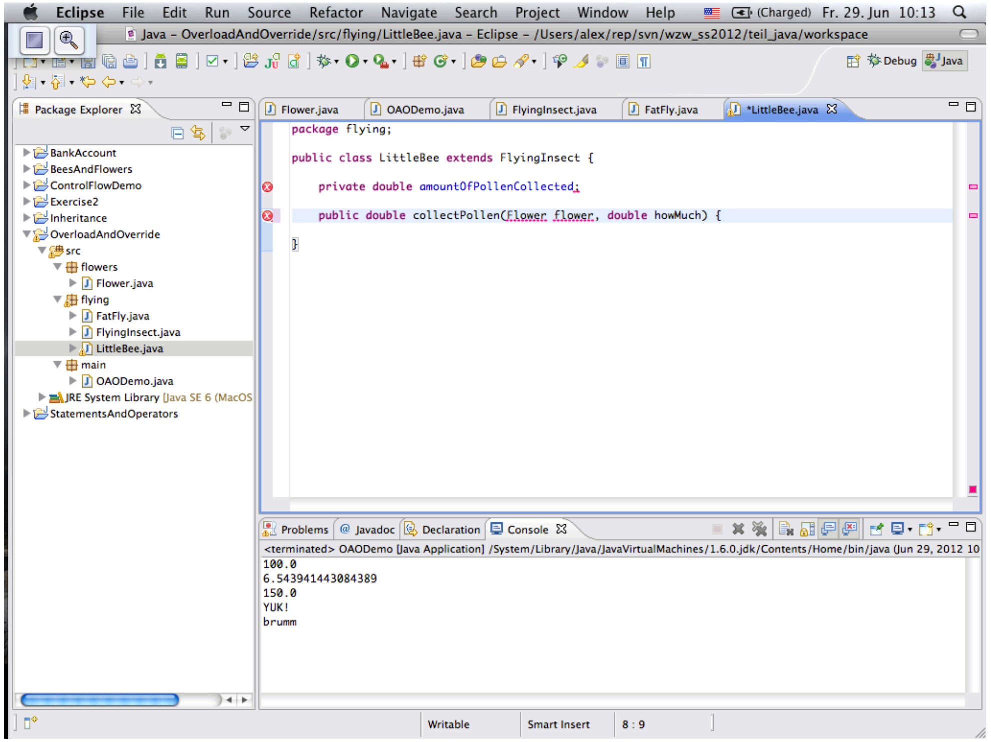
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
6.543941443084389
150.0
YUK!
brumm
```



```
package flying;

public class LittleBee extends FlyingInsect {

    private double amountOfPollenCollected;

    public double collectPollen()

}
```

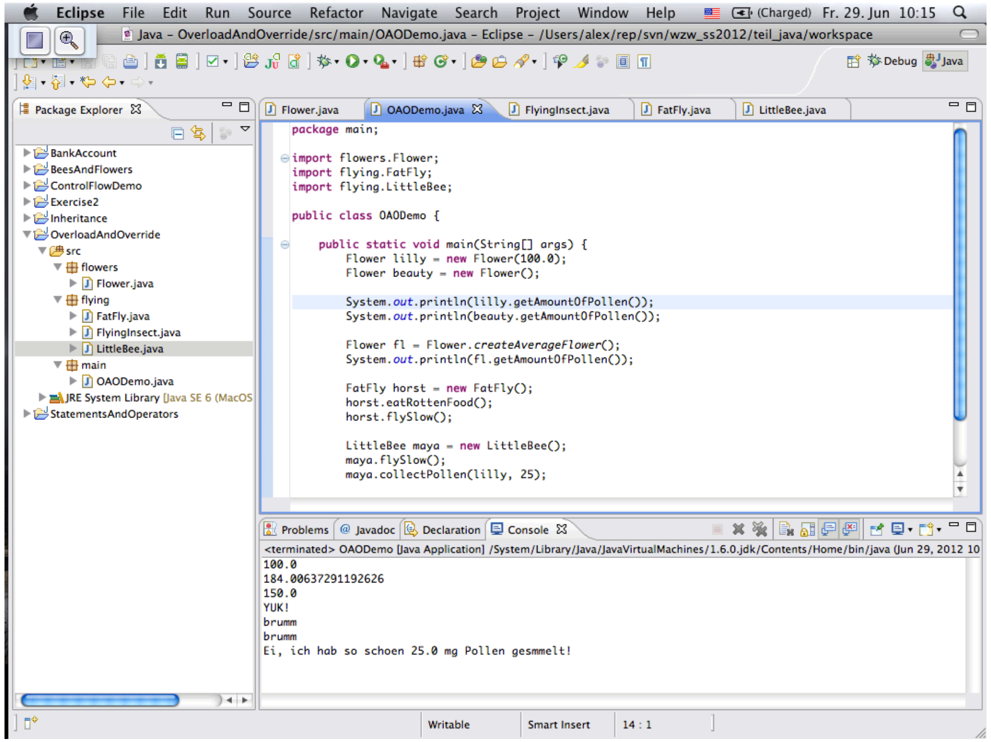
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
6.543941443084389
150.0
YUK!
brumm
```

Syntax error on token ")", { expected after this token



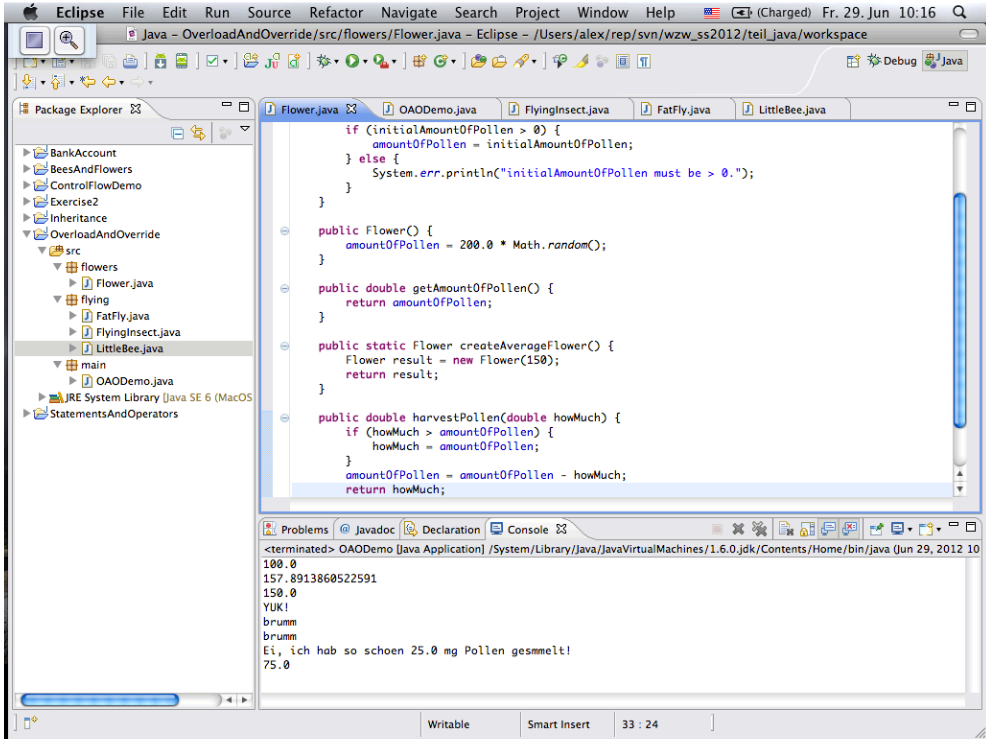
```
package flying;

public class LittleBee extends FlyingInsect {

    private double amountOfPollenCollected;

    public double collectPollen(Flower flower, double howMuch) {

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
6.543941443084389
150.0
YUK!
brumm
```



```
package main;

import flowers.Flower;
import flying.FatFly;
import flying.LittleBee;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
184.00637291192626
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
```

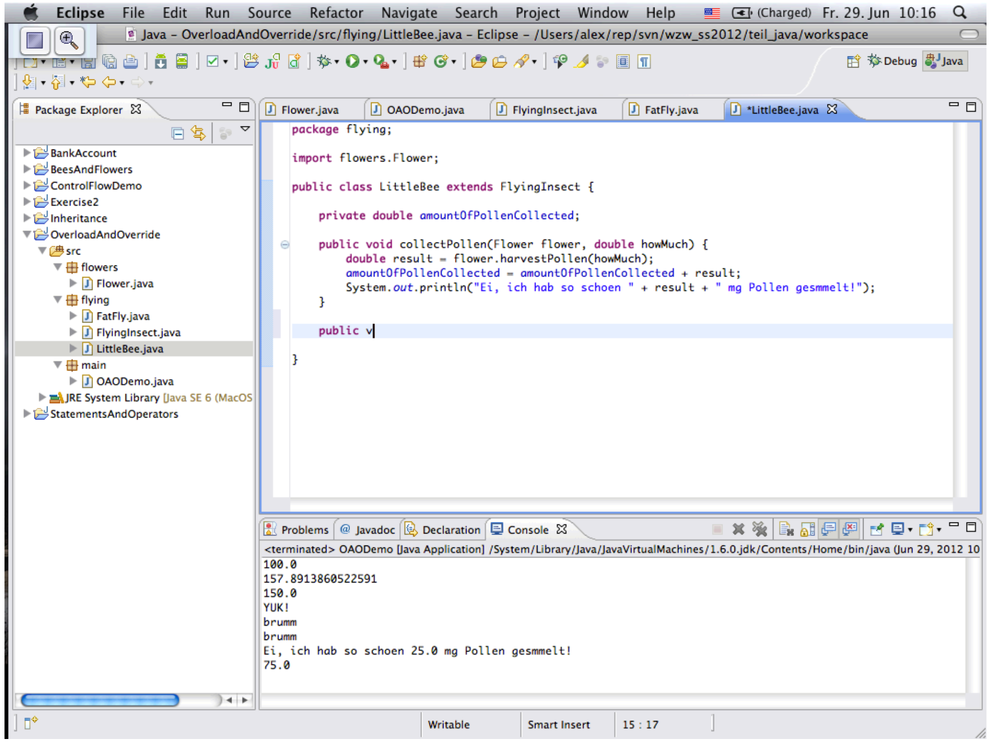


Java - OverloadAndOverride/src/flowers/Flower.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
        if (initialAmountOfPollen > 0) {
            amountOfPollen = initialAmountOfPollen;
        } else {
            System.err.println("initialAmountOfPollen must be > 0.");
        }
    }

    public Flower() {
        amountOfPollen = 200.0 * Math.random();
    }

    public double getAmountOfPollen() {
        return amountOfPollen;
    }

    public static Flower createAverageFlower() {
        Flower result = new Flower(150);
        return result;
    }

    public double harvestPollen(double howMuch) {
        if (howMuch > amountOfPollen) {
            howMuch = amountOfPollen;
        }
        amountOfPollen = amountOfPollen - howMuch;
        return howMuch;
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
157.8913860522591
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
```

Writable | Smart Insert | 33 : 24

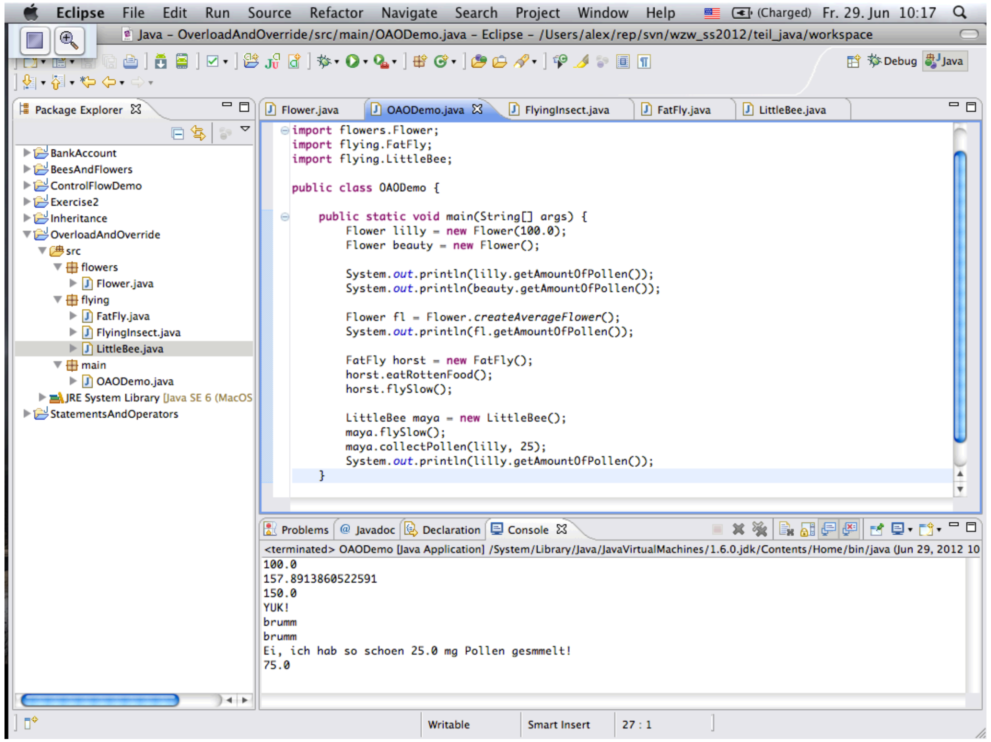


Java - OverloadAndOverride/src/flying/LittleBee.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package flying;

import flowers.Flower;

public class LittleBee extends FlyingInsect {

    private double amountOfPollenCollected;

    public void collectPollen(Flower flower, double howMuch) {
        double result = flower.harvestPollen(howMuch);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

    public v

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
157.8913860522591
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
```

Writable | Smart Insert | 15 : 17

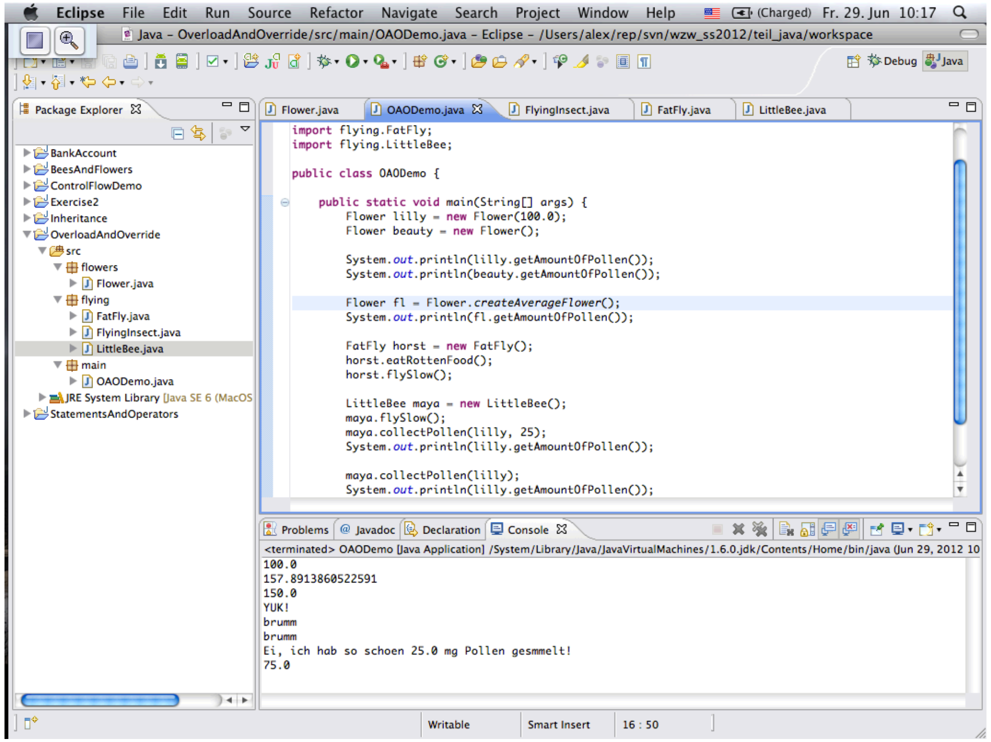


Java - OverloadAndOverride/src/main/OAODemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
import flowers.Flower;
import flying.FatFly;
import flying.LittleBee;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
        System.out.println(lilly.getAmountOfPollen());
    }
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
157.8913860522591
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
```

Writable | Smart Insert | 27 : 1

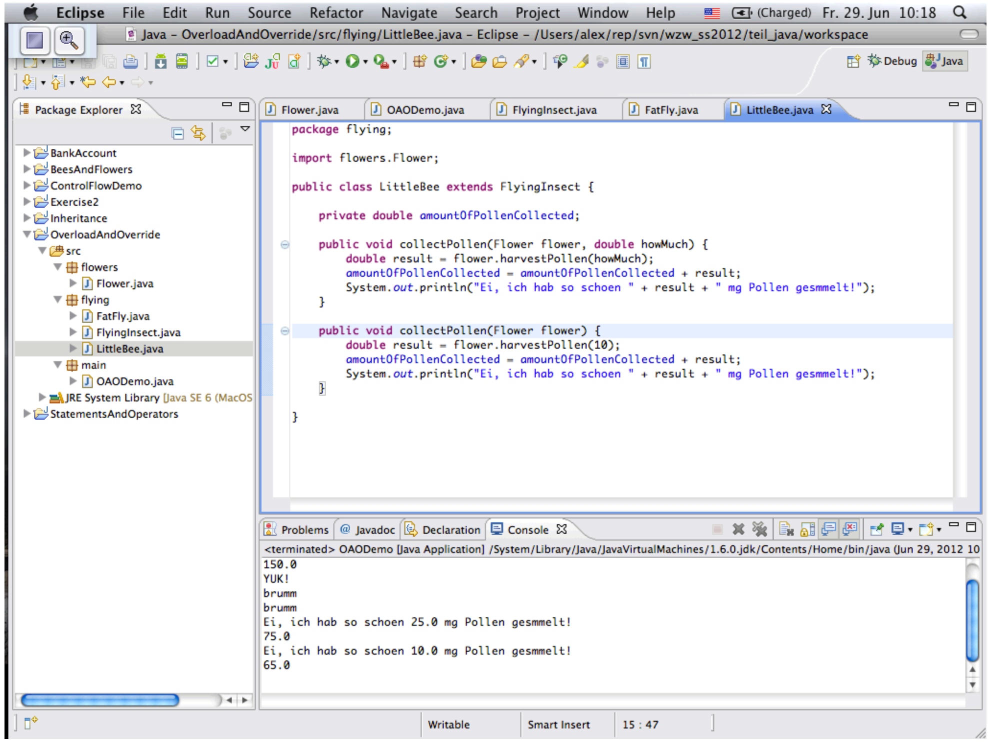


Java - OverloadAndOverride/src/main/OAODemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
import flying.FatFly;
import flying.LittleBee;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
        System.out.println(lilly.getAmountOfPollen());

        maya.collectPollen(lilly);
        System.out.println(lilly.getAmountOfPollen());
```

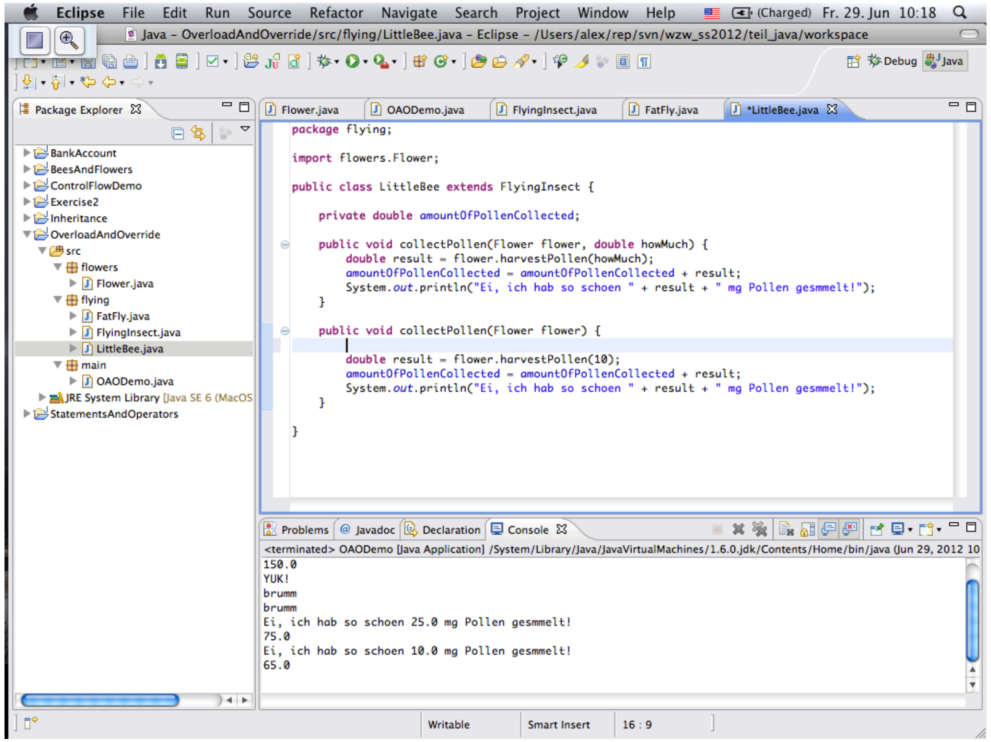
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
100.0
157.8913860522591
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
```

Writable | Smart Insert | 16 : 50



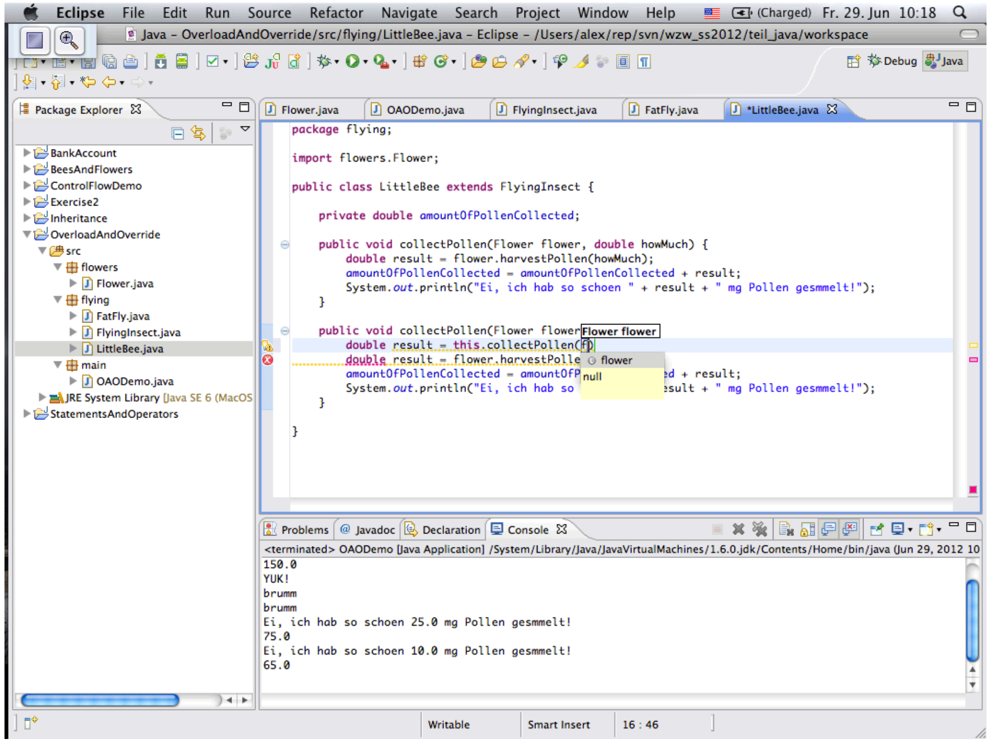
```
package flying;

import flowers.Flower;

public class LittleBee extends FlyingInsect {

    private double amountOfPollenCollected;

    public void collectPollen(Flower flower, double howMuch) {
        double result = flower.harvestPollen(howMuch);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

    public void collectPollen(Flower flower) {
        double result = flower.harvestPollen(10);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
```



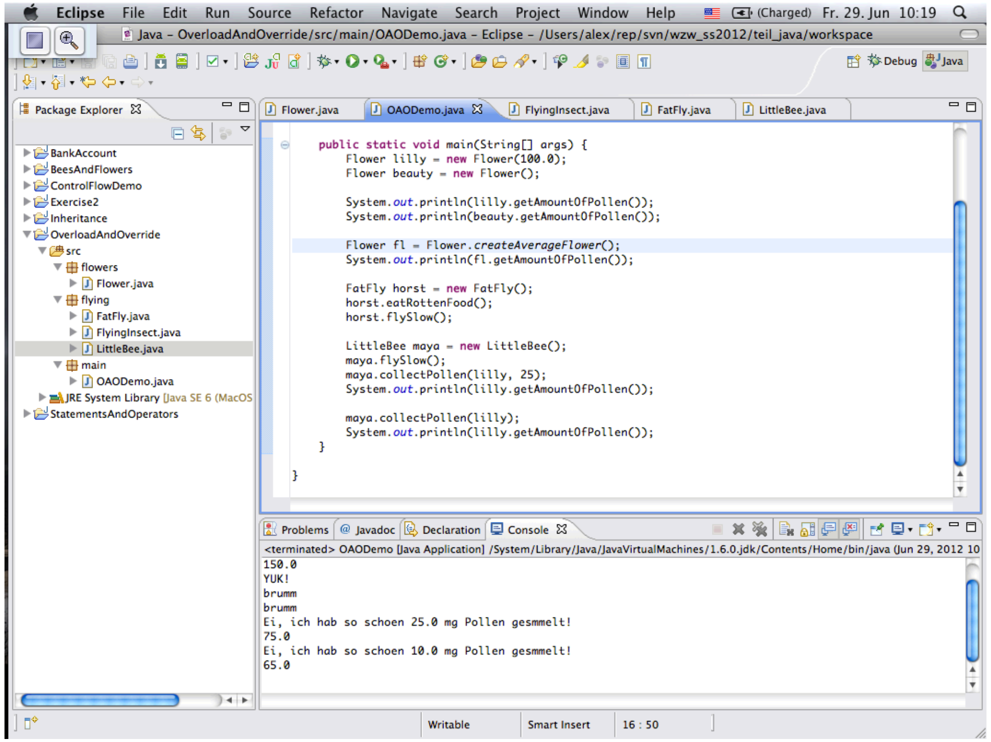
```
package flying;

import flowers.Flower;

public class LittleBee extends FlyingInsect {

    private double amountOfPollenCollected;

    public void collectPollen(Flower flower, double howMuch) {
        double result = flower.harvestPollen(howMuch);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

    public void collectPollen(Flower flower) {
        
        double result = flower.harvestPollen(10);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
```



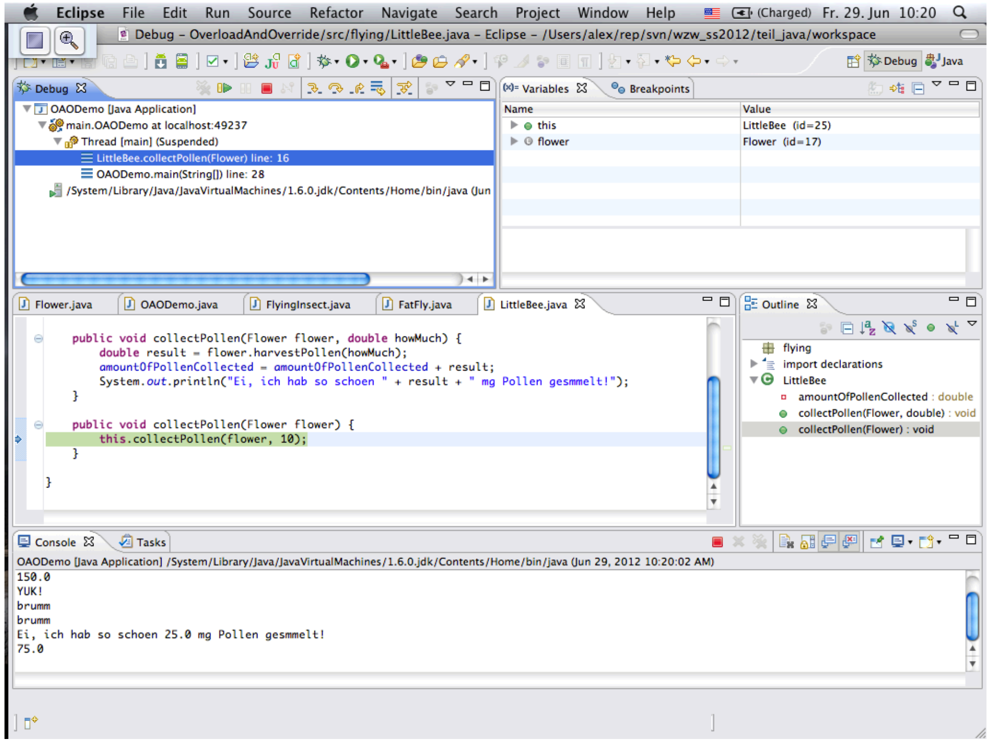
```
package flying;

import flowers.Flower;

public class LittleBee extends FlyingInsect {

    private double amountOfPollenCollected;

    public void collectPollen(Flower flower, double howMuch) {
        double result = flower.harvestPollen(howMuch);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

    public void collectPollen(Flower flower
        double result = this.collectPollen(f)
        double result = flower.harvestPolle
        amountOfPollenCollected = amountOfP                  ed + result;
        System.out.println("Ei, ich hab so                   esult + " mg Pollen gesmmelt!");
    }

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
```



```
    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
        System.out.println(lilly.getAmountOfPollen());

        maya.collectPollen(lilly);
        System.out.println(lilly.getAmountOfPollen());
    }

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
```


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 10:20
Debug
Java
Variables
Breakpoints
Flower.java
OAODemo.java
FlyingInsect.java
FatFly.java
LittleBee.java
Outline
flying
import declarations
LittleBee
amountOfPollenCollected : double
collectPollen(Flower, double) : void
collectPollen(Flower) : void
Console
Tasks

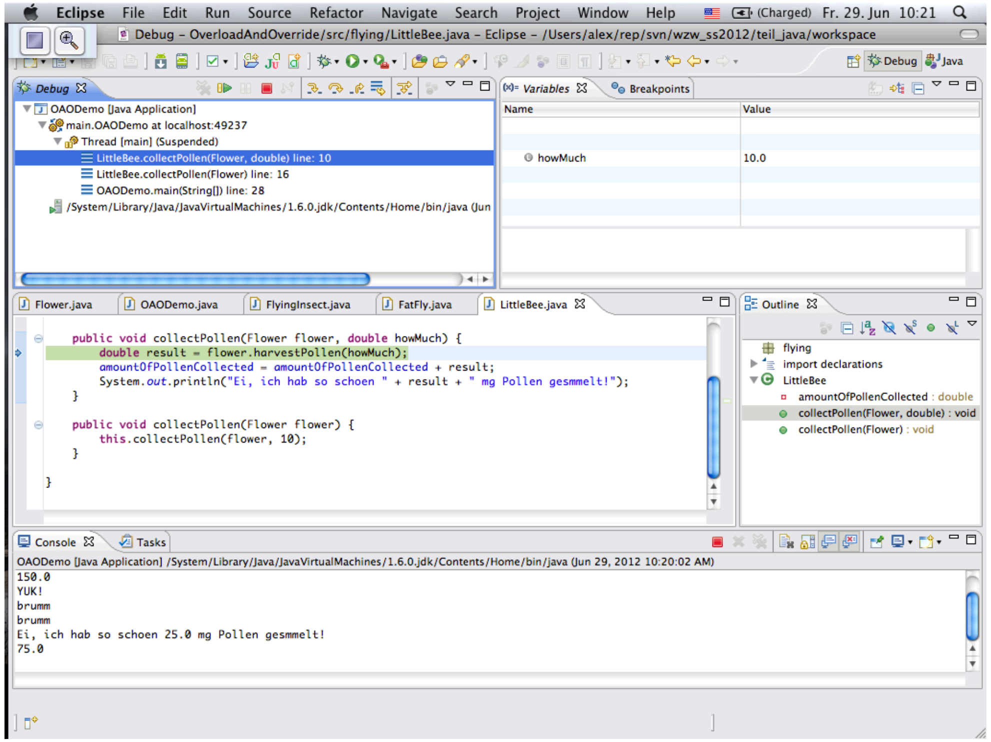
Debug - OverloadAndOverride/src/flying/LittleBee.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```
OAODemo [Java Application]
  main.OAODemo at localhost:49237
    Thread [main] (Suspended)
      LittleBee.collectPollen(Flower) line: 16
      OAODemo.main(String[]) line: 28
  /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
```

Name	Value
this	LittleBee (id=25)
flower	Flower (id=17)

```
    public void collectPollen(Flower flower, double howMuch) {
        double result = flower.harvestPollen(howMuch);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

    public void collectPollen(Flower flower) {
        this.collectPollen(flower, 10);
    }

}
```

```
OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:20:02 AM)
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
```


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 10:21
Debug
Java
Variables
Breakpoints
Flower.java
OAODemo.java
FlyingInsect.java
FatFly.java
LittleBee.java
Outline
flying
import declarations
LittleBee
amountOfPollenCollected : double
collectPollen(Flower, double) : void
collectPollen(Flower) : void
Console
Tasks

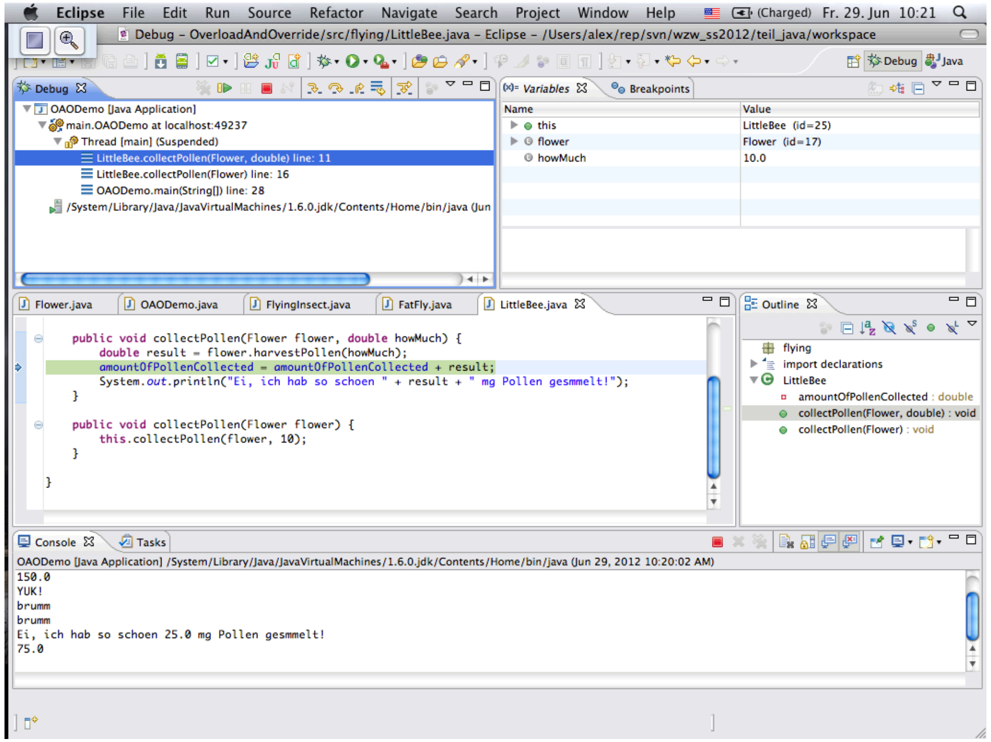
Debug - OverloadAndOverride/src/flying/LittleBee.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```
OAODemo [Java Application]
  main.OAODemo at localhost:49237
    Thread [main] (Suspended)
      LittleBee.collectPollen(Flower, double) line: 10
      LittleBee.collectPollen(Flower) line: 16
      OAODemo.main(String[]) line: 28
  /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
```

Name	Value
howMuch	10.0

```
    public void collectPollen(Flower flower, double howMuch) {
        double result = flower.harvestPollen(howMuch);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

    public void collectPollen(Flower flower) {
        this.collectPollen(flower, 10);
    }

}
```

```
OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:20:02 AM)
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
```


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 10:21
Debug
Java
Variables
Breakpoints
Flower.java
OAODemo.java
FlyingInsect.java
FatFly.java
LittleBee.java
Outline
flying
import declarations
LittleBee
amountOfPollenCollected : double
collectPollen(Flower, double) : void
collectPollen(Flower) : void
Console
Tasks

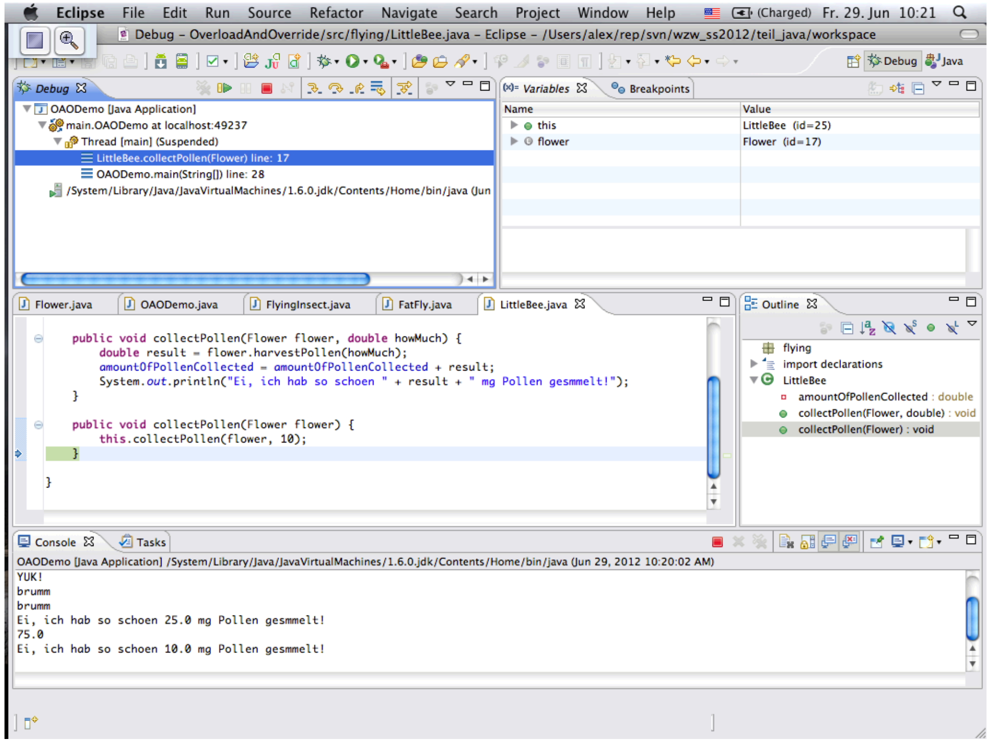
Debug - OverloadAndOverride/src/flying/LittleBee.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```
OAODemo [Java Application]
  main.OAODemo at localhost:49237
    Thread [main] (Suspended)
      LittleBee.collectPollen(Flower, double) line: 11
      LittleBee.collectPollen(Flower) line: 16
      OAODemo.main(String[]) line: 28
  /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
```

Name	Value
this	LittleBee (id=25)
flower	Flower (id=17)
howMuch	10.0

```
    public void collectPollen(Flower flower, double howMuch) {
        double result = flower.harvestPollen(howMuch);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

    public void collectPollen(Flower flower) {
        this.collectPollen(flower, 10);
    }

}
```

```
OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:20:02 AM)
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
```


Eclipse File Edit Run Source Refactor Navigate Search Project Window Help
(Charged) Fr. 29. Jun 10:21
Debug
Java
Variables
Breakpoints
Flower.java
OAODemo.java
FlyingInsect.java
FatFly.java
LittleBee.java
Outline
flying
import declarations
LittleBee
amountOfPollenCollected : double
collectPollen(Flower, double) : void
collectPollen(Flower) : void
Console
Tasks

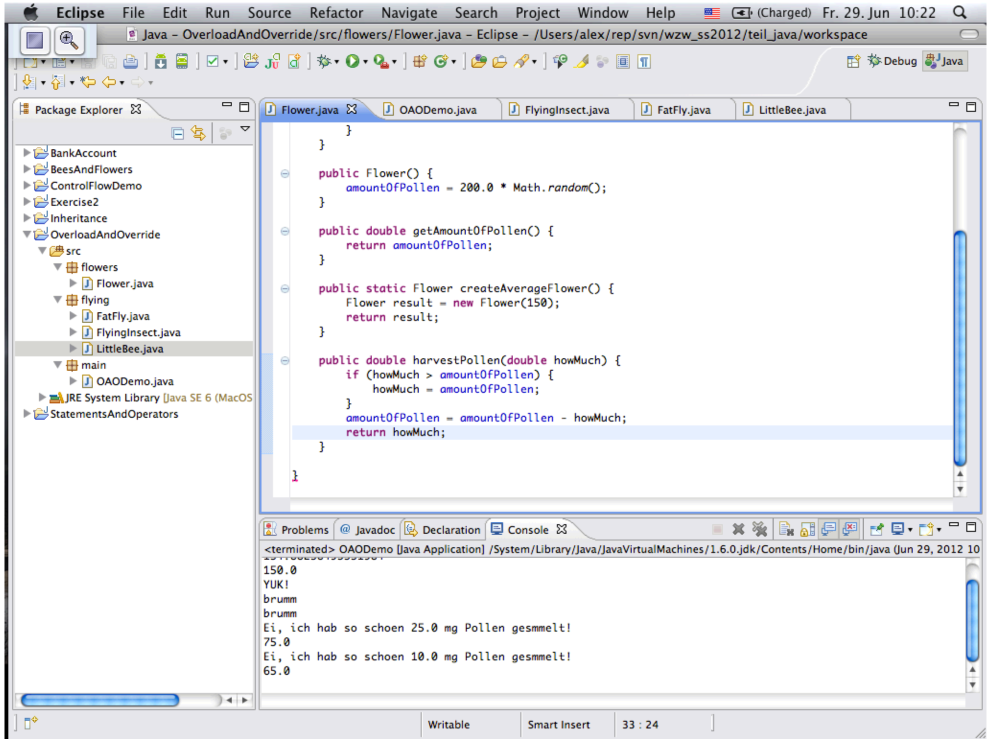
Debug - OverloadAndOverride/src/flying/LittleBee.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```
OAODemo [Java Application]
  main.OAODemo at localhost:49237
    Thread [main] (Suspended)
      LittleBee.collectPollen(Flower) line: 17
      OAODemo.main(String[]) line: 28
  /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
```

Name	Value
this	LittleBee (id=25)
flower	Flower (id=17)

```
    public void collectPollen(Flower flower, double howMuch) {
        double result = flower.harvestPollen(howMuch);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

    public void collectPollen(Flower flower) {
        this.collectPollen(flower, 10);
    }

}
```

```
OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:20:02 AM)
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
```

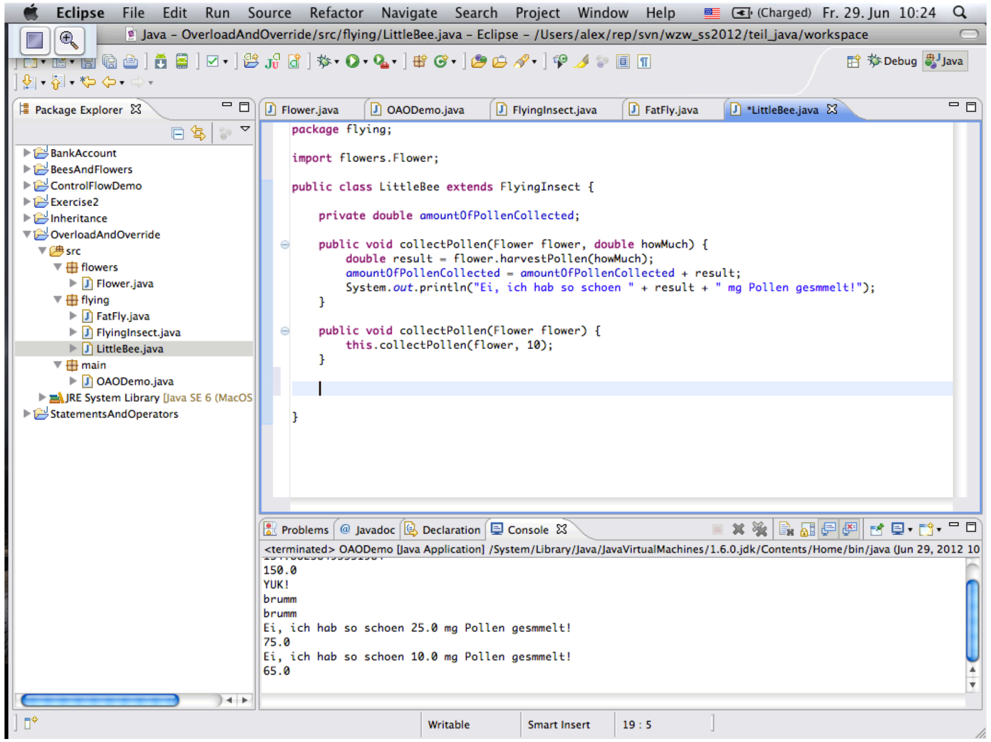


Java - OverloadAndOverride/src/flowers/Flower.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
        }
    }

    public Flower() {
        amountOfPollen = 200.0 * Math.random();
    }

    public double getAmountOfPollen() {
        return amountOfPollen;
    }

    public static Flower createAverageFlower() {
        Flower result = new Flower(150);
        return result;
    }

    public double harvestPollen(double howMuch) {
        if (howMuch > amountOfPollen) {
            howMuch = amountOfPollen;
        }
        amountOfPollen = amountOfPollen - howMuch;
        return howMuch;
    }

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
```

Writable | Smart Insert | 33 : 24

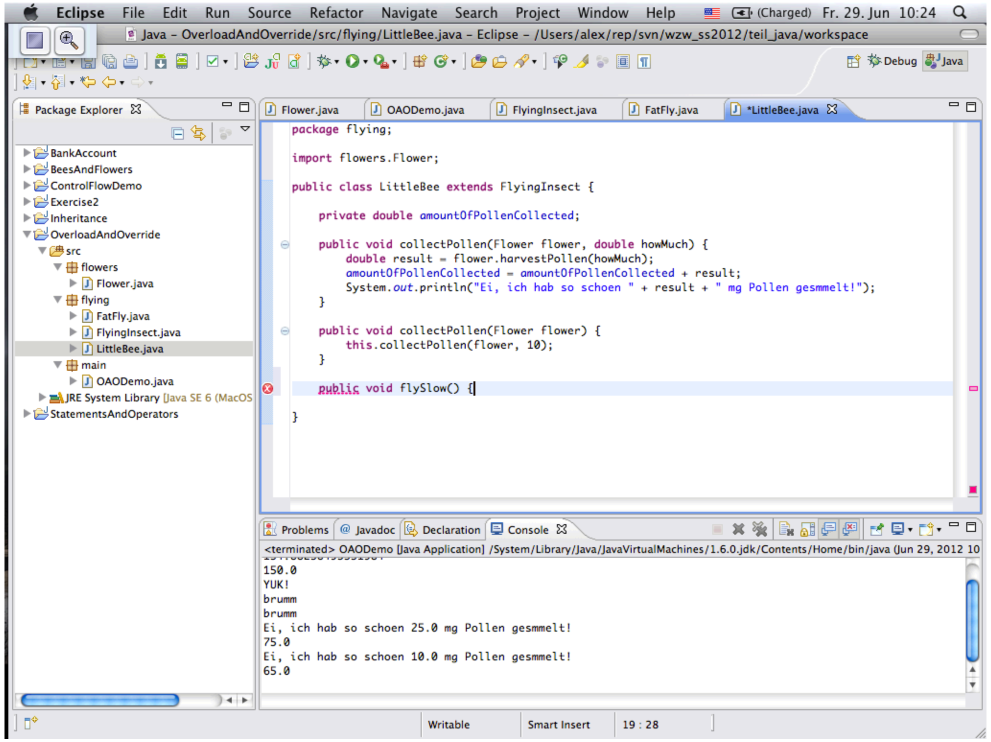

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 29. Jun 10:24
Flower.java
OAODemo.java
FlyingInsect.java
FatFly.java
*LittleBee.java

Java - OverloadAndOverride/src/flying/LittleBee.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package flying;

import flowers.Flower;

public class LittleBee extends FlyingInsect {

    private double amountOfPollenCollected;

    public void collectPollen(Flower flower, double howMuch) {
        double result = flower.harvestPollen(howMuch);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

    public void collectPollen(Flower flower) {
        this.collectPollen(flower, 10);
    }

    

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
```

Writable | Smart Insert | 19 : 5

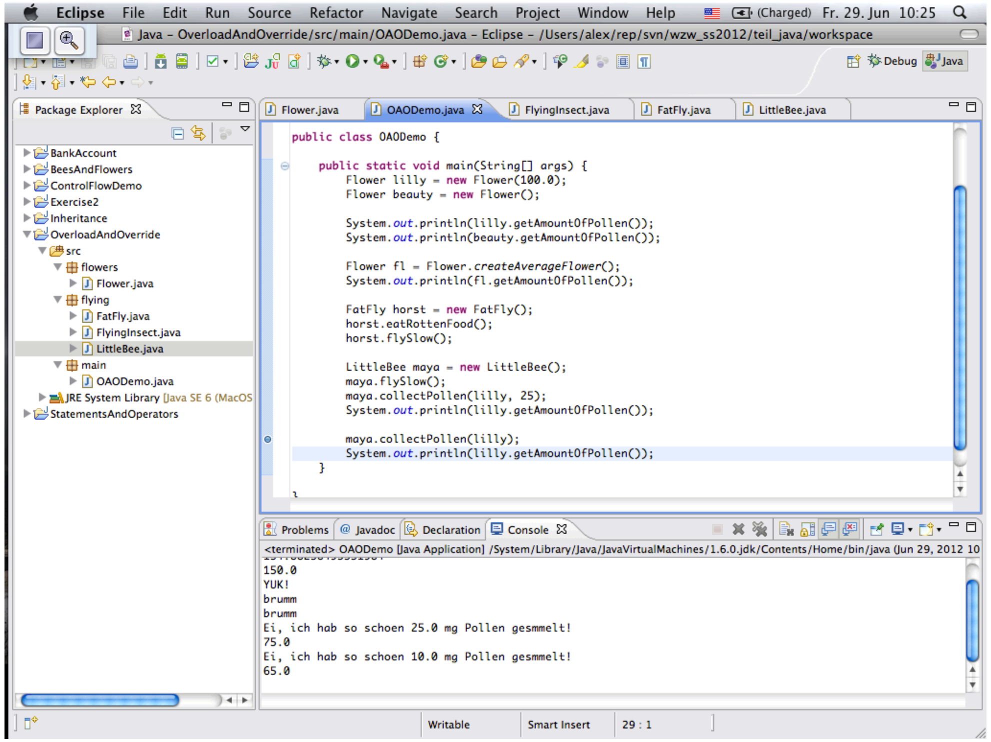

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 29. Jun 10:24
Flower.java
OAODemo.java
FlyingInsect.java
FatFly.java
*LittleBee.java

Java - OverloadAndOverride/src/flying/LittleBee.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
package flying;

import flowers.Flower;

public class LittleBee extends FlyingInsect {

    private double amountOfPollenCollected;

    public void collectPollen(Flower flower, double howMuch) {
        double result = flower.harvestPollen(howMuch);
        amountOfPollenCollected = amountOfPollenCollected + result;
        System.out.println("Ei, ich hab so schoen " + result + " mg Pollen gesmmelt!");
    }

    public void collectPollen(Flower flower) {
        this.collectPollen(flower, 10);
    }

    public void flySlow() {

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
```

Writable | Smart Insert | 19 : 28

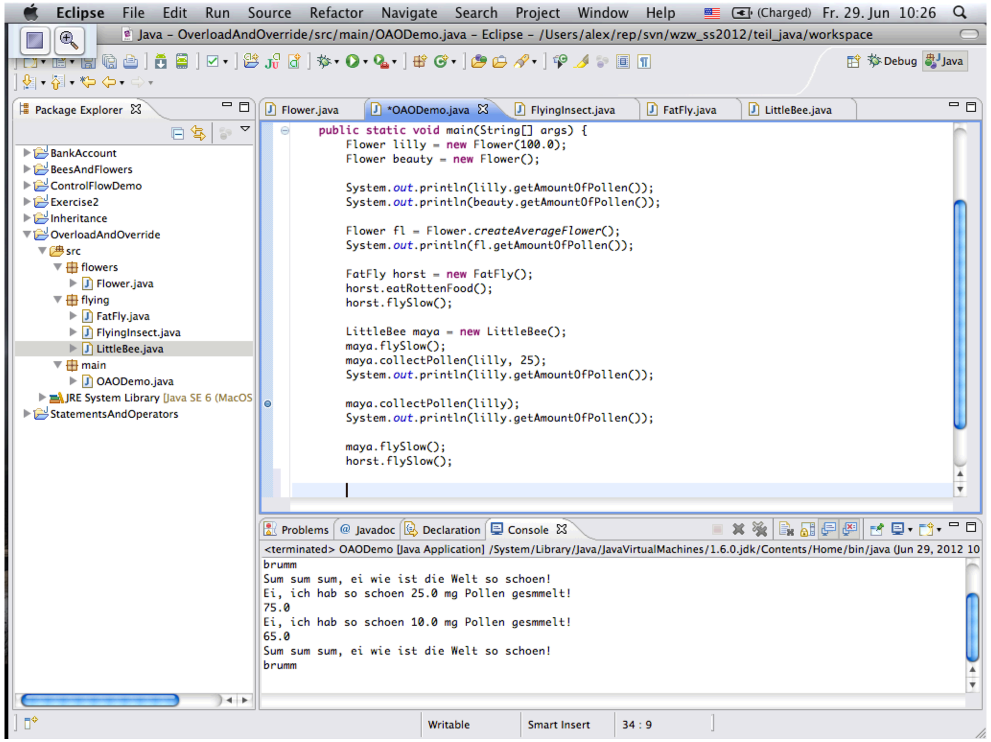

Eclipse File Edit Run Source Refactor Navigate Search Project Window Help (Charged) Fr. 29. Jun 10:25
Flower.java
OAODemo.java
FlyingInsect.java
FatFly.java
LittleBee.java

Java - OverloadAndOverride/src/main/OAODemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

```java
public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
        System.out.println(lilly.getAmountOfPollen());

        maya.collectPollen(lilly);
        System.out.println(lilly.getAmountOfPollen());
    }
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
150.0
YUK!
brumm
brumm
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
```

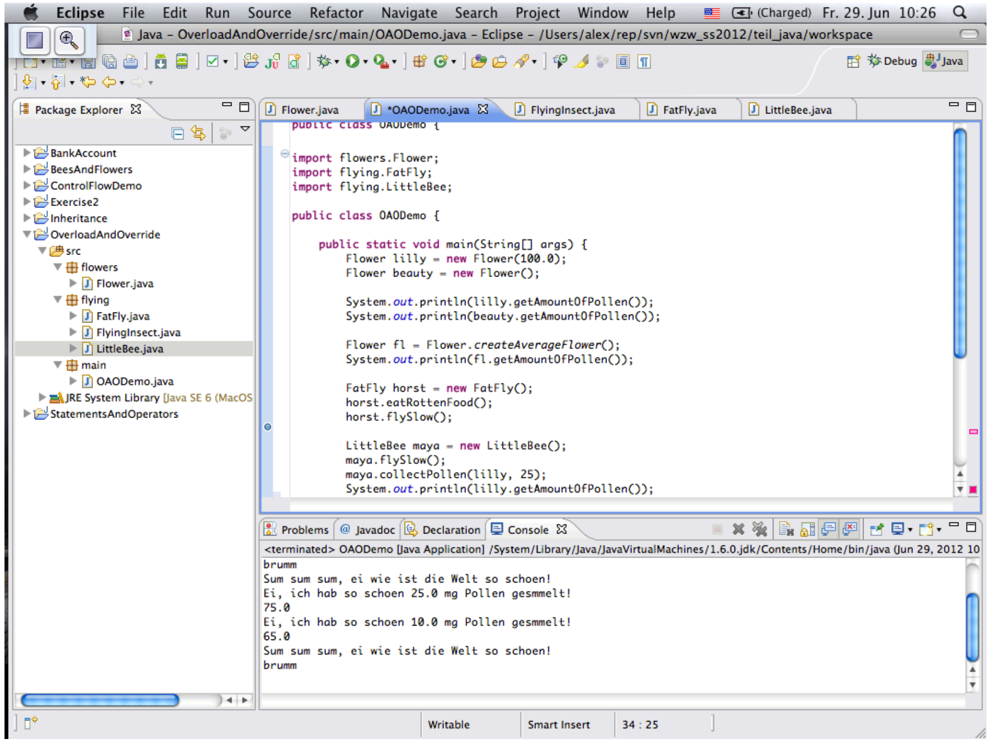
Writable | Smart Insert | 29 : 1



```java
    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
        System.out.println(lilly.getAmountOfPollen());

        maya.collectPollen(lilly);
        System.out.println(lilly.getAmountOfPollen());

        maya.flySlow();
        horst.flySlow();

        
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
brumm
Sum sum sum, ei wie ist die Welt so schoen!
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
Sum sum sum, ei wie ist die Welt so schoen!
brumm
```

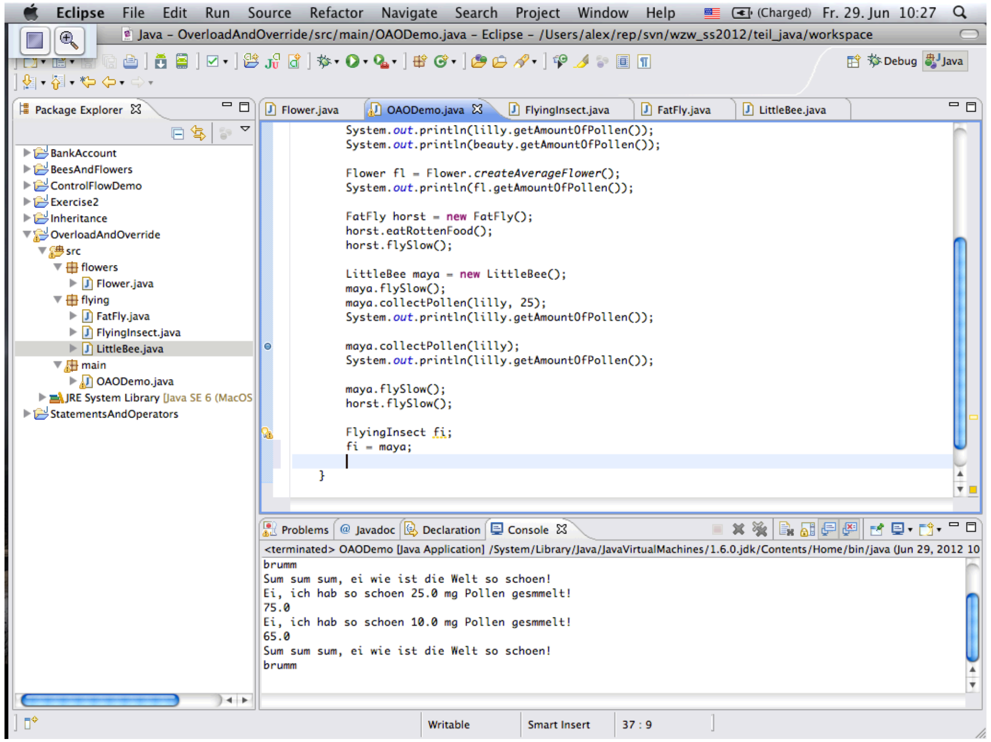
Writable | Smart Insert | 34 : 9



```java
public class OAODemo {

import flowers.Flower;
import flying.FatFly;
import flying.LittleBee;

public class OAODemo {

    public static void main(String[] args) {
        Flower lilly = new Flower(100.0);
        Flower beauty = new Flower();

        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
        System.out.println(lilly.getAmountOfPollen());
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
brumm
Sum sum sum, ei wie ist die Welt so schoen!
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
Sum sum sum, ei wie ist die Welt so schoen!
brumm
```

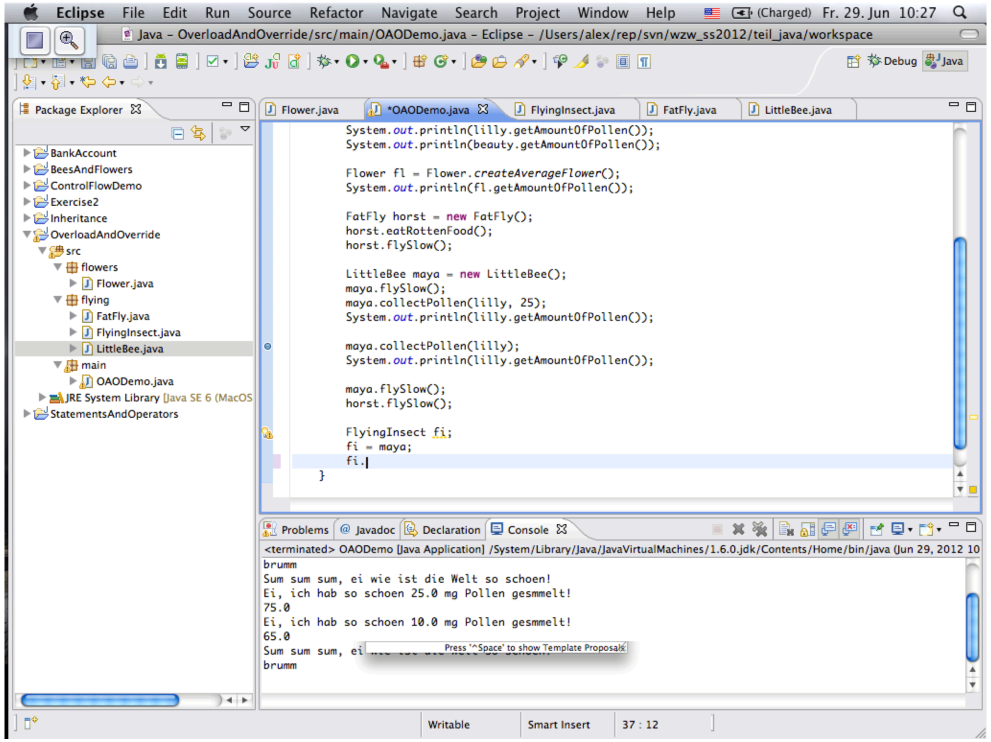
Writable | Smart Insert | 34 : 25



```java
        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
        System.out.println(lilly.getAmountOfPollen());

        maya.collectPollen(lilly);
        System.out.println(lilly.getAmountOfPollen());

        maya.flySlow();
        horst.flySlow();

        FlyingInsect fi;
        fi = maya;
        
    }
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
brumm
Sum sum sum, ei wie ist die Welt so schoen!
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
Sum sum sum, ei wie ist die Welt so schoen!
brumm
```

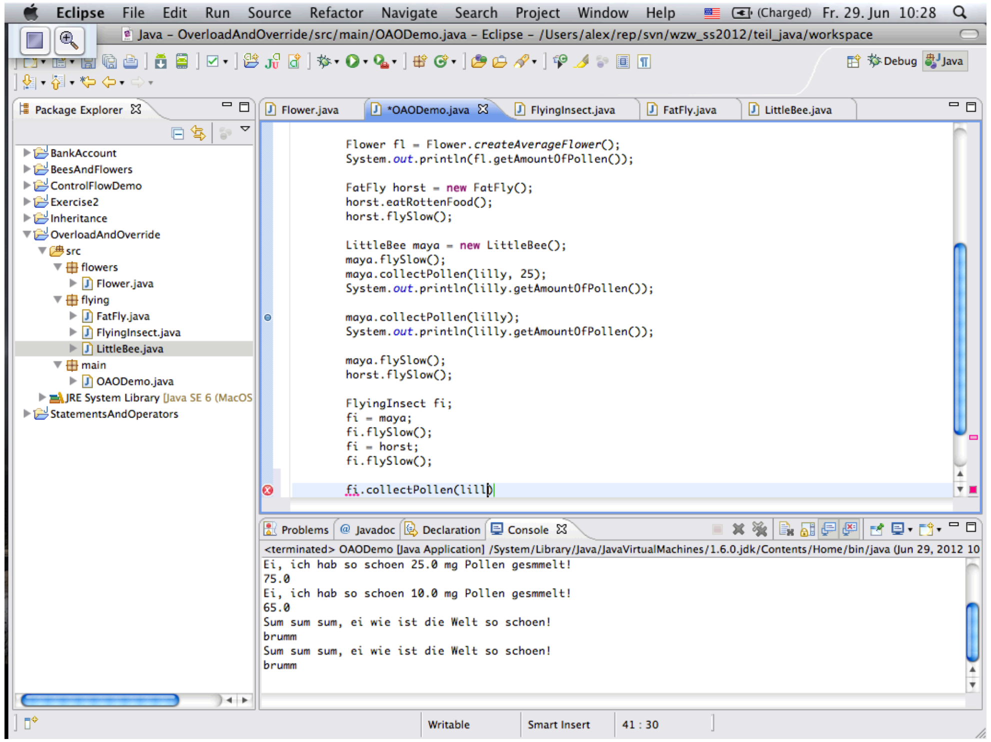
Writable | Smart Insert | 37 : 9



```java
        System.out.println(lilly.getAmountOfPollen());
        System.out.println(beauty.getAmountOfPollen());

        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
        System.out.println(lilly.getAmountOfPollen());

        maya.collectPollen(lilly);
        System.out.println(lilly.getAmountOfPollen());

        maya.flySlow();
        horst.flySlow();

        FlyingInsect fi;
        fi = maya;
        fi.
    }
```

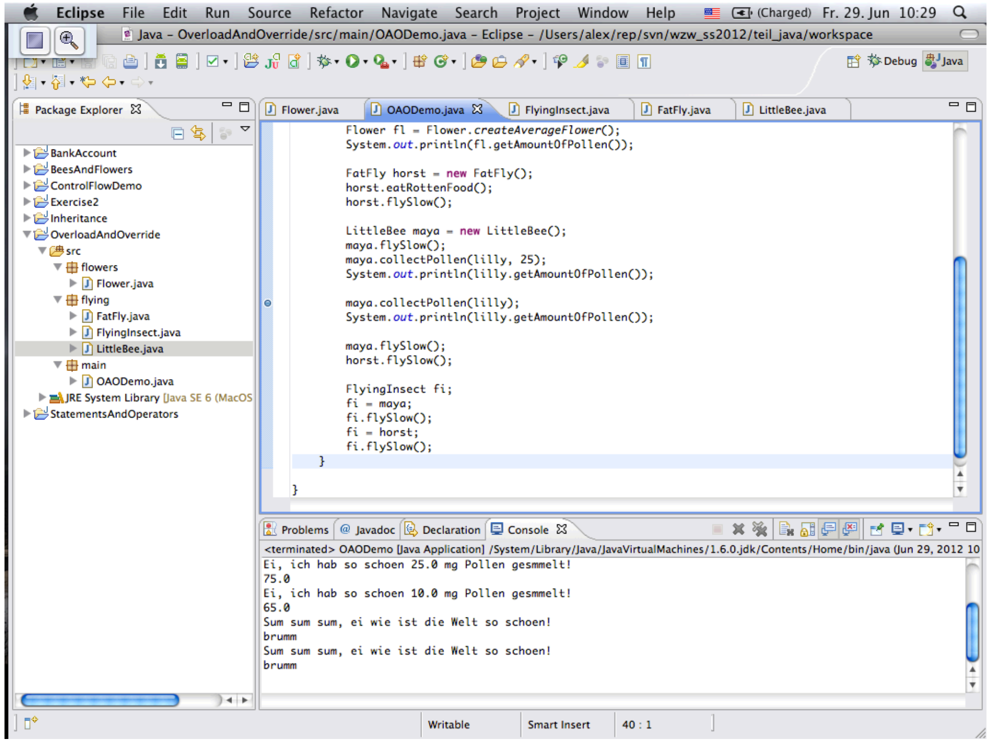
```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
brumm
Sum sum sum, ei wie ist die Welt so schoen!
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
Sum sum sum, ei
brumm
```

Writable | Smart Insert | 37 : 12



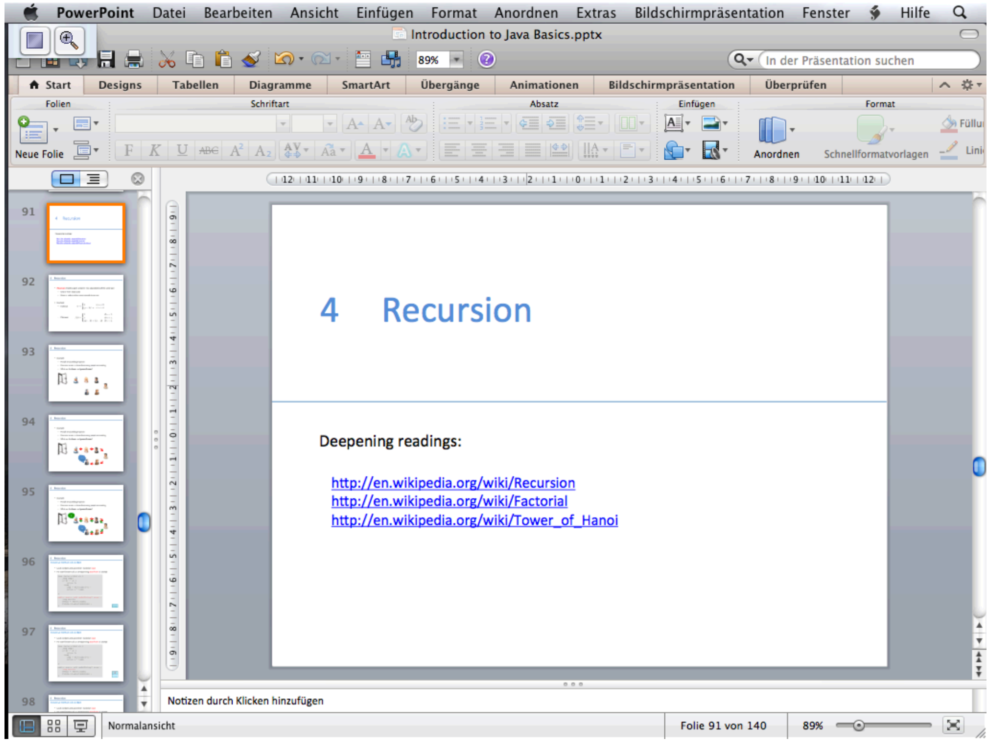
```
        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
        System.out.println(lilly.getAmountOfPollen());

        maya.collectPollen(lilly);
        System.out.println(lilly.getAmountOfPollen());

        maya.flySlow();
        horst.flySlow();

        FlyingInsect fi;
        fi = maya;
        fi.flySlow();
        fi = horst;
        fi.flySlow();

        fi.collectPollen(lill)
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
Sum sum sum, ei wie ist die Welt so schoen!
brumm
Sum sum sum, ei wie ist die Welt so schoen!
brumm
```



```
        Flower fl = Flower.createAverageFlower();
        System.out.println(fl.getAmountOfPollen());

        FatFly horst = new FatFly();
        horst.eatRottenFood();
        horst.flySlow();

        LittleBee maya = new LittleBee();
        maya.flySlow();
        maya.collectPollen(lilly, 25);
        System.out.println(lilly.getAmountOfPollen());

        maya.collectPollen(lilly);
        System.out.println(lilly.getAmountOfPollen());

        maya.flySlow();
        horst.flySlow();

        FlyingInsect fi;
        fi = maya;
        fi.flySlow();
        fi = horst;
        fi.flySlow();
    }

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
Sum sum sum, ei wie ist die Welt so schoen!
brumm
Sum sum sum, ei wie ist die Welt so schoen!
brumm
```

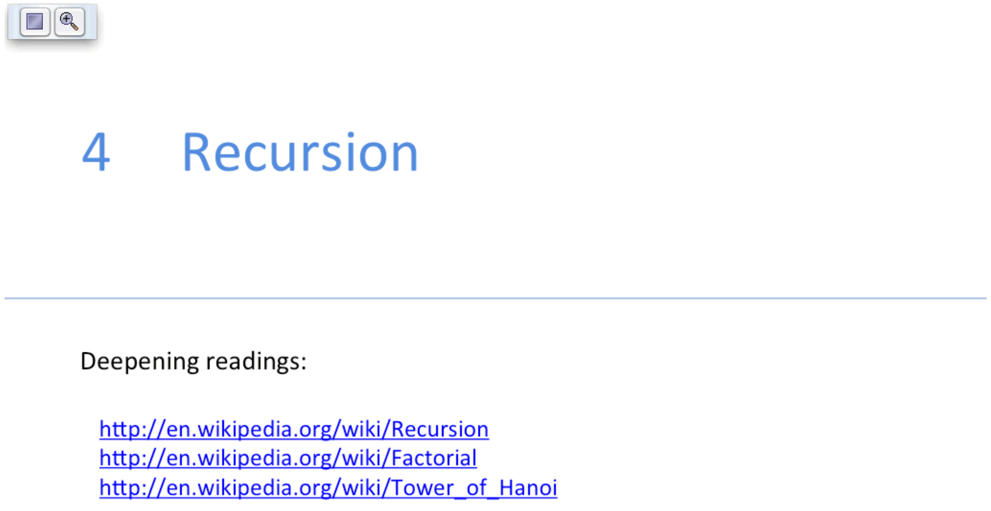


PowerPoint Datei Bearbeiten Ansicht Einfügen Format Anordnen Extras Bildschirmpräsentation Fenster Hilfe
Introduction to Java Basics.pptx
89%
In der Präsentation suchen
Start Designs Tabellen Diagramme SmartArt Übergänge Animationen Bildschirmpräsentation Überprüfen
Folien Schriftart Absatz Einfügen Format
Neue Folie
Anordnen
Schnellformatvorlagen
91
92
93
94
95
96
97
98
Notizen durch Klicken hinzufügen
Normalansicht
Folie 91 von 140
89%

4 Recursion

Deepening readings:

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



4 Recursion

Deepening readings:

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

ecursion

- **Recursion**: Divide a given problem into subproblems of the same type
  - One or more base cases
  - Rules to reduce other cases towards base case
- Example:
  - Factorial

$$n! = \begin{cases} 1 & \text{if } n = 0, \\ (n-1)! \cdot n & \text{if } n > 0. \end{cases}$$

  - Fibonacci

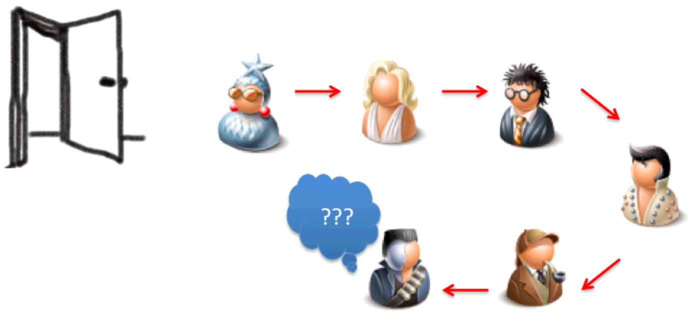
$$f(n) = \begin{cases} 1 & \text{if } n = 1, \\ 1 & \text{if } n = 2, \\ f(n-1) + f(n-2) & \text{if } n > 2. \end{cases}$$

ecursion

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

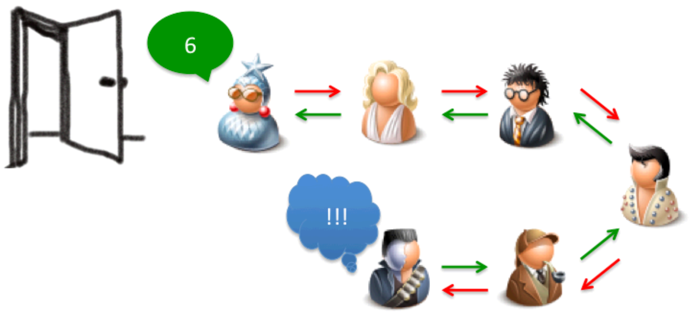
ecursion

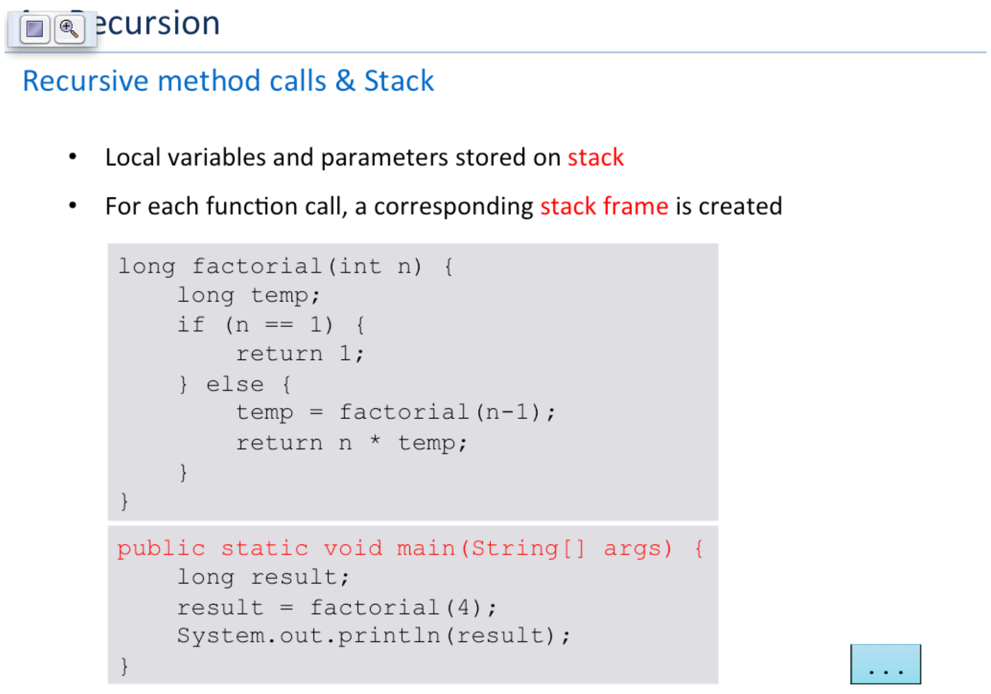
- Example:
  - People are standing in queue
  - Doorman wants to know how many people are waiting
  - What are the **base-** and **general cases**?



ecursion

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





ecursion

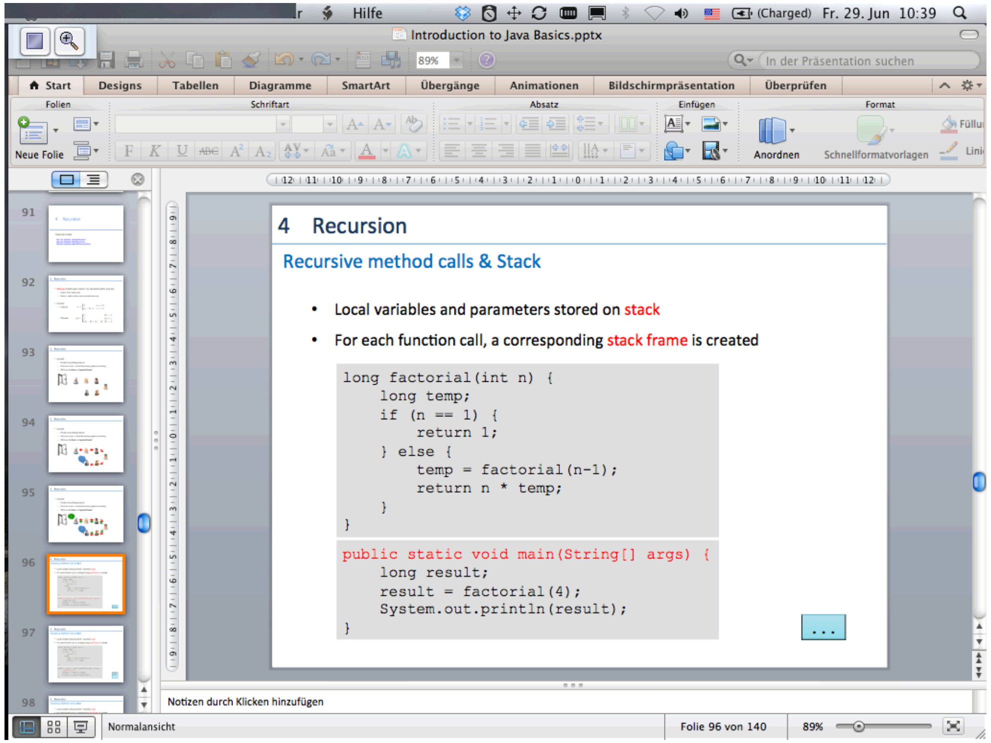
Recursive method calls & Stack

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

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

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

. . .



4 Recursion

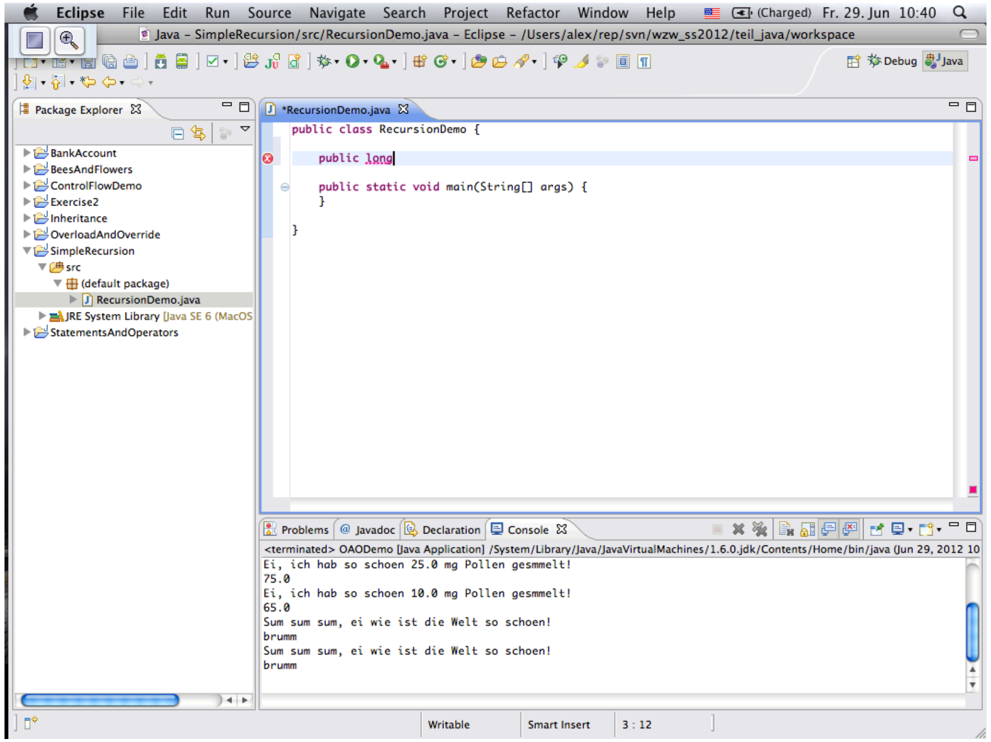
Recursive method calls & Stack

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

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

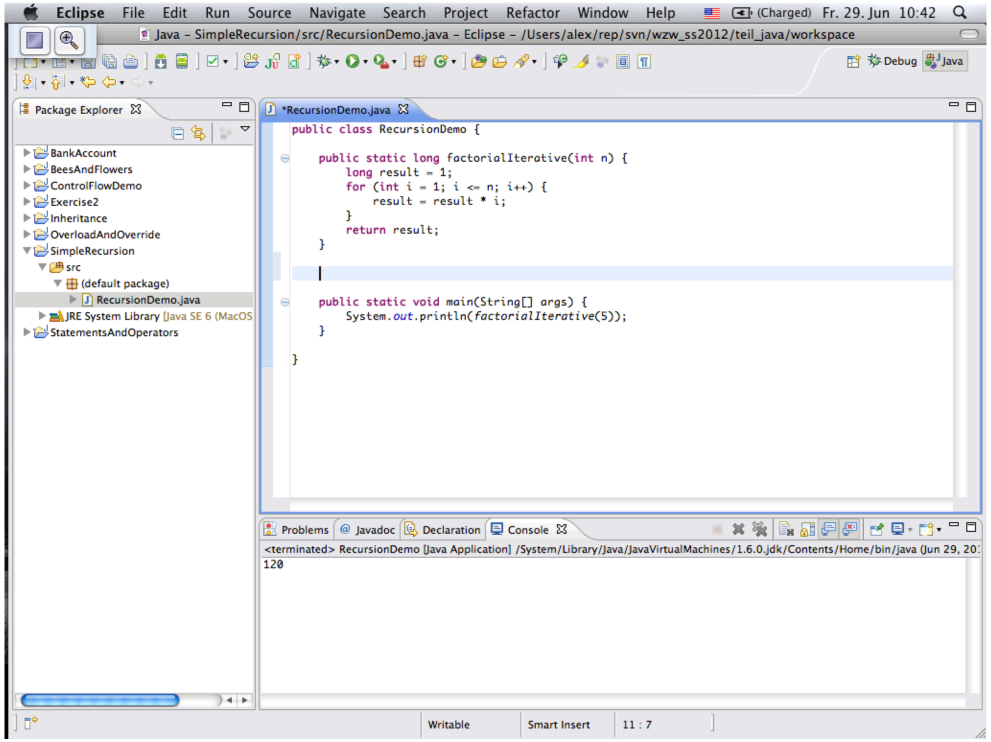
```
public static void main(String[] args) {
    long result;
    result = factorial(4);
    System.out.println(result);
}
```

. . .



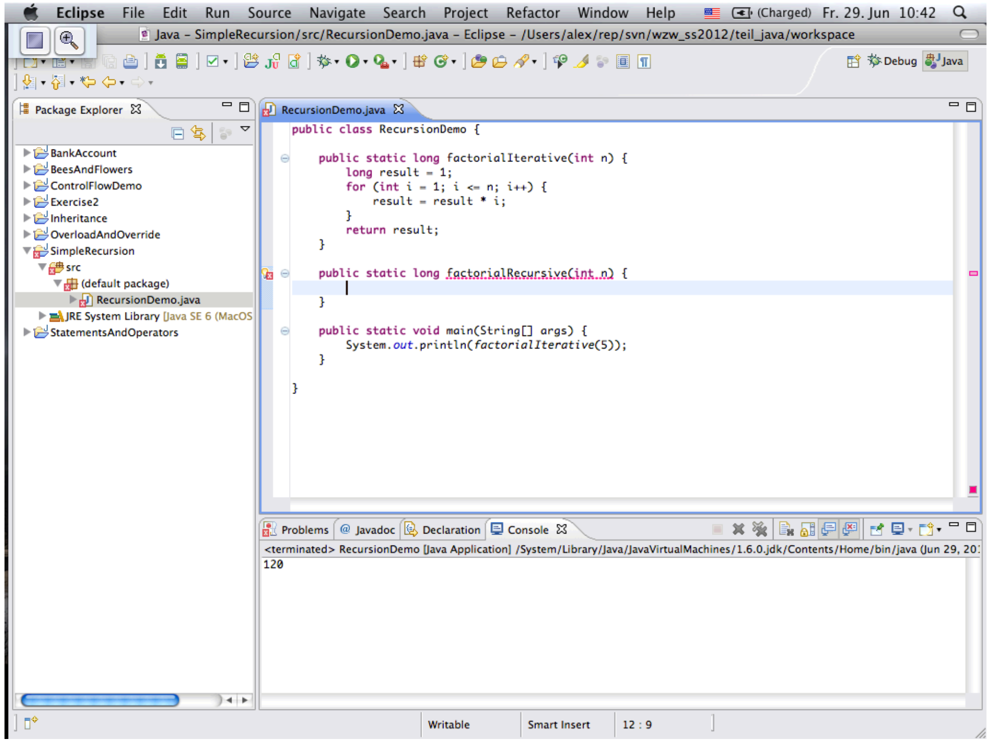
```
public class RecursionDemo {

    public long

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

}
```

```
<terminated> OAODemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10
Ei, ich hab so schoen 25.0 mg Pollen gesmmelt!
75.0
Ei, ich hab so schoen 10.0 mg Pollen gesmmelt!
65.0
Sum sum sum, ei wie ist die Welt so schoen!
brumm
Sum sum sum, ei wie ist die Welt so schoen!
brumm
```



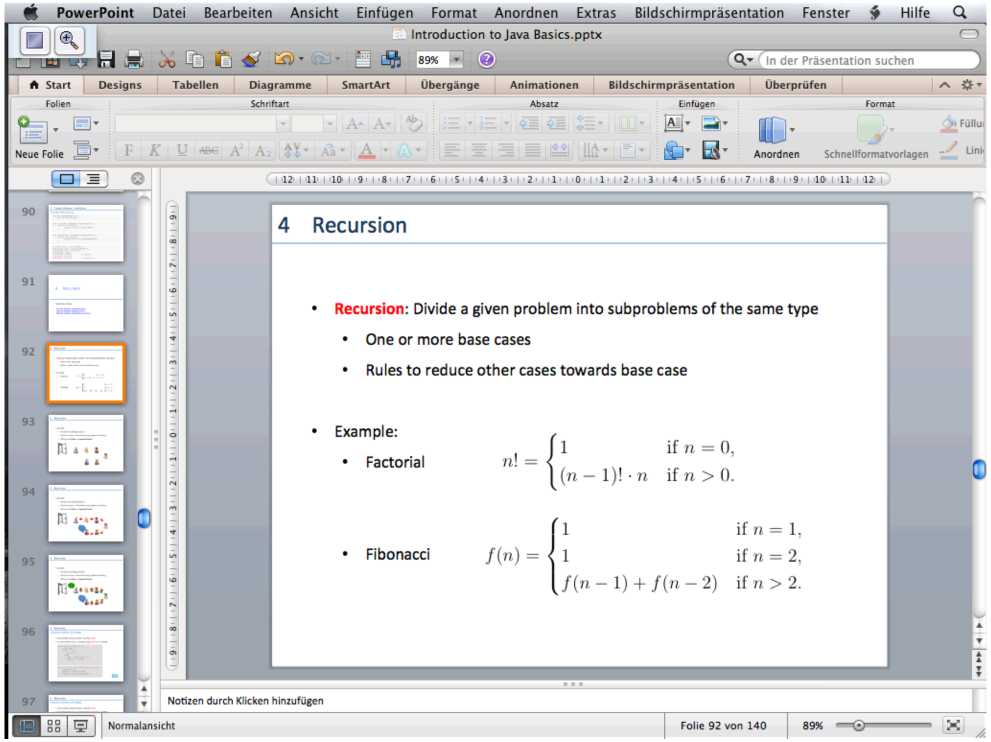
```
public class RecursionDemo {

    public static long factorialIterative(int n) {
        long result = 1;
        for (int i = 1; i <= n; i++) {
            result = result * i;
        }
        return result;
    }



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

}
```

```
<terminated> RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 20:
120
```



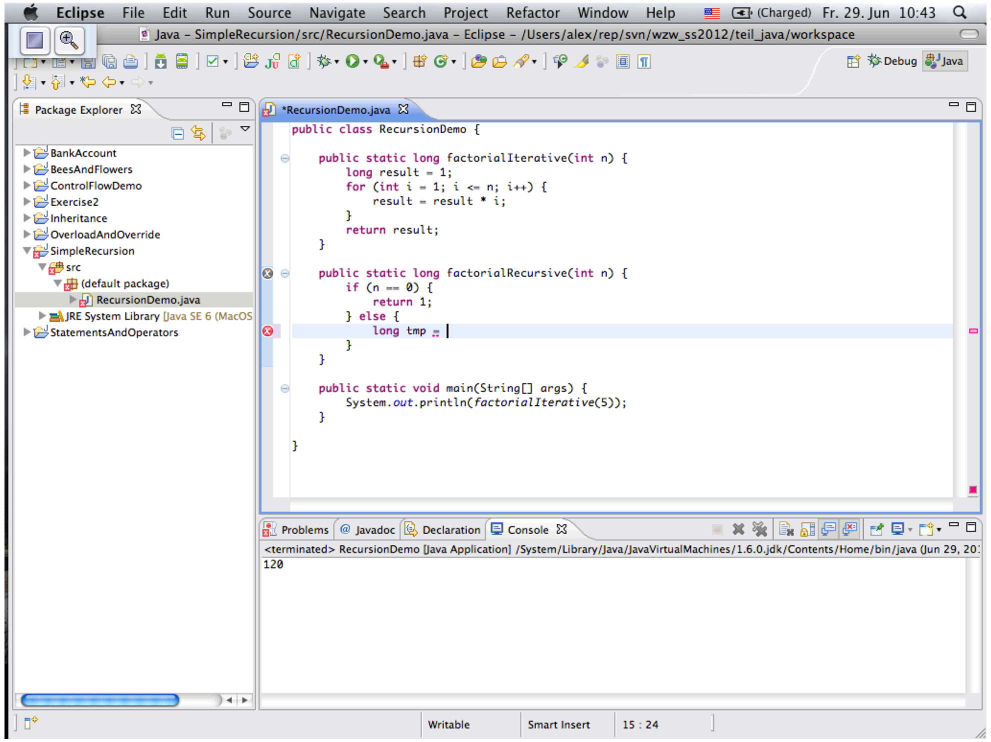
```
public class RecursionDemo {

    public static long factorialIterative(int n) {
        long result = 1;
        for (int i = 1; i <= n; i++) {
            result = result * i;
        }
        return result;
    }

    public static long factorialRecursive(int n) {
        
    }

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

}
```

```
<terminated> RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 20
120
```



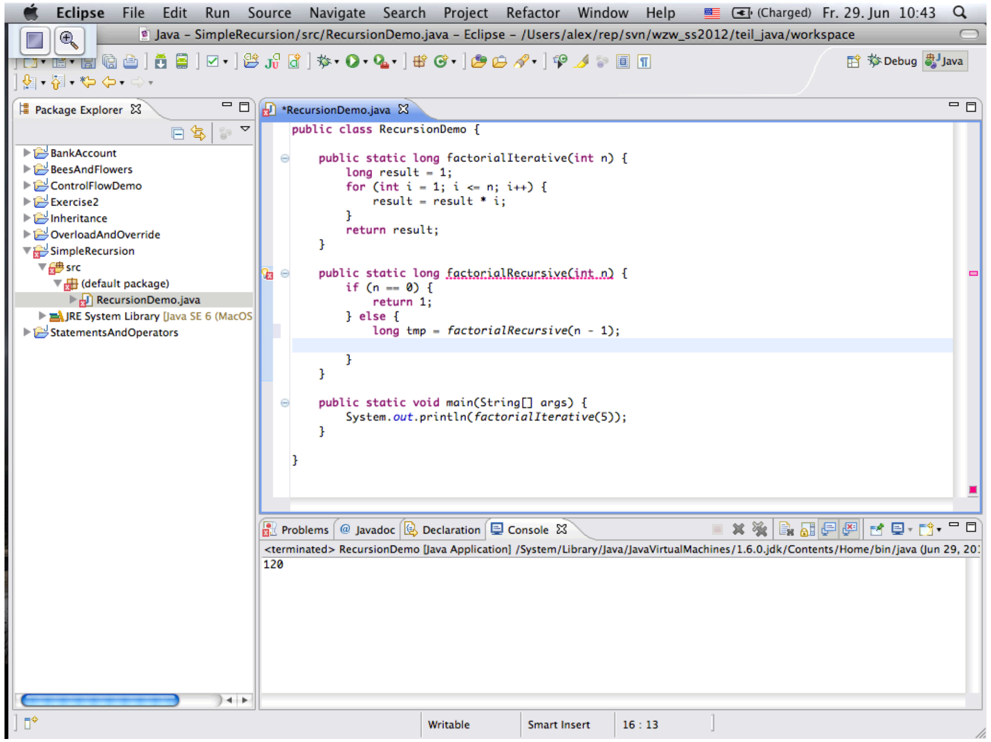
4 Recursion

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



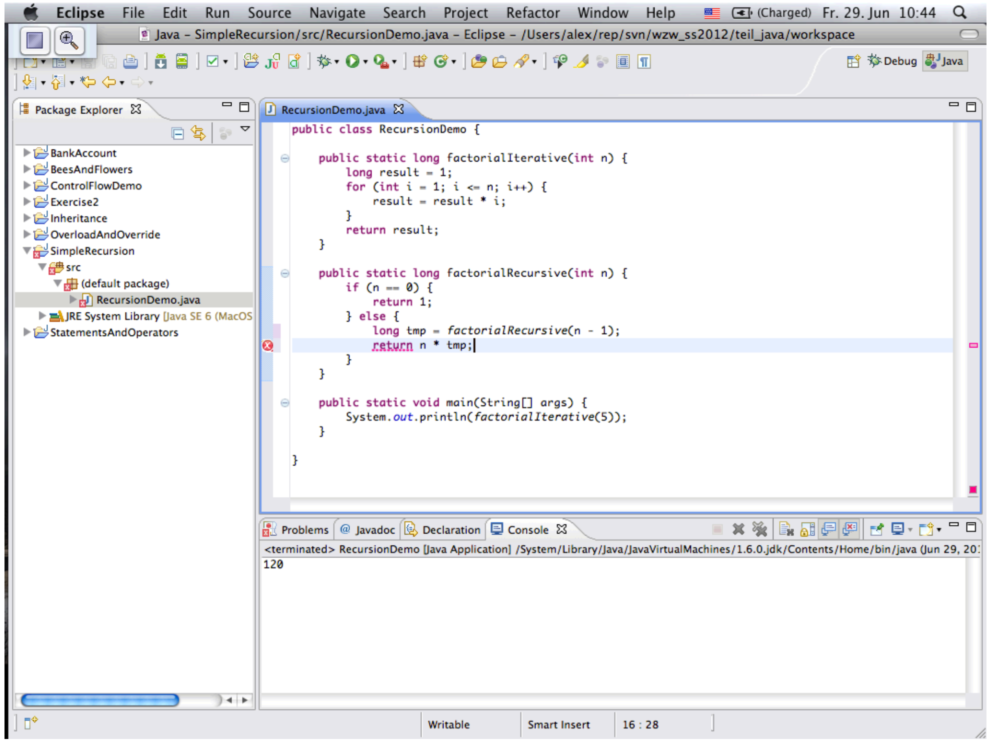
```
public class RecursionDemo {

    public static long factorialIterative(int n) {
        long result = 1;
        for (int i = 1; i <= n; i++) {
            result = result * i;
        }
        return result;
    }

    public static long factorialRecursive(int n) {
        if (n == 0) {
            return 1;
        } else {
            long tmp = 
        }
    }

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

}
```

```
<terminated> RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 20
120
```



```
public class RecursionDemo {

    public static long factorialIterative(int n) {
        long result = 1;
        for (int i = 1; i <= n; i++) {
            result = result * i;
        }
        return result;
    }

    public static long factorialRecursive(int n) {
        if (n == 0) {
            return 1;
        } else {
            long tmp = factorialRecursive(n - 1);

        }
    }

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

}
```

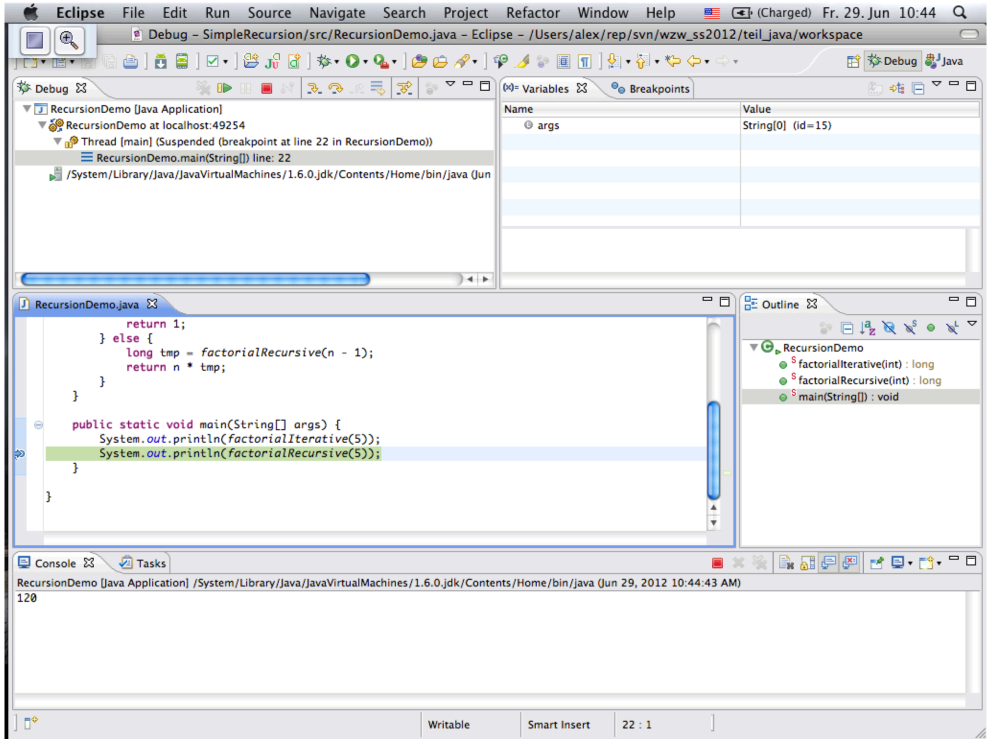
```
<terminated> RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 20
120
```


Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 29. Jun 10:44
Java – SimpleRecursion/src/RecursionDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Package Explorer
RecursionDemo.java
Problems
Javadoc
Declaration
Console
Writable
Smart Insert
16 : 28

- BankAccount
- BeesAndFlowers
- ControlFlowDemo
- Exercise2
- Inheritance
- OverloadAndOverride
- SimpleRecursion
  - src
    - (default package)
      - RecursionDemo.java
  - JRE System Library [Java SE 6 (MacOS
- StatementsAndOperators

```java
public class RecursionDemo {

    public static long factorialIterative(int n) {
        long result = 1;
        for (int i = 1; i <= n; i++) {
            result = result * i;
        }
        return result;
    }

    public static long factorialRecursive(int n) {
        if (n == 0) {
            return 1;
        } else {
            long tmp = factorialRecursive(n - 1);
            return n * tmp;
        }
    }

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

}
```

```
<terminated> RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 20:
120
```


Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 29. Jun 10:44
Debug – SimpleRecursion/src/RecursionDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Variables
Breakpoints
RecursionDemo.java
Outline
Console
Tasks
Writable
Smart Insert
22 : 1

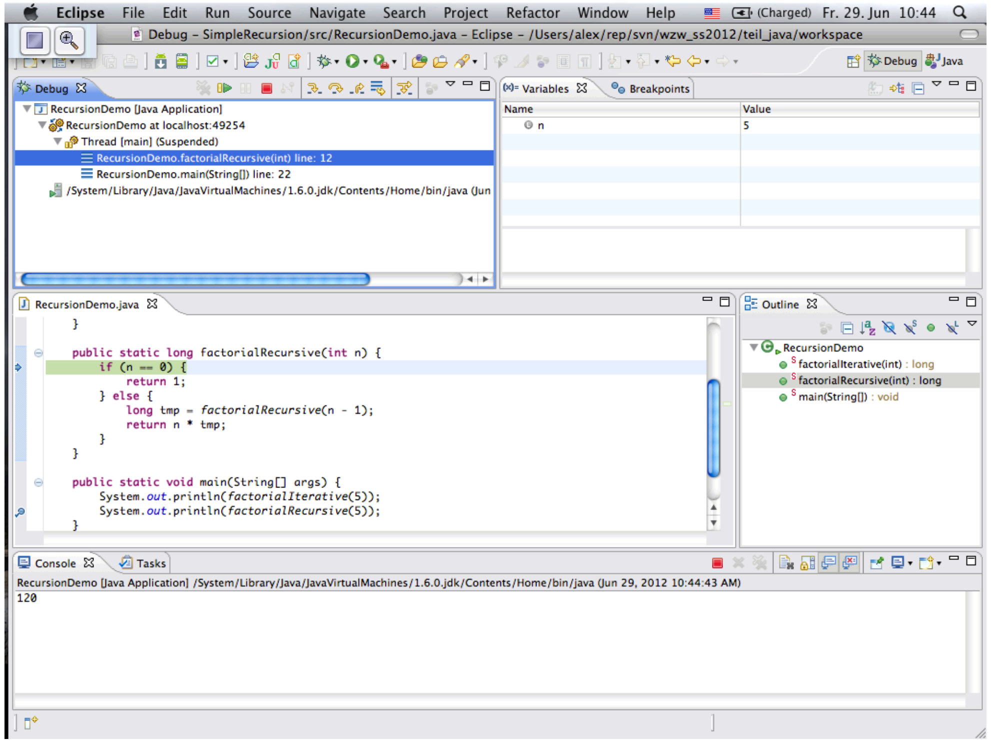
- RecursionDemo [Java Application]
  - RecursionDemo at localhost:49254
    - Thread [main] (Suspended (breakpoint at line 22 in RecursionDemo))
      - RecursionDemo.main(String[]) line: 22
  - /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun

Name	Value
args	String[0] (id=15)

```java
            return 1;
        } else {
            long tmp = factorialRecursive(n - 1);
            return n * tmp;
        }
    }

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

}
```

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

```
RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:44:43 AM)
120
```


Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 29. Jun 10:44
Debug – SimpleRecursion/src/RecursionDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Variables
Breakpoints
RecursionDemo.java
Outline
Console
Tasks

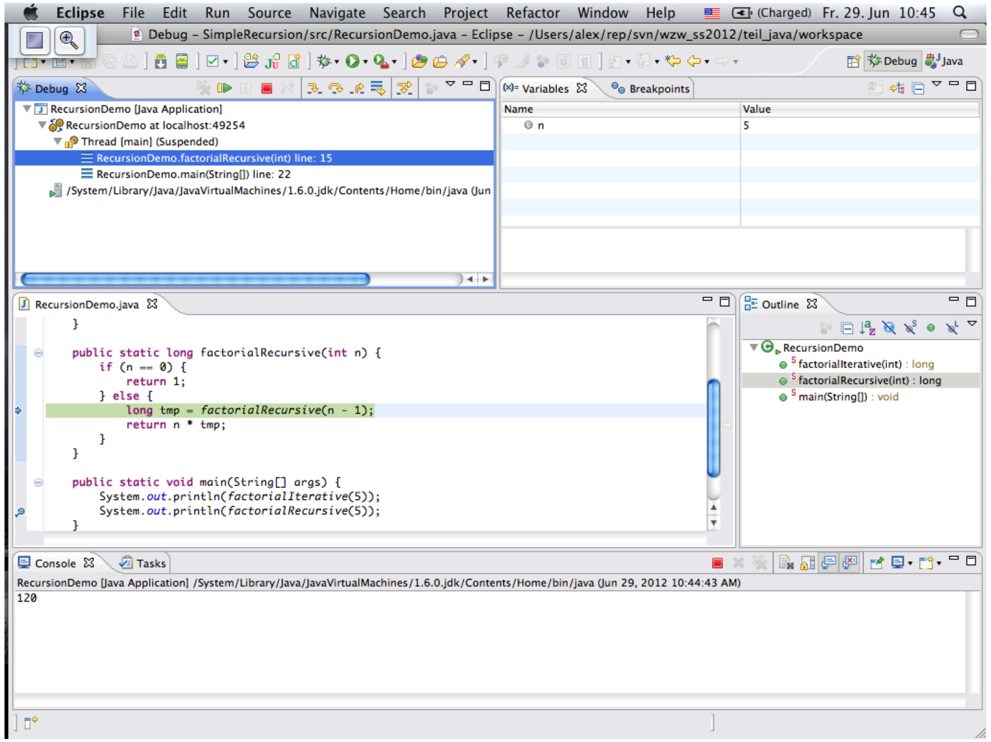
- RecursionDemo [Java Application]
  - RecursionDemo at localhost:49254
    - Thread [main] (Suspended)
      - RecursionDemo.factorialRecursive(int) line: 12
      - RecursionDemo.main(String[]) line: 22
  - /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun

Name	Value
n	5

```java
    }

    public static long factorialRecursive(int n) {
        if (n == 0) {
            return 1;
        } else {
            long tmp = factorialRecursive(n - 1);
            return n * tmp;
        }
    }

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

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

```
RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:44:43 AM)
120
```


Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 29. Jun 10:45
Debug – SimpleRecursion/src/RecursionDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Variables
Breakpoints
RecursionDemo.java
Outline
Console
Tasks

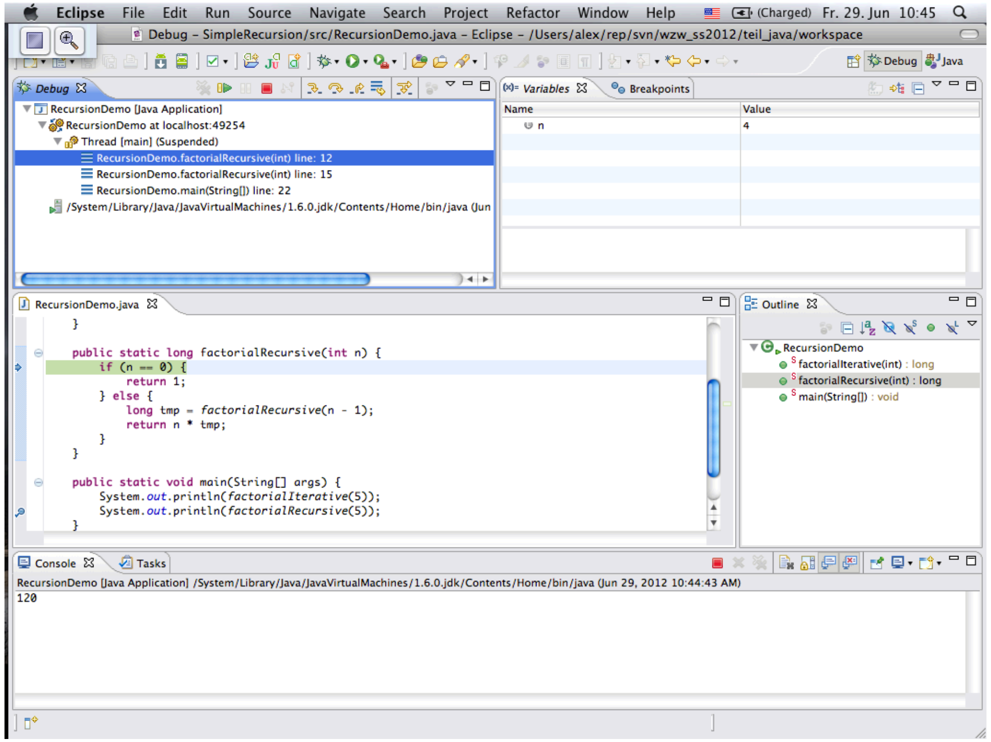
- RecursionDemo [Java Application]
  - RecursionDemo at localhost:49254
    - Thread [main] (Suspended)
      - RecursionDemo.factorialRecursive(int) line: 15
      - RecursionDemo.main(String[]) line: 22
  - /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun

Name	Value
n	5

```java
    }

    public static long factorialRecursive(int n) {
        if (n == 0) {
            return 1;
        } else {
            long tmp = factorialRecursive(n - 1);
            return n * tmp;
        }
    }

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

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

```
RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:44:43 AM)
120
```


Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 29. Jun 10:45
Debug – SimpleRecursion/src/RecursionDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Debug
Variables
Breakpoints
RecursionDemo.java
Outline
Console
Tasks

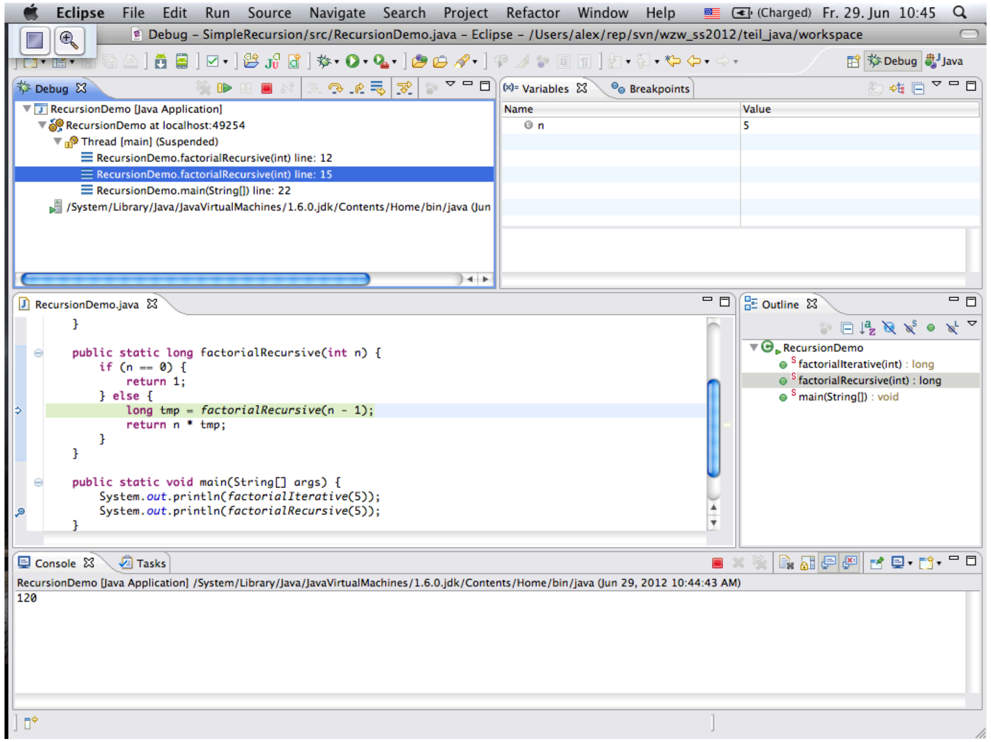
```
RecursionDemo [Java Application]
  RecursionDemo at localhost:49254
    Thread [main] (Suspended)
      RecursionDemo.factorialRecursive(int) line: 12
      RecursionDemo.factorialRecursive(int) line: 15
      RecursionDemo.main(String[]) line: 22
  /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
```

Name	Value
n	4

```java
    }

    public static long factorialRecursive(int n) {
        if (n == 0) {
            return 1;
        } else {
            long tmp = factorialRecursive(n - 1);
            return n * tmp;
        }
    }

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

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

```
RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:44:43 AM)
120
```


Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 29. Jun 10:45
Debug – SimpleRecursion/src/RecursionDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Debug
Variables
Breakpoints
RecursionDemo.java
Outline
Console
Tasks

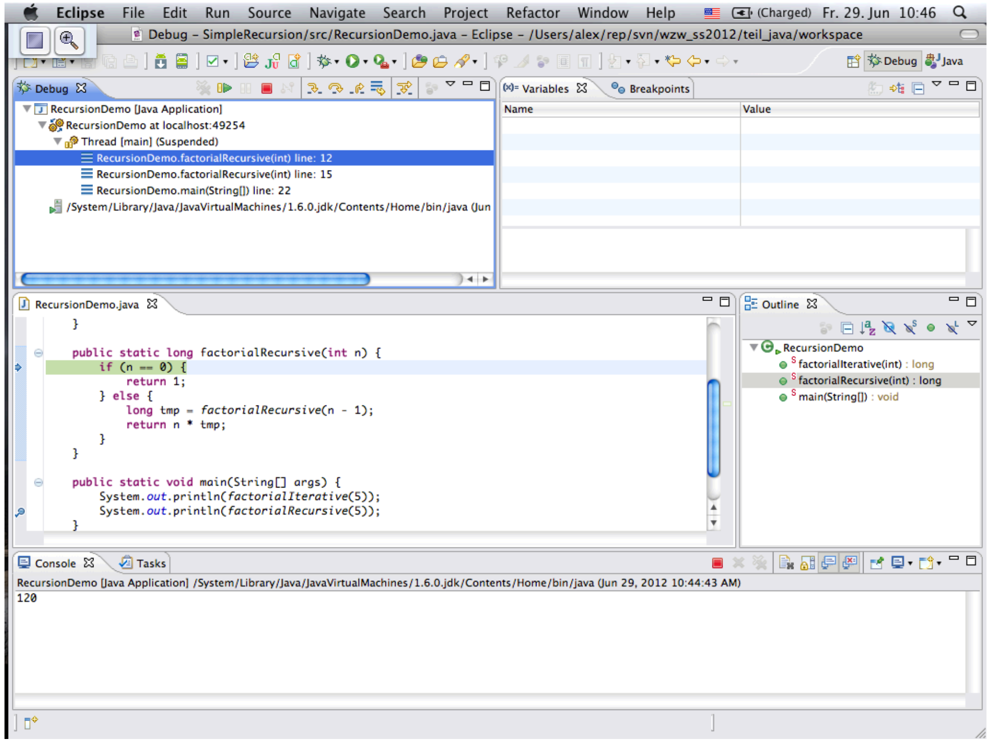
```
RecursionDemo [Java Application]
  RecursionDemo at localhost:49254
    Thread [main] (Suspended)
      RecursionDemo.factorialRecursive(int) line: 12
      RecursionDemo.factorialRecursive(int) line: 15
      RecursionDemo.main(String[]) line: 22
  /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
```

Name	Value
n	5

```java
    }

    public static long factorialRecursive(int n) {
        if (n == 0) {
            return 1;
        } else {
            long tmp = factorialRecursive(n - 1);
            return n * tmp;
        }
    }

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

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

```
RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:44:43 AM)
120
```


Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 29. Jun 10:46
Debug – SimpleRecursion/src/RecursionDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Debug
Variables
Breakpoints
RecursionDemo.java
Outline
Console
Tasks

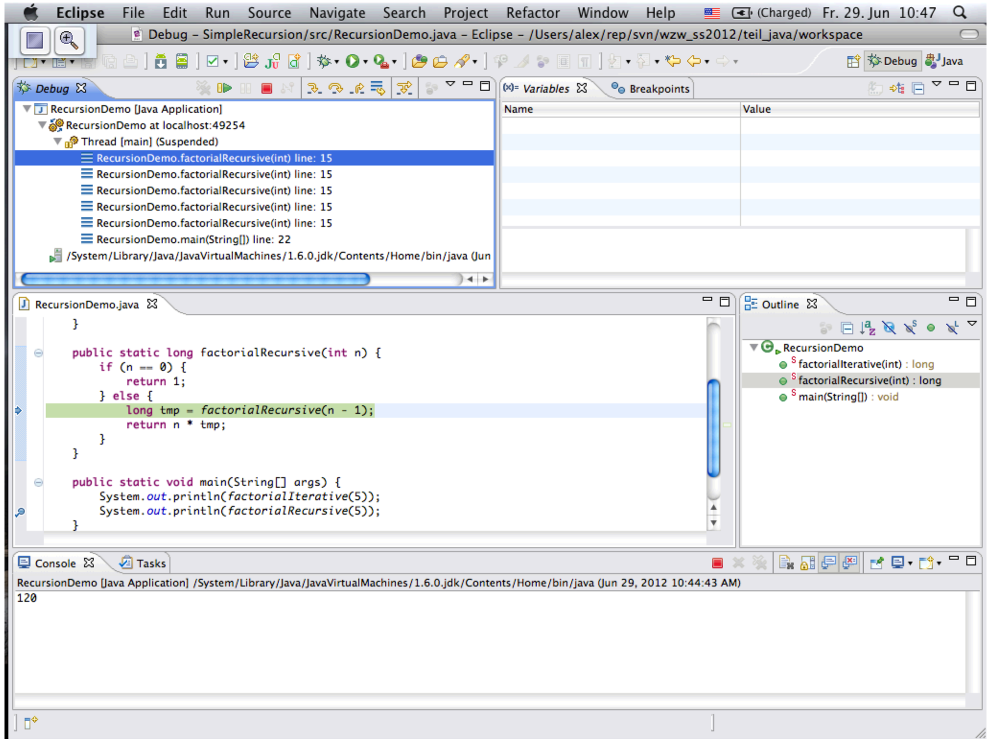
```
RecursionDemo [Java Application]
  RecursionDemo at localhost:49254
    Thread [main] (Suspended)
      RecursionDemo.factorialRecursive(int) line: 12
      RecursionDemo.factorialRecursive(int) line: 15
      RecursionDemo.main(String[]) line: 22
  /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
```

Name	Value

```java
    }

    public static long factorialRecursive(int n) {
        if (n == 0) {
            return 1;
        } else {
            long tmp = factorialRecursive(n - 1);
            return n * tmp;
        }
    }

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

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

```
RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:44:43 AM)
120
```


Eclipse File Edit Run Source Navigate Search Project Refactor Window Help (Charged) Fr. 29. Jun 10:47
Debug – SimpleRecursion/src/RecursionDemo.java – Eclipse – /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Debug
Java
Debug
Variables
Breakpoints
RecursionDemo.java
Outline
Console
Tasks

```
RecursionDemo [Java Application]
  RecursionDemo at localhost:49254
    Thread [main] (Suspended)
      RecursionDemo.factorialRecursive(int) line: 15
      RecursionDemo.factorialRecursive(int) line: 15
      RecursionDemo.factorialRecursive(int) line: 15
      RecursionDemo.factorialRecursive(int) line: 15
      RecursionDemo.factorialRecursive(int) line: 15
      RecursionDemo.main(String[]) line: 22
  /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun
```

Name	Value

```java
    }

    public static long factorialRecursive(int n) {
        if (n == 0) {
            return 1;
        } else {
            long tmp = factorialRecursive(n - 1);
            return n * tmp;
        }
    }

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

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

```
RecursionDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jun 29, 2012 10:44:43 AM)
120
```