Script generated by TTT

Title: groh: profile1 (12.06.2015)

Date: Fri Jun 12 09:21:01 CEST 2015

Duration: 86:19 min

Pages: 105

Primitive Typen: Numerische Typen

$\in \mathbb{Z}$ $\in \mathbb{Q}$ „$\approx \mathbb{R}$“

byte	short	int	long	float	double
8 bit	16 bit	32 bit	64 bit	32 bit	64 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]	[+/- ≈ $1.4*10^{-45}$, +/- ≈ $3.4*10^{38}$]	[+/- ≈ $4.9*10^{-324}$, +/- ≈ $1.8\ 10^{308}$]

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

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

54

Klassen und Objekte in Java

```
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;
       }
}
```

42

Interfaces

Idee: Spezifiziere nur *welche* Methoden eine Klasse haben muss, die das Interface implementieren will (Analogie: einzuhaltender Vertrag)

```
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
}
```

48

Variablen

- **Definition Variable** (informell): mit einem Bezeichner (Name) versehener Platzhalter für Werte eines bestimmten Typs. Variablen haben eine Adresse im Speicher.

- Variablen (bzw. ihre Werte) haben einen Typ:
  - primitiver Typ oder
  - Referenztyp

	(Typ-)Definition	Deklaration	Instantiierung	Manipulation	Test auf Gleichheit
Primitiv	(vordefiniert)	int a;	a = 117;	a = b + 42;	a == b;

Referenz	class Student {
	// Fields and
	// methods ...
	}
		Student heiner;	heiner = new Student();	heiner = horst;
				je nach Betrachtungsweise auch (Vorsicht!←→ genaue Definition! Siehe Folie 58ff.):
				heiner.age = 21;
				heiner.yawn();
					heiner.equals(horst);

Variablen

- **Definition Variable** (informell): mit einem Bezeichner (Name) versehener Platzhalter für Werte eines bestimmten Typs. Variablen haben eine Adresse im Speicher.

- Variablen (bzw. ihre Werte) haben einen Typ:
  - primitiver Typ oder
  - Referenztyp

	(Typ-)Definition	Deklaration	Instantiierung	Manipulation	Test auf Gleichheit
Primitiv	(vordefiniert)	int a;	a = 117;	a = b + 42;	a == b;

Referenz	class Student {
	// Fields and
	// methods ...
	}
		Student heiner;	heiner = new Student();	heiner = horst;
				je nach Betrachtungsweise auch (Vorsicht!←→ genaue Definition! Siehe Folie 58ff.):
				heiner.age = 21;
				heiner.yawn();
					heiner.equals(horst);

51

Variablen

- **Definition Variable** (informell): mit einem Bezeichner (Name) versehener Platzhalter für Werte eines bestimmten Typs. Variablen haben eine Adresse im Speicher.

- Variablen (bzw. ihre Werte) haben einen Typ:
  - primitiver Typ oder
  - Referenztyp

	(Typ-)Definition	Deklaration	Instantiierung	Manipulation	Test auf Gleichheit
Primitiv	(vordefiniert)	int a;	a = 117;	a = b + 42;	a == b;

Referenz	class Student {
	// Fields and
	// methods ...
	}
		Student heiner;	heiner = new Student();	heiner = horst;
				je nach Betrachtungsweise auch (Vorsicht!←→ genaue Definition! Siehe Folie 58ff.):
				heiner.age = 21;
				heiner.yawn();
					heiner.equals(horst);

51

Variablen

- **Definition Variable** (informell): mit einem Bezeichner (Name) versehener Platzhalter für Werte eines bestimmten Typs. Variablen haben eine Adresse im Speicher.

- Variablen (bzw. ihre Werte) haben einen Typ:
  - primitiver Typ oder
  - Referenztyp

	(Typ-)Definition	Deklaration	Instantiierung	Manipulation	Test auf Gleichheit
Primitiv	(vordefiniert)	int a;	a = 117;	a = b + 42;	a == b;

Referenz	class Student {
	// Fields and
	// methods ...
	}
		Student heiner;	heiner = new Student();	heiner = horst;
				je nach Betrachtungsweise auch (Vorsicht!←→ genaue Definition! Siehe Folie 58ff.):
				heiner.age = 21;
				heiner.yawn();
					heiner.equals(horst);

51

Variablen

- **Definition Variable** (informell): mit einem Bezeichner (Name) versehener Platzhalter für Werte eines bestimmten Typs. Variablen haben eine Adresse im Speicher.

- Variablen (bzw. ihre Werte) haben einen Typ:
  - primitiver Typ oder
  - Referenztyp

	(Typ-)Definition	Deklaration	Instantiierung	Manipulation	Test auf Gleicheit
Primitiv	(vordefiniert)	int a;	a = 117;	a = b + 42;	a == b;

Referenz	class Student {
	// Fields and
	// methods ...
	}
		Student heiner;	heiner = new Student();	heiner = horst;
				je nach Betrachtungsweise auch (Vorsicht!<--> genaue Definition! Siehe Folie 58ff.):
				heiner.age = 21;
				heiner.yawn();
					heiner.equals(horst);

Referenztypen

- Referenztyp-Variable „zeigt“ auf ein Objekt
- Typ der Variable ist die Klasse des Objekts

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

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

Vereinfachtes Speicher-Modell

Zellnr (Adresse)	Zellname (Variablenname)	Zellinhalt
...	...	...
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
...	...	...

Referenztypen

- Referenztyp-Variable „zeigt“ auf ein Objekt
- Typ der Variable ist die Klasse des Objekts

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

bike1.gear = 3;

bike1 = bike2;

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

Vereinfachtes Speicher-Modell

Zellnr (Adresse)	Zellname (Variablenname)	Zellinhalt
...	...	...
1149	bike1	<1405>
1150	bike1.cadence	0
1151	bike1.speed	0
1152	bike1.gear	3
...	...	...
1327	bike2	<1405>
...	...	...
1405	bike2.cadence	0
1406	bike2.speed	0
1407	bike2.gear	1
...	...	...

Referenztypen

- Referenztyp-Variable „zeigt“ auf ein Objekt
- Typ der Variable ist die Klasse des Objekts

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

bike1.gear = 3;

bike1 = bike2;

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

Vereinfachtes Speicher-Modell

Zellnr (Adresse)	Zellname (Variablenname)	Zellinhalt
...	...	...
1149	bike1	<1405>
1150	bike1.cadence	0
1151	bike1.speed	0
1152	bike1.gear	3
...	...	...
1327	bike2	<1405>
...	...	...
1405	bike2.cadence	0
1406	bike2.speed	0
1407	bike2.gear	1
...	...	...

Referenztypen

- Referenztyp-Variable „zeigt“ auf ein Objekt
- Typ der Variable ist die Klasse des Objekts

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

bike1.gear = 3;




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

Vereinfachtes Speicher-Modell

Zellnr (Adresse)	Zellname (Variablenname)	Zellinhalt
...	...	...
1149	bike1	<1150>
1150	bike1.cadence	0
1151	bike1.speed	0
1152	**bike1.gear**	**3**
...	...	...
1327	bike2	<1405>
...	...	...
1405	bike2.cadence	0
1406	bike2.speed	0
1407	bike2.gear	1
...	...	...

Referenztypen

- Referenztyp-Variable „zeigt“ auf ein Objekt
- Typ der Variable ist die Klasse des Objekts

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

bike1.gear = 3;




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

Vereinfachtes Speicher-Modell

Zellnr (Adresse)	Zellname (Variablenname)	Zellinhalt
...	...	...
1149	bike1	<1150>
1150	bike1.cadence	0
1151	bike1.speed	0
1152	**bike1.gear**	**3**
...	...	...
1327	bike2	<1405>
...	...	...
1405	bike2.cadence	0
1406	bike2.speed	0
1407	bike2.gear	1
...	...	...

59

Referenztypen

- Referenztyp-Variable „zeigt“ auf ein Objekt
- Typ der Variable ist die Klasse des Objekts

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

bike1.gear = 3;

bike1 = bike2;



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

Vereinfachtes Speicher-Modell

Zellnr (Adresse)	Zellname (Variablenname)	Zellinhalt
...	...	...
1149	bike1	<1405>
1150	bike1.cadence	0
1151	bike1.speed	0
1152	bike1.gear	3
...	...	...
1327	bike2	<1405>
...	...	...
1405	bike2.cadence	0
1406	bike2.speed	0
1407	bike2.gear	1
...	...	...

Referenztypen

- Referenztyp-Variable „zeigt“ auf ein Objekt
- Typ der Variable ist die Klasse des Objekts

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

bike1.gear = 3;

bike1 = bike2;



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

Vereinfachtes Speicher-Modell

Zellnr (Adresse)	Zellname (Variablenname)	Zellinhalt
...	...	...
1149	bike1	<1405>
1150	bike1.cadence	0
1151	bike1.speed	0
1152	bike1.gear	3
...	...	...
1327	bike2	<1405>
...	...	...
1405	bike2.cadence	0
1406	bike2.speed	0
1407	bike2.gear	1
...	...	...

60

Operatoren

Zuweisungs (Assignment) Operator

```
=        a = b+1;      boolean ccc = (a==b);      bike2 = bike1.copy();
```


Ausdrücke (Expressions)

Expression: Legale Kombination aus Konstanten, Variablen und Operatoren (inklusive Methodenaufrufe und Objekterzeugung mit `new`)

- Jede Expression hat einen (evaluiert zu einem) Wert der einen bestimmten Typ hat

Beispiel:

gegeben:

```
int a = 73;
Bicycle bike;
```

Expression	evaluiert zu	Typ
48	48	int
2.0 / 3.0	0.6666666666...6	double
true && false	false	boolean
15 / 8	1	int
(17 + (3 * 9)) % 3	2	int
a + 1	74	int
a = 9	9	int
new Bicycle()	(Referenz auf Bicycle Objekt)	Bicycle
bike = new Bicycle()	(Referenz auf Bicycle Objekt)	Bicycle
new double[20]	(Referenz auf Array von double)	double[]
bike.cadence	0	int

Ausdrücke (Expressions)

Expression: Legale Kombination aus Konstanten, Variablen und Operatoren (inklusive Methodenaufrufe und Objekterzeugung mit `new`)

- Jede Expression hat einen (evaluiert zu einem) Wert der einen bestimmten Typ hat

Beispiel:

gegeben:

```
int a = 73;
Bicycle bike;
```

Expression	evaluiert zu	Typ
48	48	int
2.0 / 3.0	0.6666666666...6	double
true && false	false	boolean
15 / 8	1	int
(17 + (3 * 9)) % 3	2	int
a + 1	74	int
a = 9	9	int
new Bicycle()	(Referenz auf Bicycle Objekt)	Bicycle
bike = new Bicycle()	(Referenz auf Bicycle Objekt)	Bicycle
new double[20]	(Referenz auf Array von double)	double[]
bike.cadence	0	int

77

Ausdrücke (Expressions)

Expression: Legale Kombination aus Konstanten, Variablen und Operatoren (inklusive Methodenaufrufe und Objekterzeugung mit `new`)

- Jede Expression hat einen (evaluiert zu einem) Wert der einen bestimmten Typ hat

Beispiel:

gegeben:

```
int a = 73;
Bicycle bike;
```

Expression	evaluiert zu	Typ
48	48	int
2.0 / 3.0	0.6666666666...6	double
true && false	false	boolean
15 / 8	1	int
(17 + (3 * 9)) % 3	2	int
a + 1	74	int
a = 9	9	int
new Bicycle()	(Referenz auf Bicycle Objekt)	Bicycle
bike = new Bicycle()	(Referenz auf Bicycle Objekt)	Bicycle
new double[20]	(Referenz auf Array von double)	double[]
bike.cadence	0	int

77

Expressions: Seiteneffekte

- Manche Expressions haben sogenannte Seiteneffekte (in den meisten Fällen ist dies der einzig wichtige Aspekt)

Beispiel:

gegeben:

```
int a = 48;
int b;
```

Expression	Wert	Seiteneffekt
`a = 84`	`84`	Wert 84 zu a zuweisen
`b = (a = 20)`	`20`	Wert 20 zu a und b zuweisen
`new Bicycle()`	(Referenz auf Bicycle Objekt)	Kreire und initialisiere eine neue Instanz (ein neues Objekt) der Klasse `Bicycle` im Speicher
`new double[20]`	(Referenz auf Array von `double`)	Kreire und initialisiere ein neues Array von 20 `double` Variablen im Speicher
`a++`	`48`	Wert 49 zu a zuweisen
`b = a++`	48	Werte 48 an b und 49 an a zuweisen
`++a`	`49`	Wert 49 zu a zuweisen
`b = ++a`	49	Werte 49 an b und 49 an a zuweisen

Expressions: Seiteneffekte

- Manche Expressions haben sogenannte Seiteneffekte (in den meisten Fällen ist dies der einzig wichtige Aspekt)

Beispiel:

gegeben:

```
int a = 48;
int b;
```

Expression	Wert	Seiteneffekt
`a = 84`	`84`	Wert 84 zu a zuweisen
`b = (a = 20)`	`20`	Wert 20 zu a und b zuweisen
`new Bicycle()`	(Referenz auf Bicycle Objekt)	Kreire und initialisiere eine neue Instanz (ein neues Objekt) der Klasse `Bicycle` im Speicher
`new double[20]`	(Referenz auf Array von `double`)	Kreire und initialisiere ein neues Array von 20 `double` Variablen im Speicher
`a++`	`48`	Wert 49 zu a zuweisen
`b = a++`	48	Werte 48 an b und 49 an a zuweisen
`++a`	`49`	Wert 49 zu a zuweisen
`b = ++a`	49	Werte 49 an b und 49 an a zuweisen

Statements

Statement: Komplette ausführbare Einheit. Endet mit „ ; “

Expression statements: (sind Expressions die mit ; abgeschlossen wurden)

- Zuweisungen `a = (17 + (3 * 9)) % 3;`
- Zuweisung unter Benutzung von ++ oder −− `a++;`
- Methodenaufrufe `someObject.methodOne();`
- Objekterzeugung `someObject = new SomeClass();`

Deklarationen `int a;` `SomeClass someObject;`

Blocks (nächste Folie)

Kontrollfluss-Statements (kommt gleich)

Statements

Statement: Komplette ausführbare Einheit. Endet mit „ ; “

Expression statements: (sind Expressions die mit ; abgeschlossen wurden)

- Zuweisungen `a = (17 + (3 * 9)) % 3;`
- Zuweisung unter Benutzung von ++ oder −− `a++;`
- Methodenaufrufe `someObject.methodOne();`
- Objekterzeugung `someObject = new SomeClass();`

Deklarationen `int a;` `SomeClass someObject;`

Blocks (nächste Folie)

Kontrollfluss-Statements (kommt gleich)

Statements

Statement: Komplette ausführbare Einheit. Endet mit „ ; “

Expression statements: (sind Expressions die mit ; abgeschlossen wurden)

- Zuweisungen `a = (17 + (3 * 9)) % 3;`
- Zuweisung unter Benutzung von ++ oder -- `a++;`
- Methodenaufrufe `someObject.methodOne();`
- Objekterzeugung `someObject = new SomeClass();`

Deklarationen `int a;` `SomeClass someObject;`

Blocks (nächste Folie)

Kontrollfluss-Statements (kommt gleich)

Kontrollfluss-Statements

Kontrollfluss-Statements erlauben es, von der sequentiellen Abfolge der Abarbeitung der Statements abzuweichen.

- Bedingte Verzweigungen (conditionals): `if, if else, switch`
- Schleifen (loops): `while, do while, for`
- Verzweigungen (branches): `break, continue, return`

Schleifen: `while` `do while`

`while`: Mache etwas, *solange* eine Bedingung gilt (zu `true` evaluiert)

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

⇨ Ausgabe: `#:1 #:2 #:3 #:4 #:5 #:6 #:7`

Schleifen: `for`

`for`: (Üblicherweise:) Mache etwas eine festgelegte Anzahl von Malen (Iterationen)

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

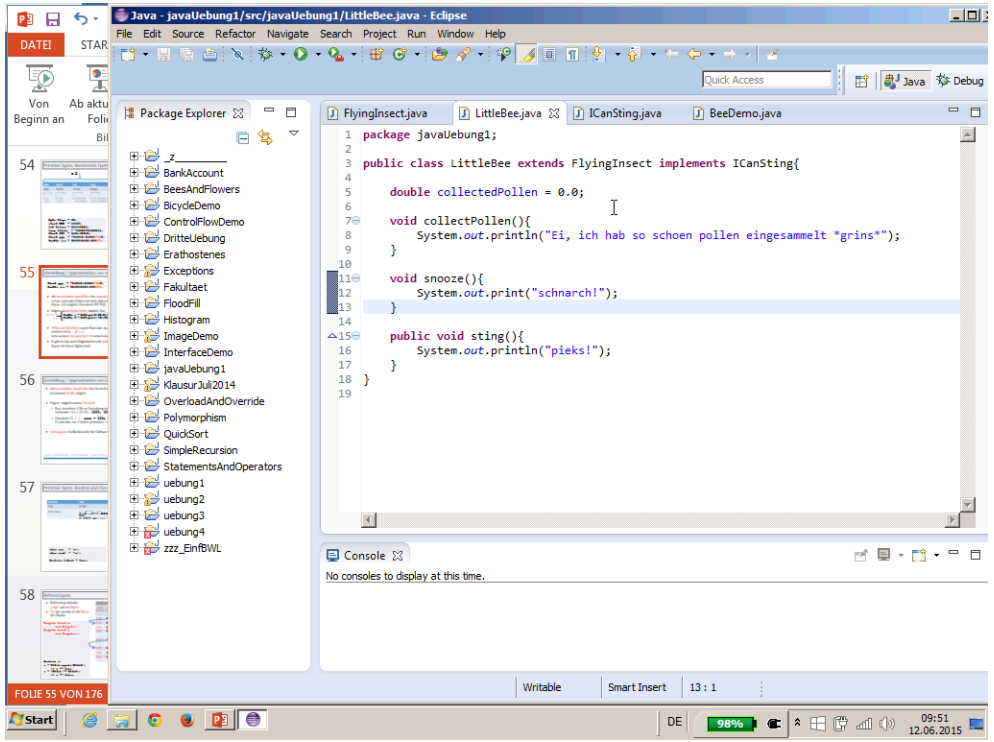
⇨ Ausgabe: `#:0 #:1 #:2 #:3 #:4 #:5 #:6`

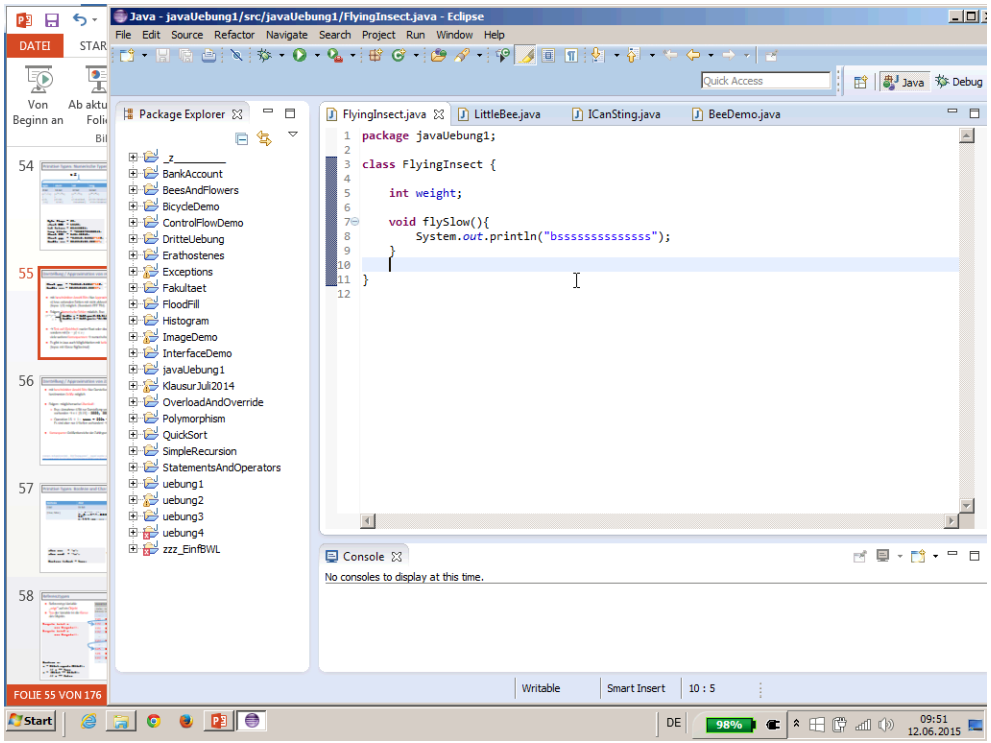
Schleifen: for

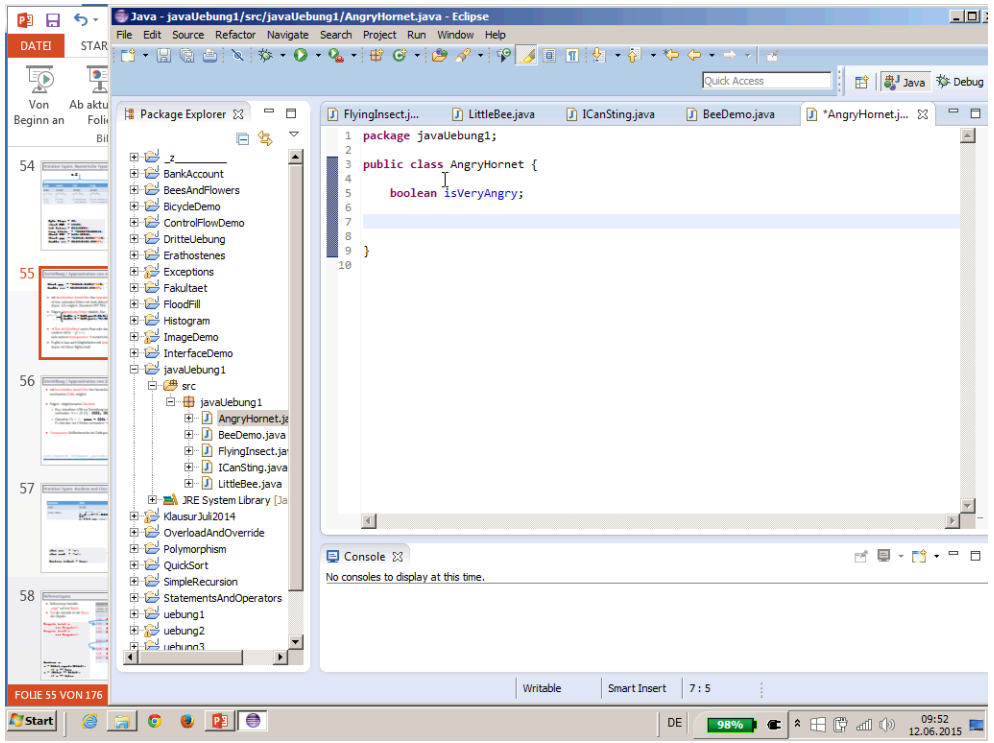
for: (Üblicherweise:) Mache etwas eine festgelegte Anzahl von Malen (Iterationen)

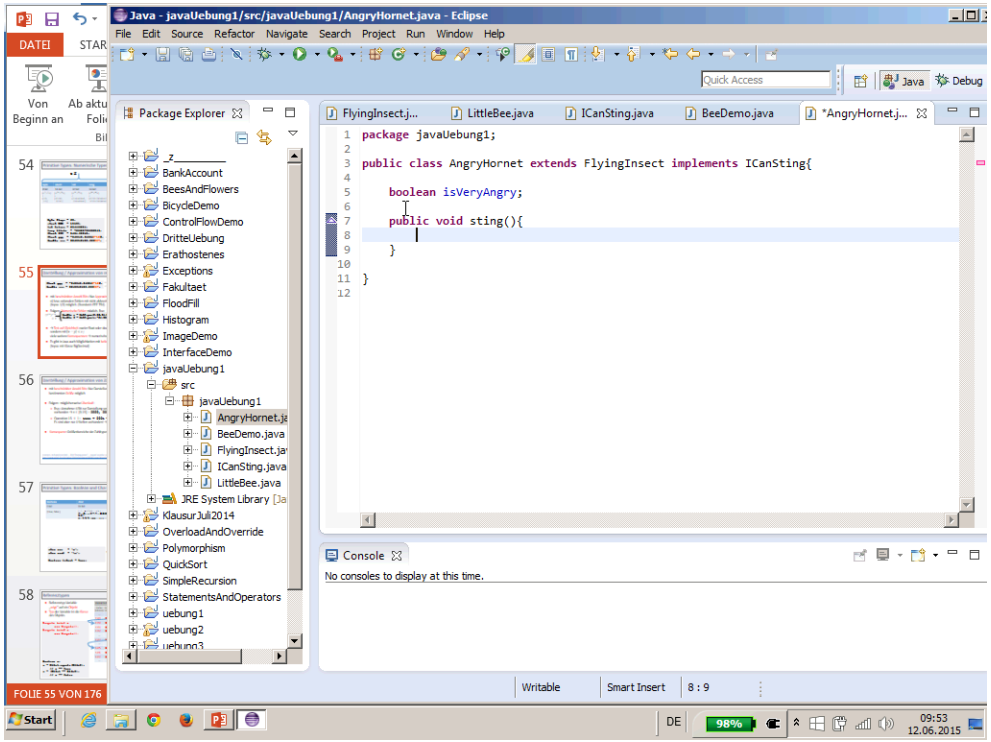
```
for (int i=0; i<7; i++) { // loop will be executed 7 times
        System.out.print("#:" + i + " ");
}
```

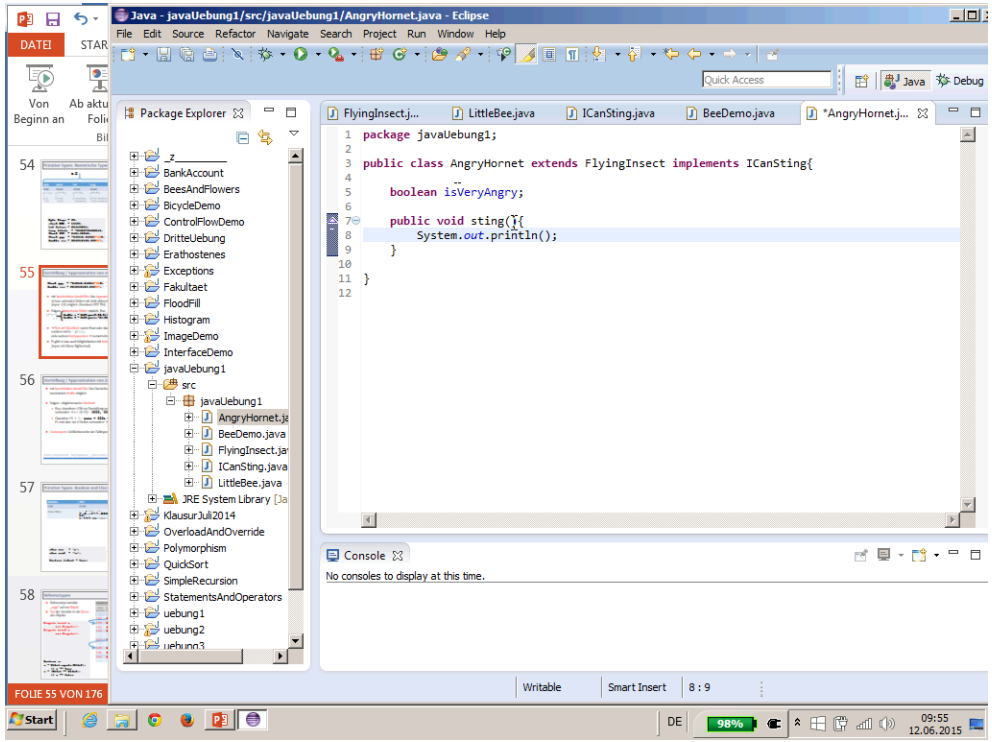
⇨ Ausgabe: #:0 #:1 #:2 #:3 #:4 #:5 #:6

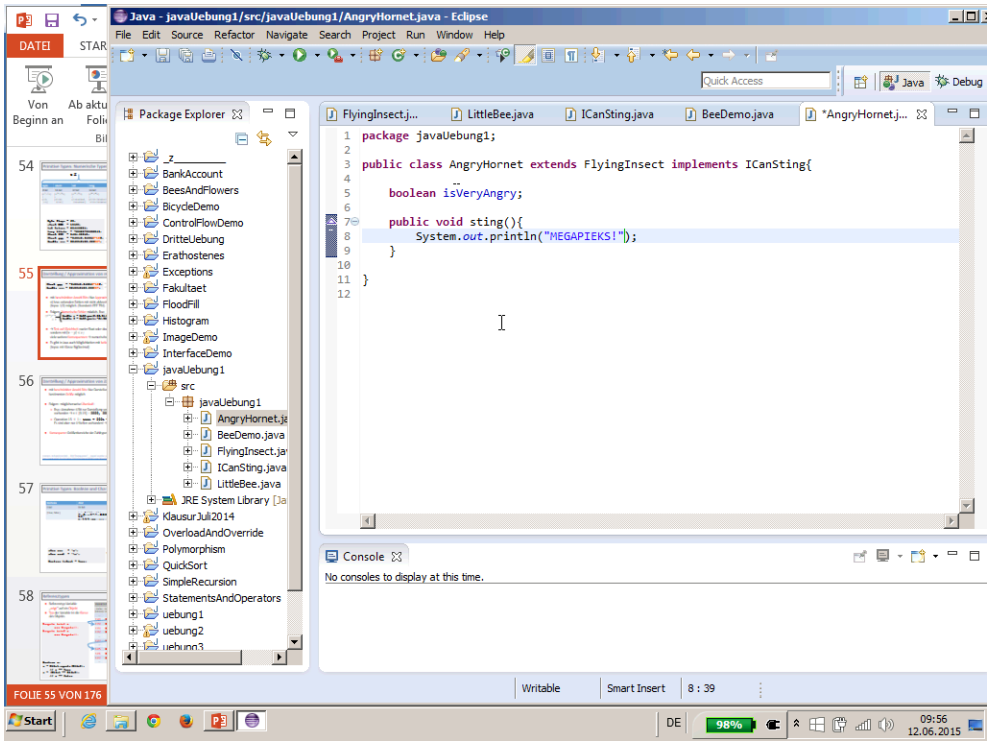


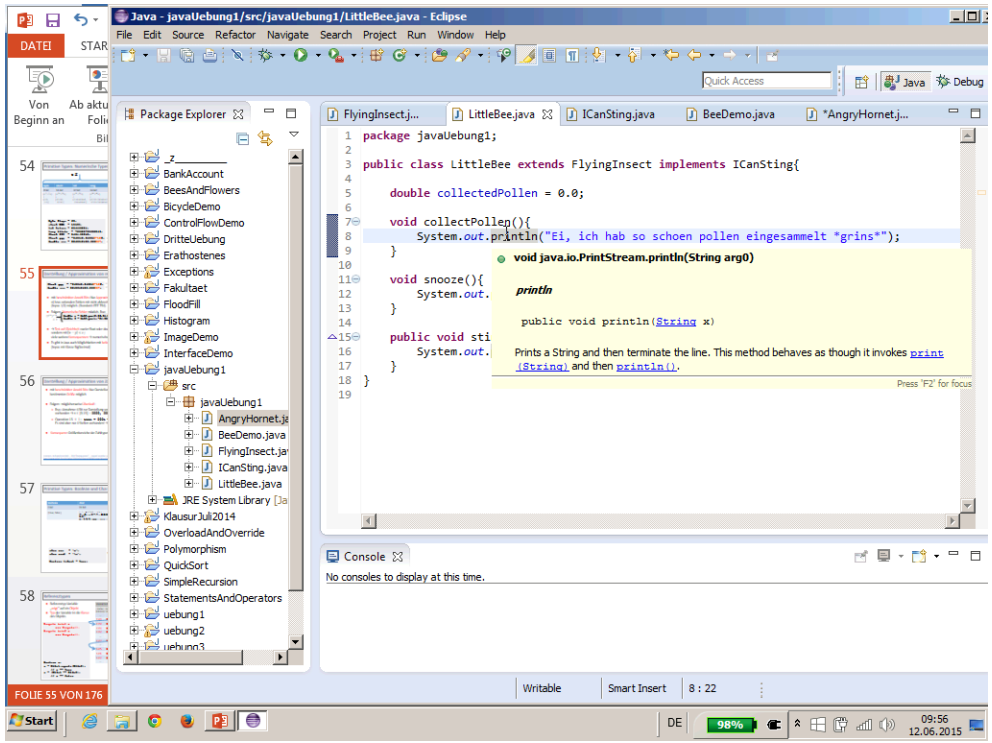


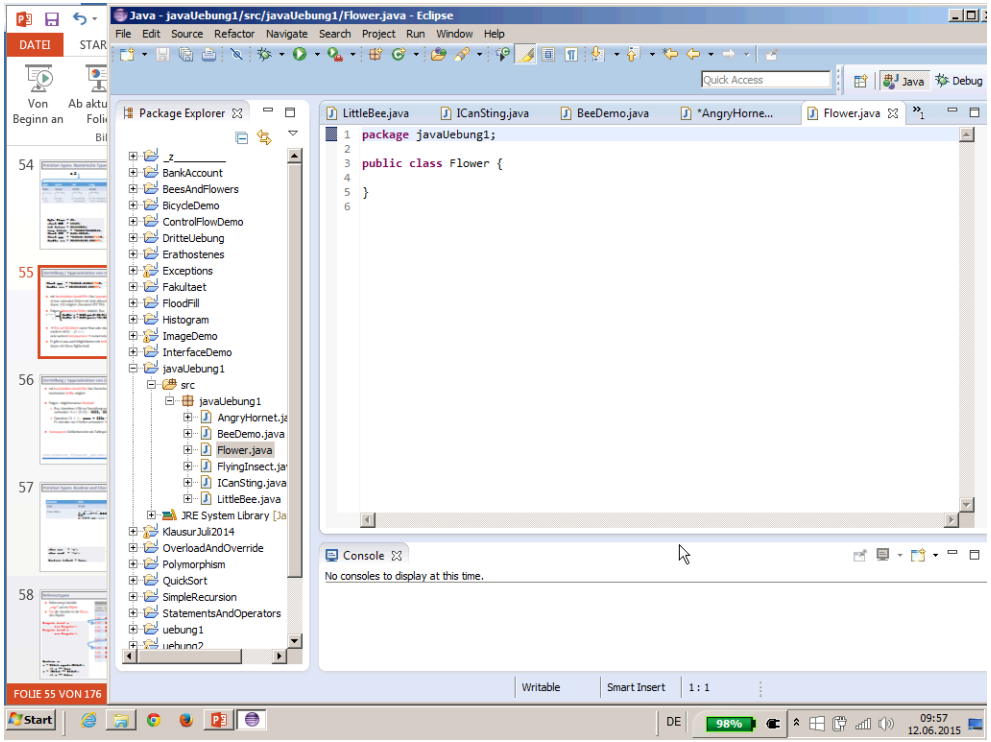


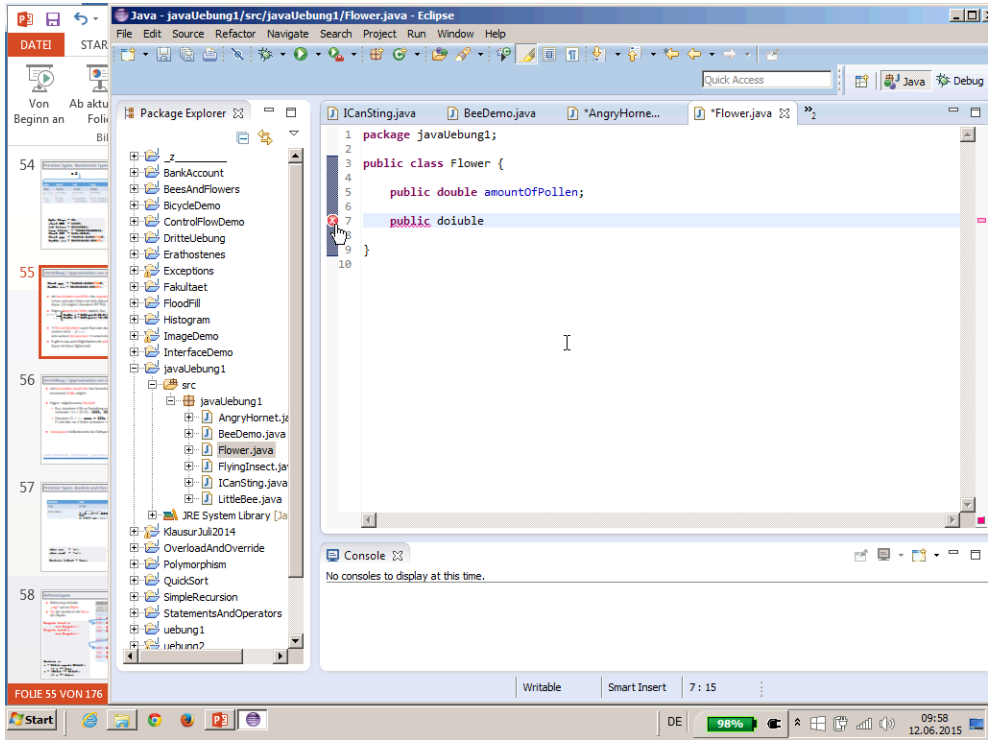

Java - javaUebung1/src/javaUebung1/AngryHornet.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsect.j... LittleBee.java ICanSting.java BeeDemo.java *AngryHornet.j...
1 package javaUebung1;
2
3 public class AngryHornet extends FlyingInsect implements ICanSting{
4
5 boolean isVeryAngry;
6
7 public void sting(){
8
9 }
10
11 }
12
Console
No consoles to display at this time.
Writable Smart Insert 8 : 9
FOLIE 55 VON 176
09:53
12.06.2015

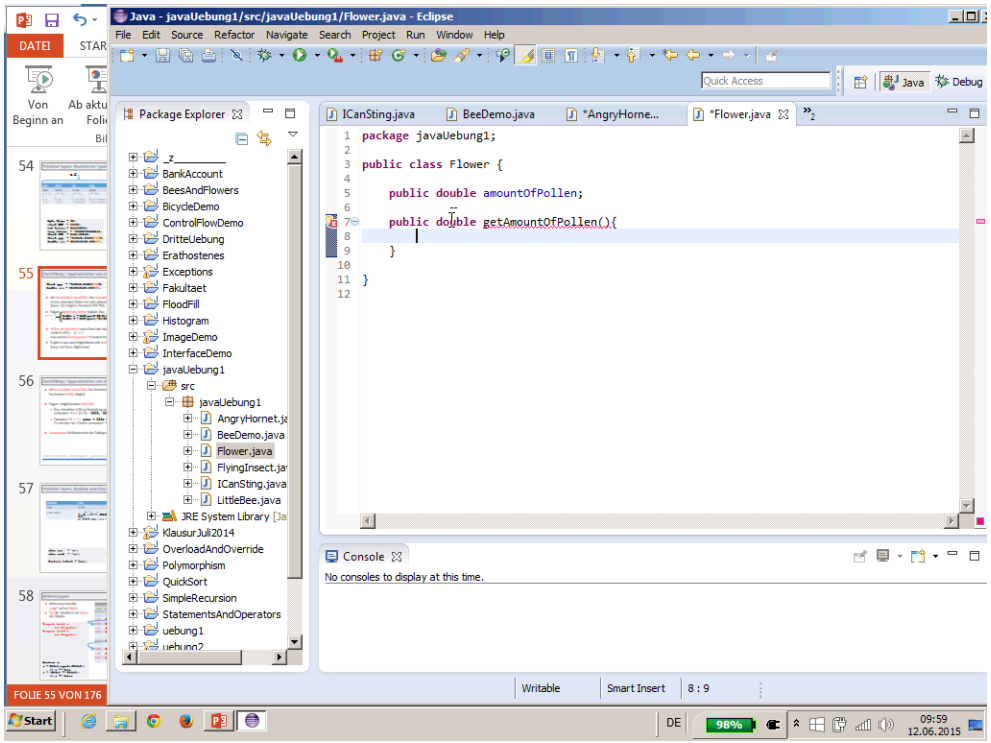

Java - javaUebung1/src/javaUebung1/AngryHornet.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsect.j... LittleBee.java ICanSting.java BeeDemo.java *AngryHornet.j...
1 package javaUebung1;
2
3 public class AngryHornet extends FlyingInsect implements ICanSting{
4
5 boolean isVeryAngry;
6
7 public void sting(){
8 System.out.println();
9 }
10
11 }
12
Console
No consoles to display at this time.
Writable Smart Insert 8 : 9
FOLIE 55 VON 176
09:55
12.06.2015

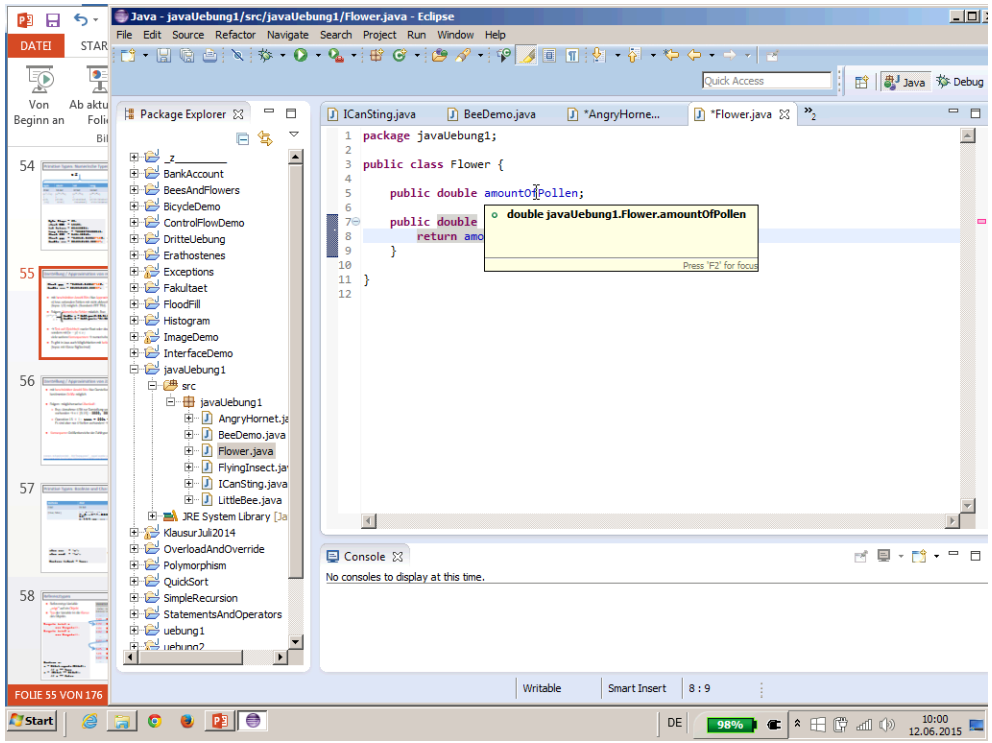

Java - javaUebung1/src/javaUebung1/AngryHornet.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsect.j... LittleBee.java ICanSting.java BeeDemo.java *AngryHornet.j...
1 package javaUebung1;
2
3 public class AngryHornet extends FlyingInsect implements ICanSting{
4
5 boolean isVeryAngry;
6
7 public void sting(){
8 System.out.println("MEGAPIEKS!");
9 }
10
11 }
12
Console
No consoles to display at this time.
Writable Smart Insert 8 : 39
FOLIE 55 VON 176
09:56
12.06.2015

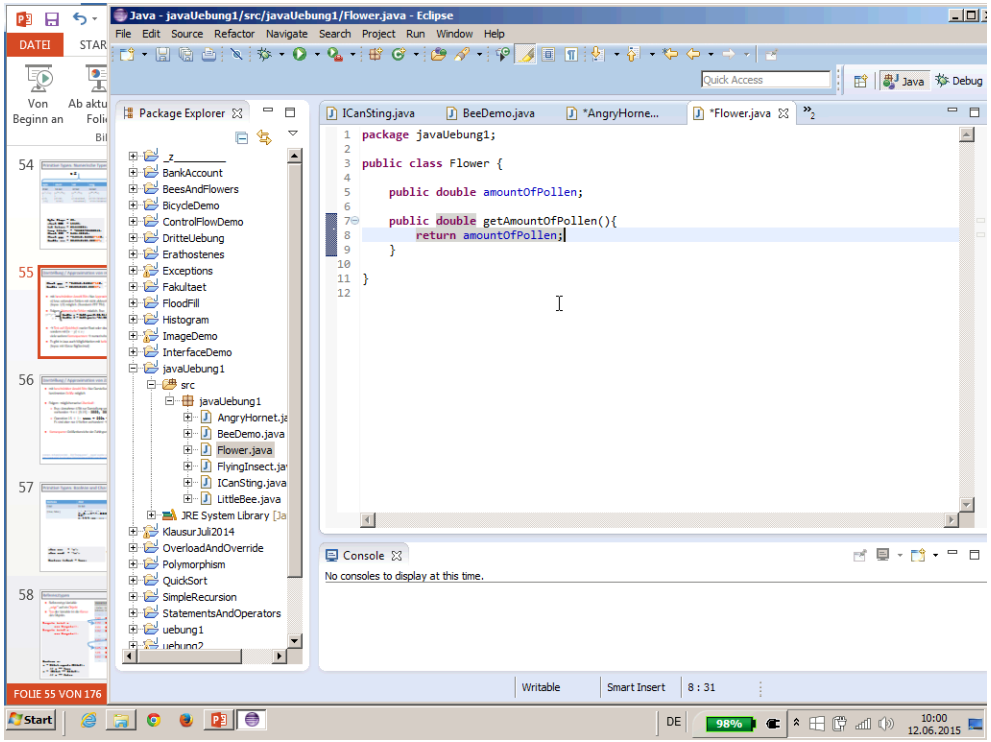

Java - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsect.j... LittleBee.java ICanSting.java BeeDemo.java *AngryHornet.j...
1 package javaUebung1;
2
3 public class LittleBee extends FlyingInsect implements ICanSting{
4
5 double collectedPollen = 0.0;
6
7 void collectPollen(){
8 System.out.println("Ei, ich hab so schoen pollen eingesammelt *grins*");
9 }
10
11 void snooze(){
12 System.out.
13 }
14
15 public void sti
16 System.out.
17 }
18 }
19
void java.io.PrintStream.println(String arg0)
println
public void println(String x)
Prints a String and then terminate the line. This method behaves as though it invokes print(String) and then println().
Press 'F2' for focus
Console
No consoles to display at this time.
Writable Smart Insert 8 : 22
FOLIE 55 VON 176
09:56
12.06.2015

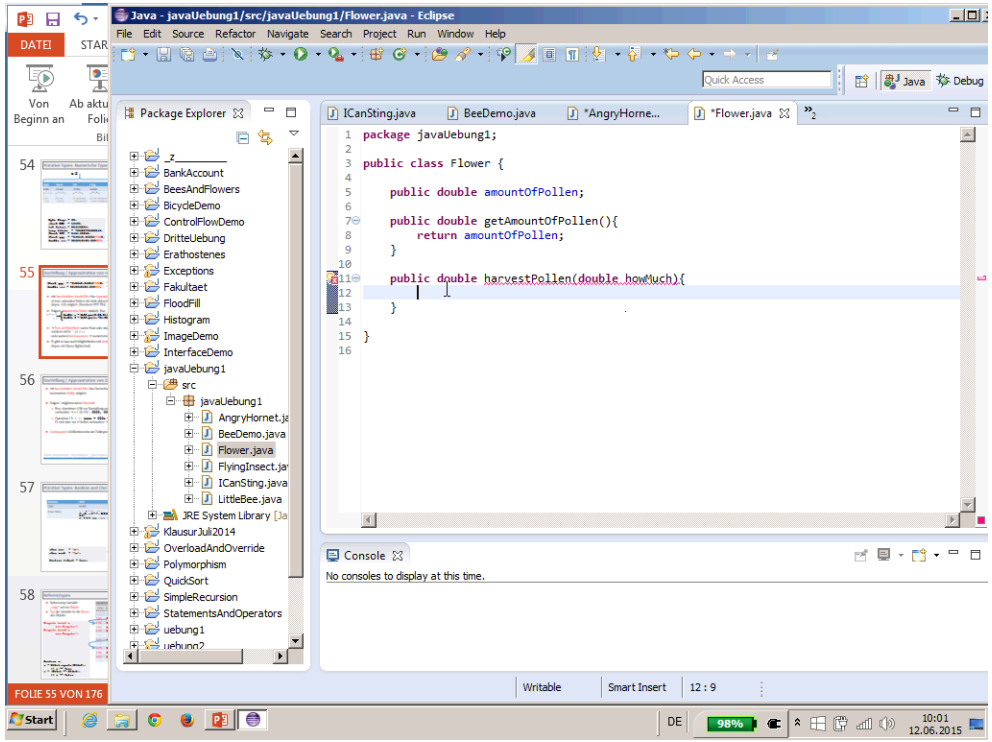

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
*AngryHorne...
Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 }
6
Console
No consoles to display at this time.
Writable
Smart Insert
1 : 1
FOLIE 55 VON 176
09:57
12.06.2015

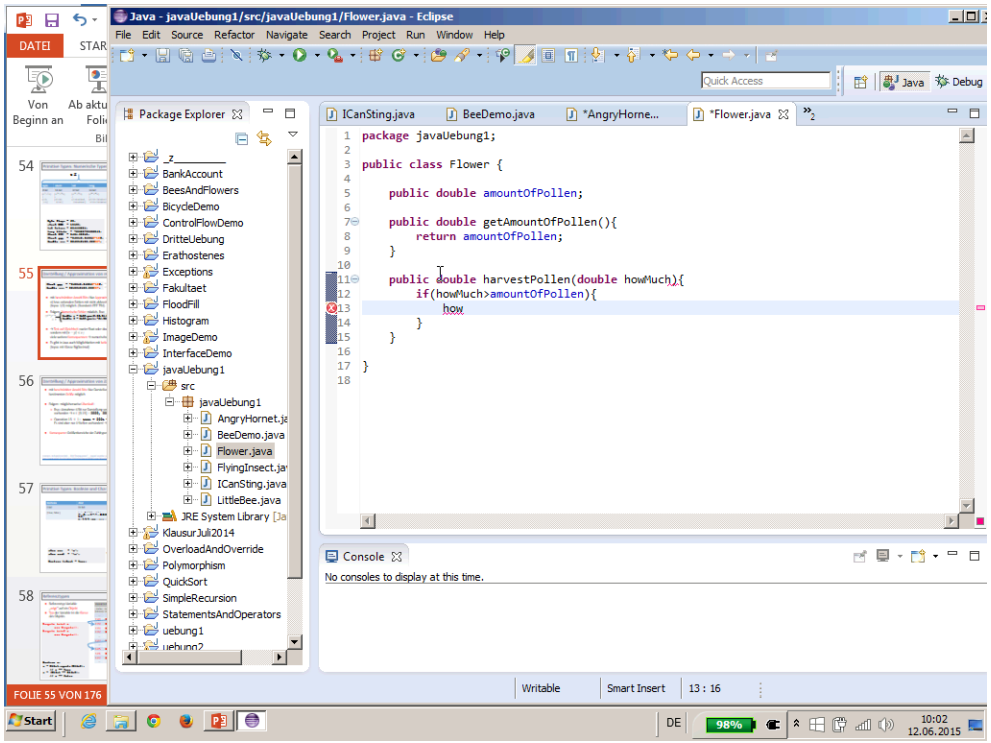

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
BeeDemo.java
*AngryHorne...
*Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen;
6
7 public doiuble
8
9 }
10
Console
No consoles to display at this time.
Writable
Smart Insert
7 : 15
FOLIE 55 VON 176
09:58
12.06.2015

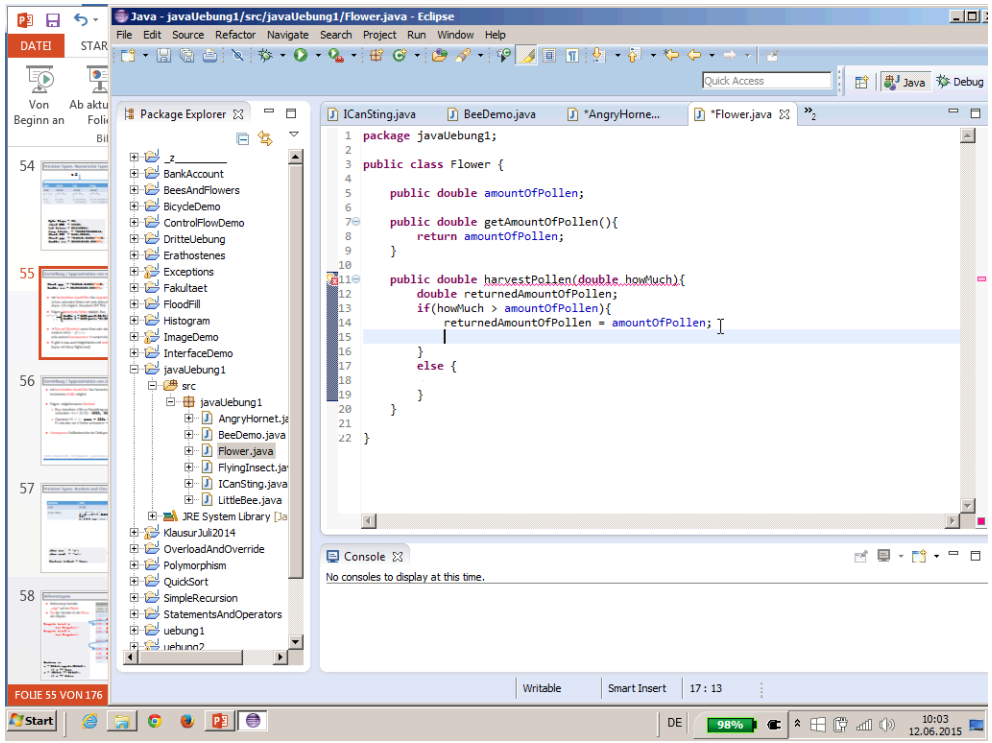

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
BeeDemo.java
*AngryHorne...
*Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen;
6
7 public double getAmountOfPollen(){
8
9 }
10
11 }
12
Console
No consoles to display at this time.
Writable
Smart Insert
8 : 9
FOLIE 55 VON 176
09:59
12.06.2015

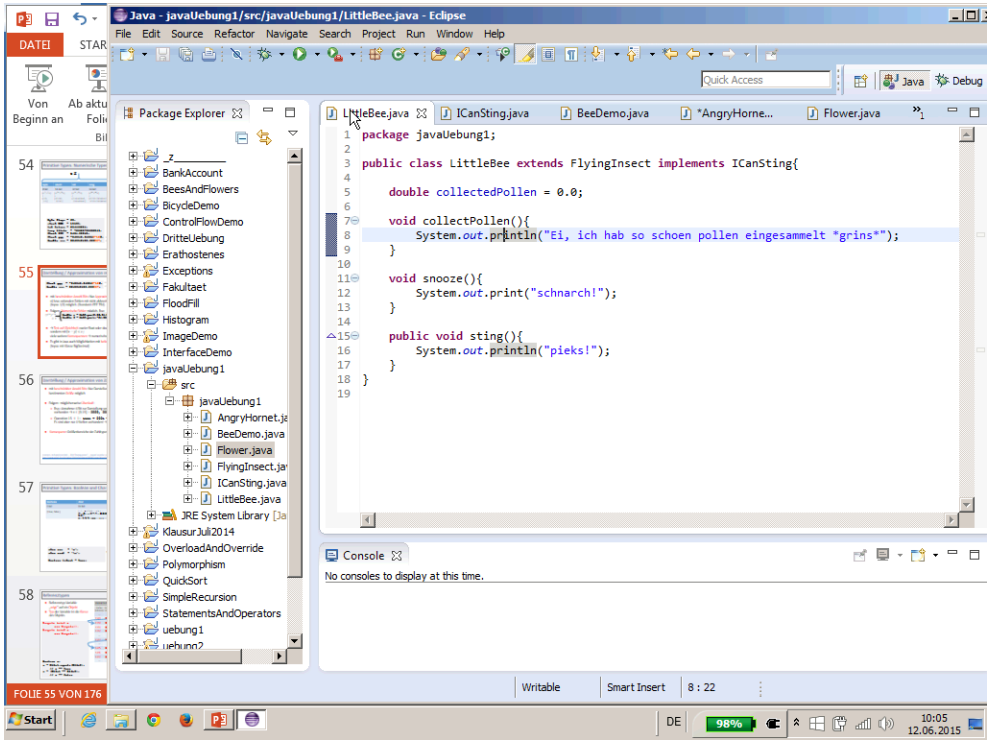

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
BeeDemo.java
*AngryHorne...
*Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen;
6
7 public double
8 return amo
9 }
10
11 }
12
double javaUebung1.Flower.amountOfPollen
Press 'F2' for focus
Console
No consoles to display at this time.
Writable
Smart Insert
8 : 9
FOLIE 55 VON 176
10:00
12.06.2015

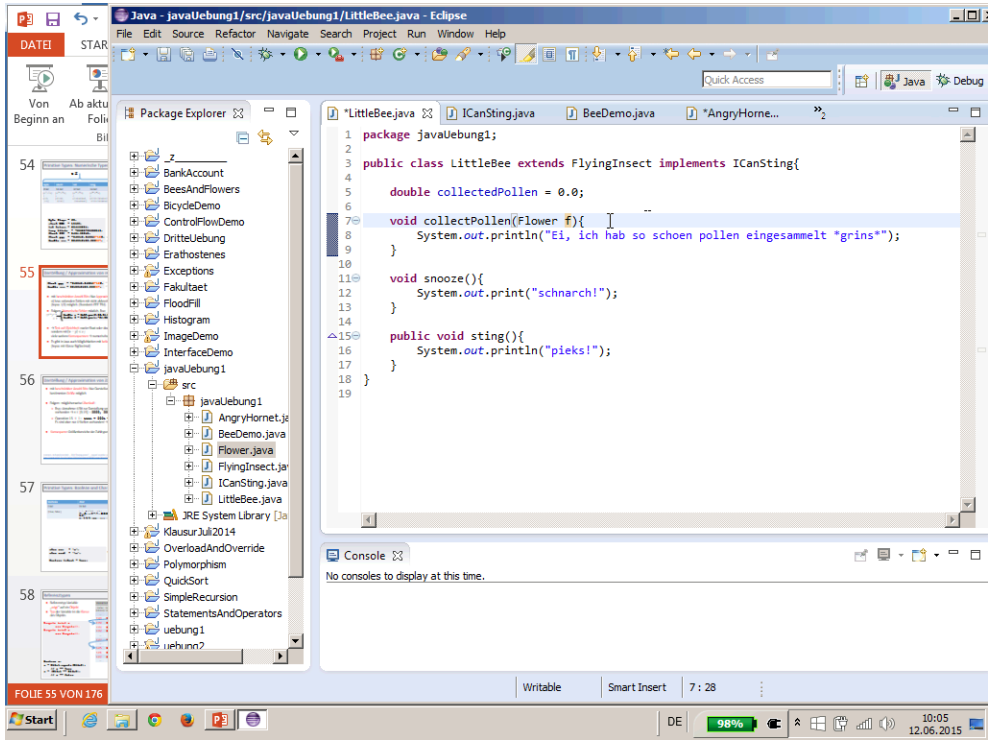

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
BeeDemo.java
*AngryHorne...
*Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen;
6
7 public double getAmountOfPollen(){
8 return amountOfPollen;
9 }
10
11 }
12
Console
No consoles to display at this time.
Writable
Smart Insert
8 : 31
FOLIE 55 VON 176

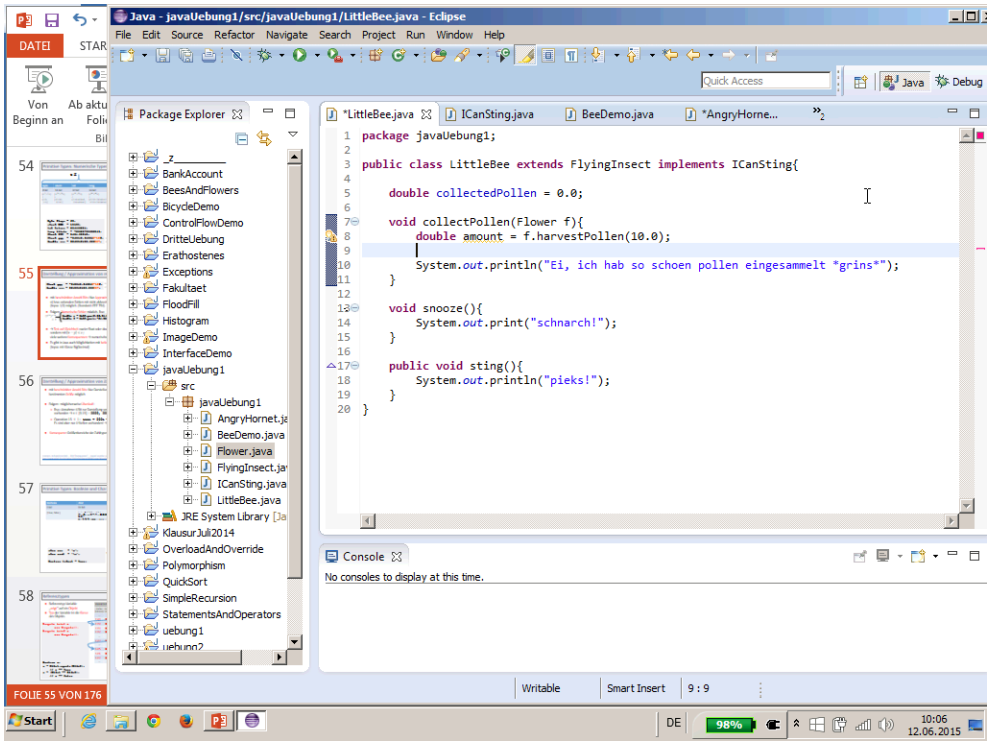

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
BeeDemo.java
*AngryHorne...
*Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen;
6
7 public double getAmountOfPollen(){
8 return amountOfPollen;
9 }
10
11 public double harvestPollen(double howMuch){
12
13 }
14
15 }
16
Console
No consoles to display at this time.
Writable
Smart Insert
12 : 9
FOLIE 55 VON 176

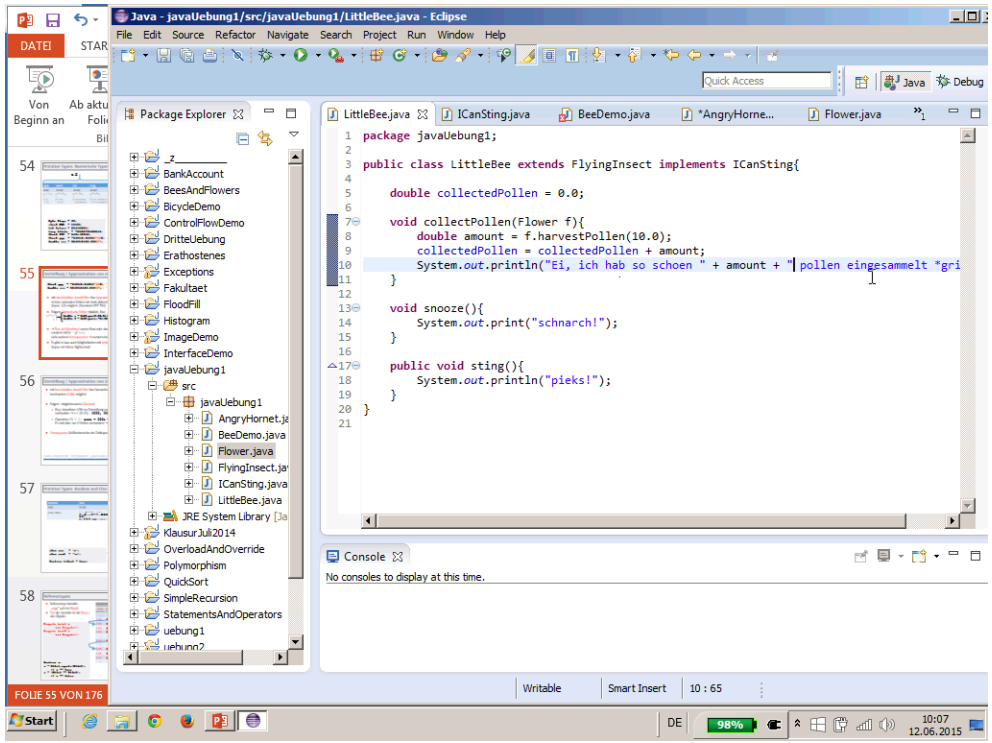

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
BeeDemo.java
*AngryHorne...
*Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen;
6
7 public double getAmountOfPollen(){
8 return amountOfPollen;
9 }
10
11 public double harvestPollen(double howMuch){
12 if(howMuch>amountOfPollen){
13 how
14 }
15 }
16
17 }
18
Console
No consoles to display at this time.
Writable
Smart Insert
13 : 16
FOLIE 55 VON 176

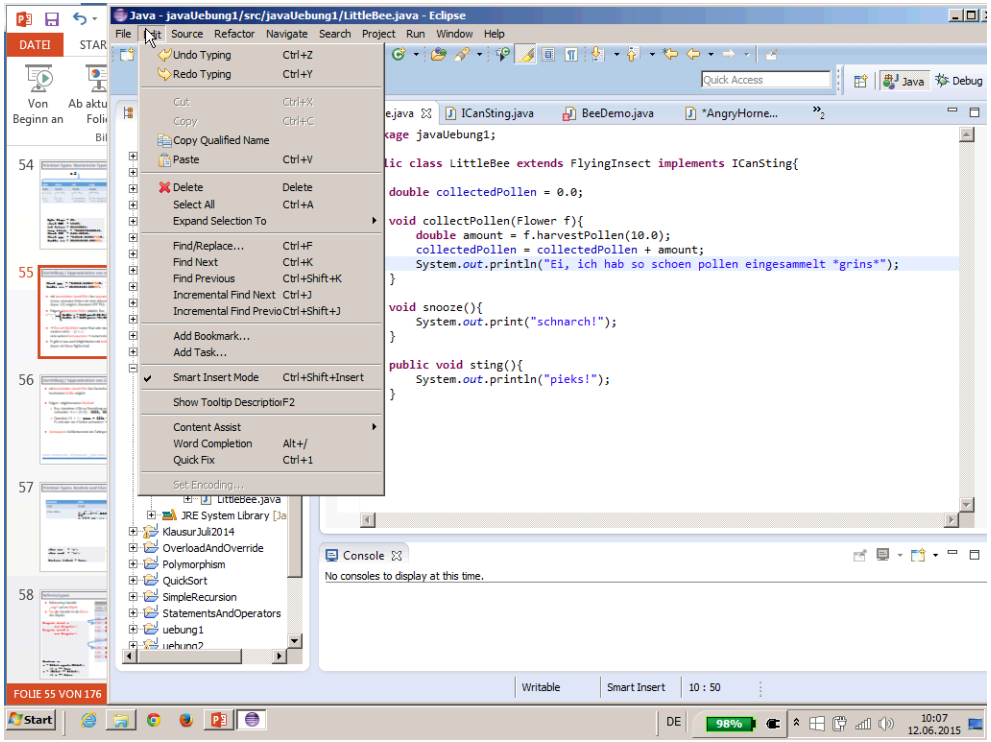

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
BeeDemo.java
*AngryHorne...
*Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen;
6
7 public double getAmountOfPollen(){
8 return amountOfPollen;
9 }
10
11 public double harvestPollen(double howMuch){
12 double returnedAmountOfPollen;
13 if(howMuch > amountOfPollen){
14 returnedAmountOfPollen = amountOfPollen;
15
16 }
17 else {
18
19 }
20 }
21
22 }
Console
No consoles to display at this time.
Writable
Smart Insert
17 : 13
FOLIE 55 VON 176

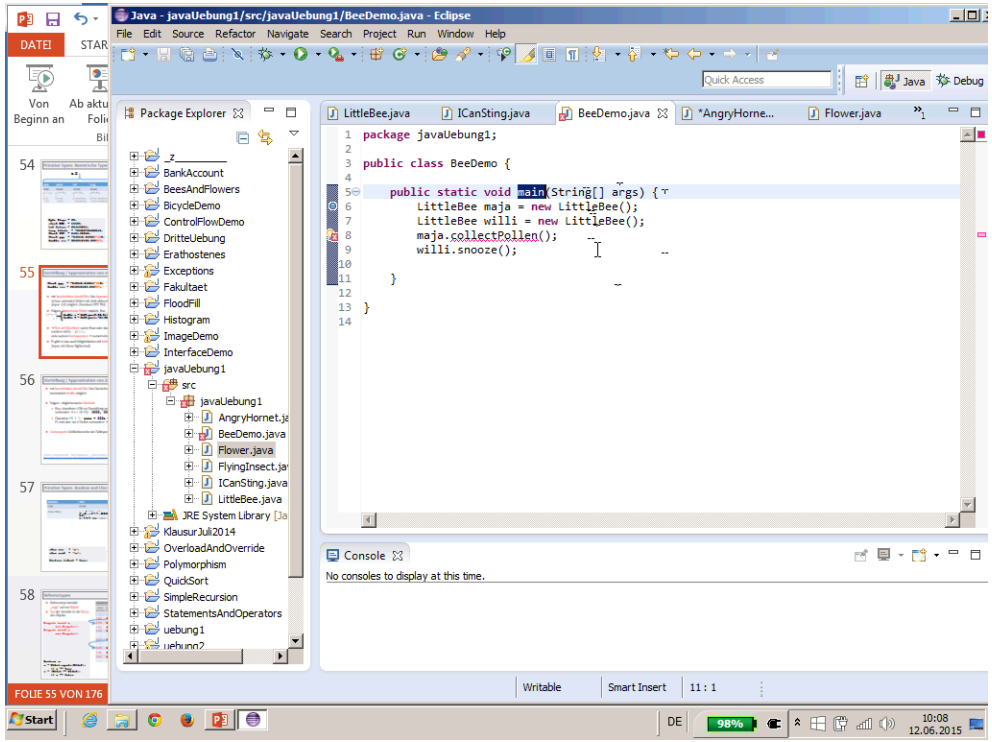

Java - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
LittleBee.java
ICanSting.java
BeeDemo.java
*AngryHorne...
Flower.java
package javaUebung1;
public class LittleBee extends FlyingInsect implements ICanSting{
double collectedPollen = 0.0;
void collectPollen(){
System.out.println("Ei, ich hab so schoen pollen eingesammelt *grins*");
}
void snooze(){
System.out.print("schnarch!");
}
public void sting(){
System.out.println("pieks!");
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
8 : 22

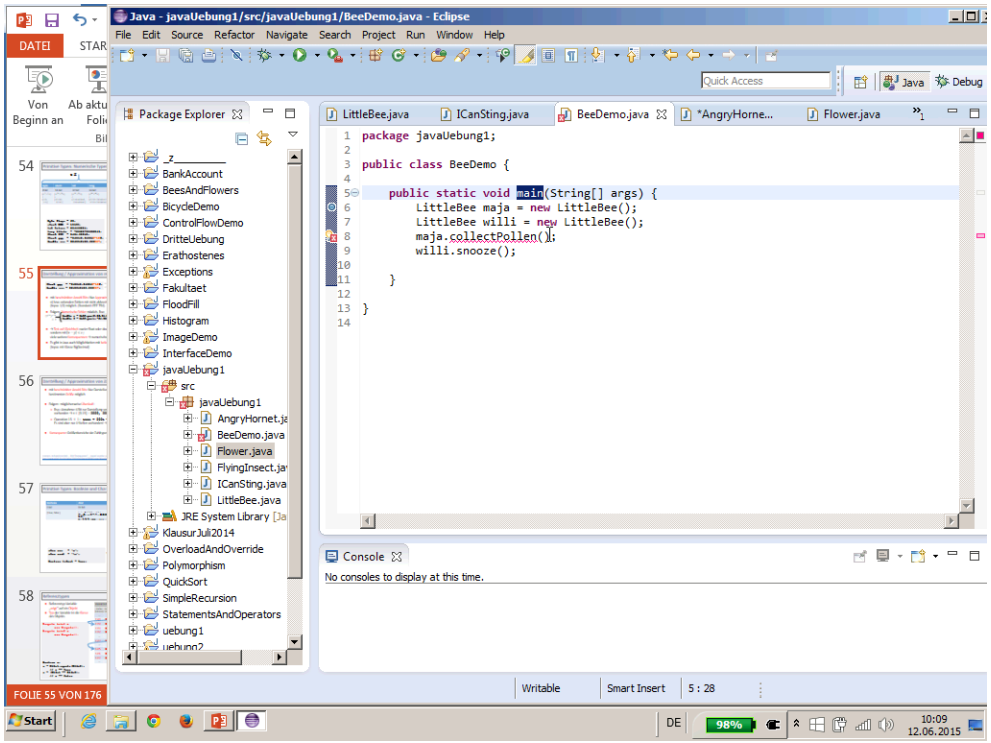

Java - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
*LittleBee.java
ICanSting.java
BeeDemo.java
*AngryHorne...
package javaUebung1;
public class LittleBee extends FlyingInsect implements ICanSting{
double collectedPollen = 0.0;
void collectPollen(Flower f){
System.out.println("Ei, ich hab so schoen pollen eingesammelt *grins*");
}
void snooze(){
System.out.print("schnarch!");
}
public void sting(){
System.out.println("pieks!");
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
7 : 28

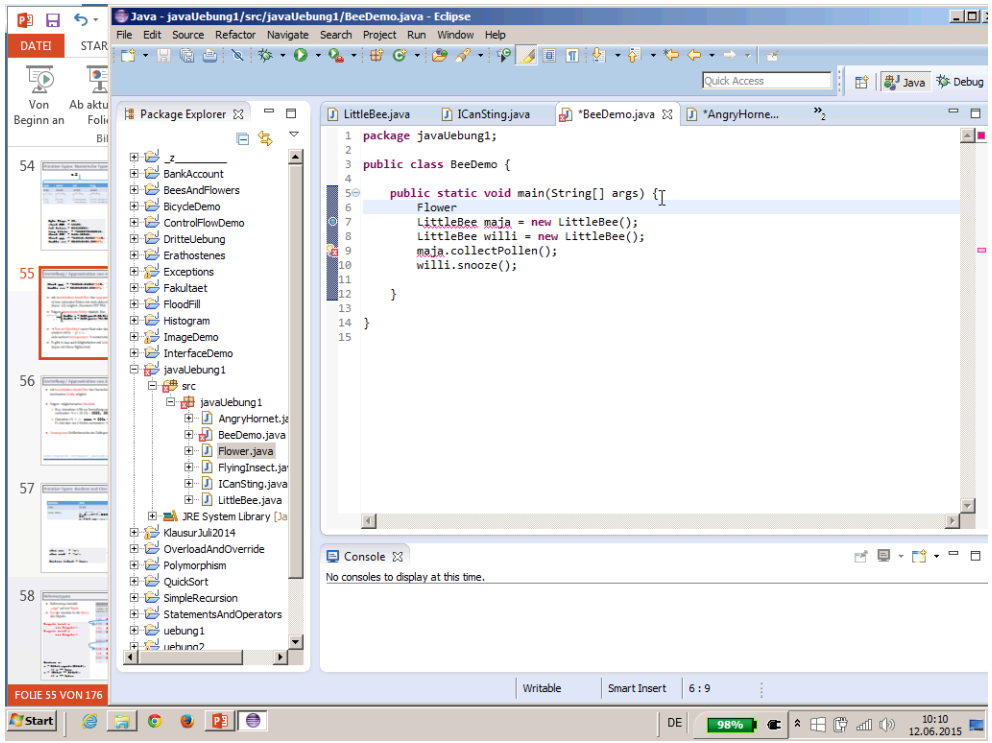

Java - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
*LittleBee.java
ICanSting.java
BeeDemo.java
*AngryHorne...
package javaUebung1;
public class LittleBee extends FlyingInsect implements ICanSting{
double collectedPollen = 0.0;
void collectPollen(Flower f){
double amount = f.harvestPollen(10.0);
System.out.println("Ei, ich hab so schoen pollen eingesammelt *grins*");
}
void snooze(){
System.out.print("schnarch!");
}
public void sting(){
System.out.println("pieks!");
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
9 : 9

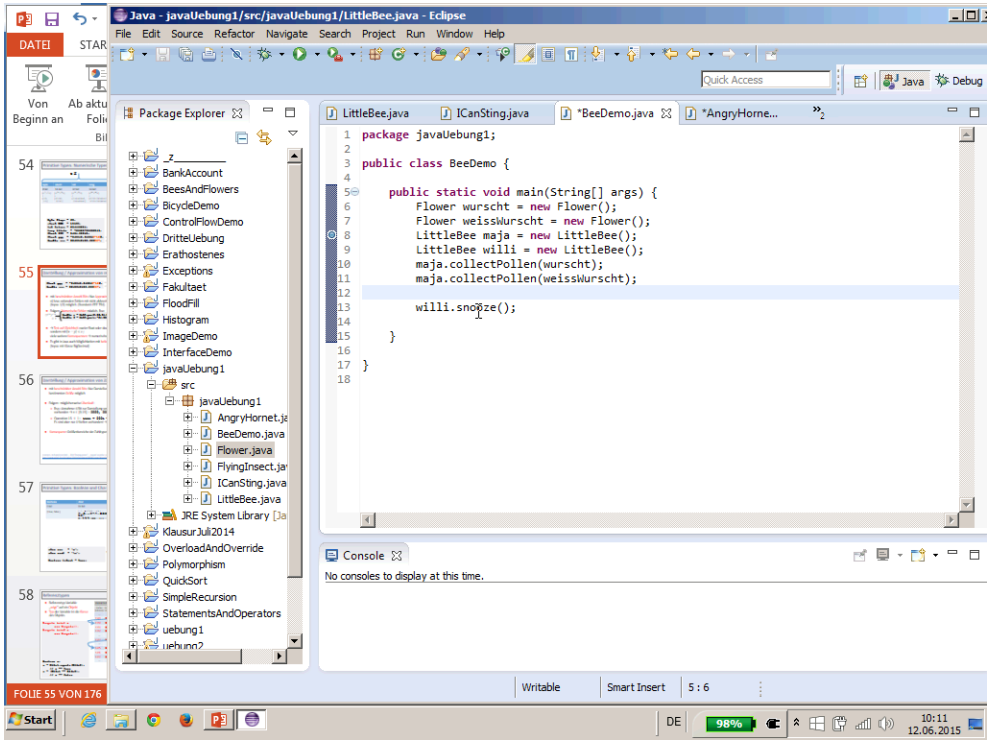

Java - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
LittleBee.java
ICanSting.java
BeeDemo.java
*AngryHorne...
Flower.java
package javaUebung1;
public class LittleBee extends FlyingInsect implements ICanSting{
double collectedPollen = 0.0;
void collectPollen(Flower f){
double amount = f.harvestPollen(10.0);
collectedPollen = collectedPollen + amount;
System.out.println("Ei, ich hab so schoen " + amount + " pollen eingesammelt *gri
}
void snooze(){
System.out.print("schnarch!");
}
public void sting(){
System.out.println("pieks!");
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
10 : 65

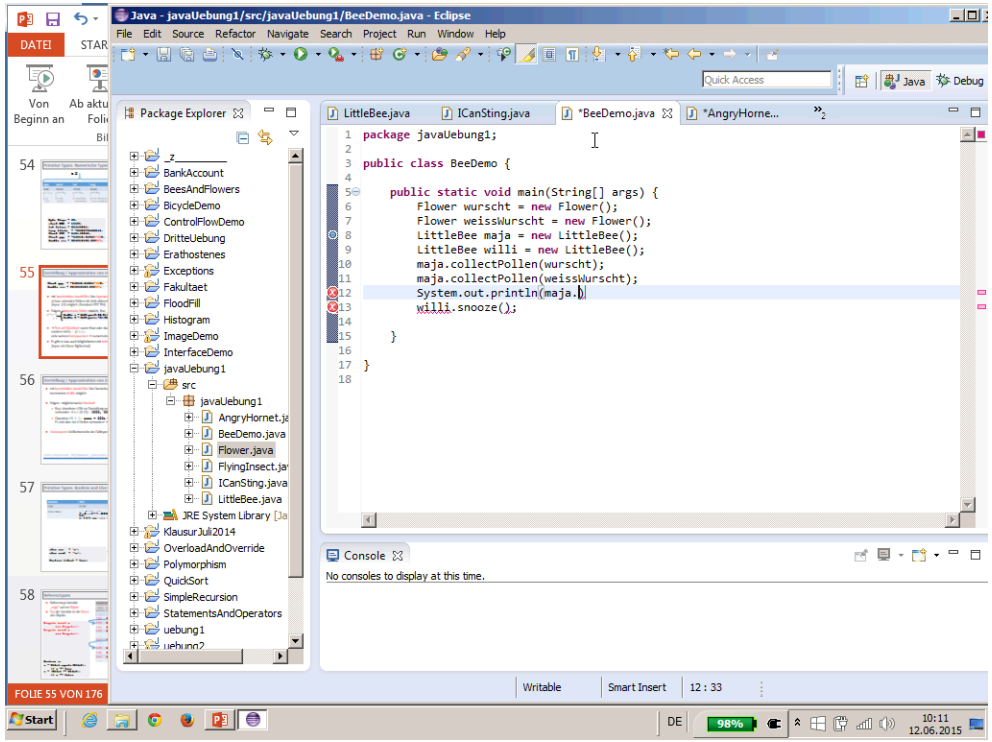

Java - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Undo Typing Ctrl+Z
Redo Typing Ctrl+Y
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Select All Ctrl+A
Expand Selection To
Find/Replace... Ctrl+F
Find Next Ctrl+K
Find Previous Ctrl+Shift+K
Incremental Find Next Ctrl+J
Add Bookmark...
Add Task...
Smart Insert Mode Ctrl+Shift+Insert
Content Assist
Word Completion Alt+/
Quick Fix Ctrl+1
ICanSting.java
BeeDemo.java
*AngryHorne...
kage javaUebung1;
lic class LittleBee extends FlyingInsect implements ICanSting{
double collectedPollen = 0.0;
void collectPollen(Flower f){
double amount = f.harvestPollen(10.0);
collectedPollen = collectedPollen + amount;
System.out.println("Ei, ich hab so schoen pollen eingesammelt *grins*");
}
void snooze(){
System.out.print("schnarch!");
}
public void sting(){
System.out.println("pieks!");
}
Console
No consoles to display at this time.
Writable
Smart Insert
10 : 50

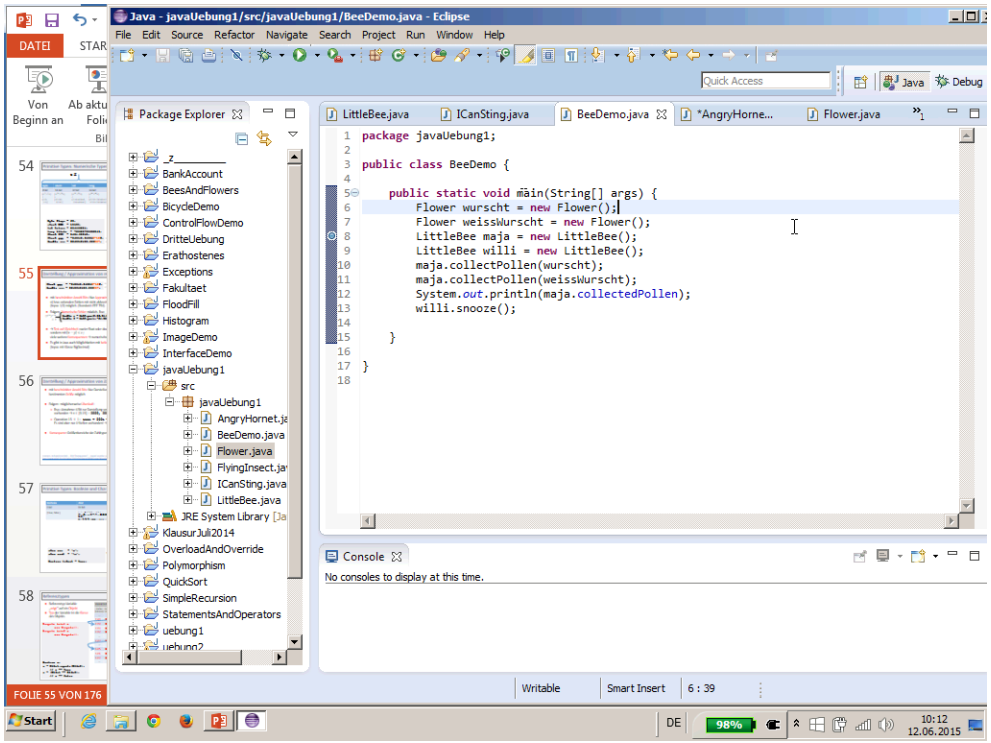

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
*AngryHorne...
Flower.java
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen();
willi.snooze();
}
}
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
src
AngryHornet.ja
BeeDemo.java
Flower.java
FlyingInsect.ja
ICanSting.java
LittleBee.java
JRE System Library
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
Console
No consoles to display at this time.
Writable
Smart Insert
11 : 1

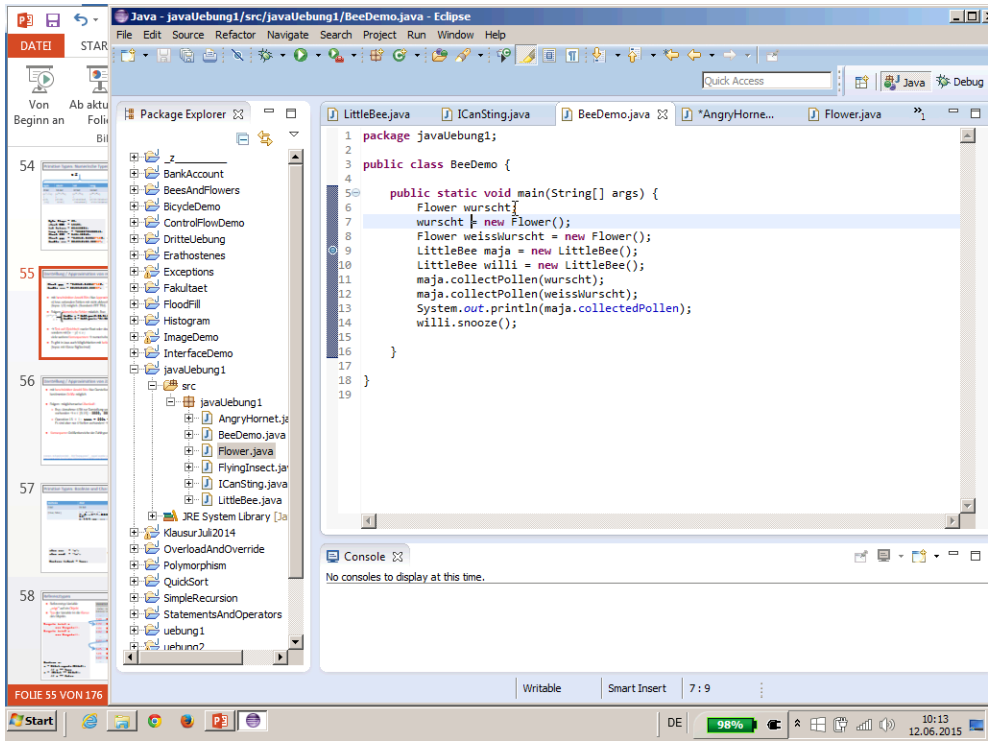

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
*AngryHorne...
Flower.java
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen();
willi.snooze();
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
5 : 28

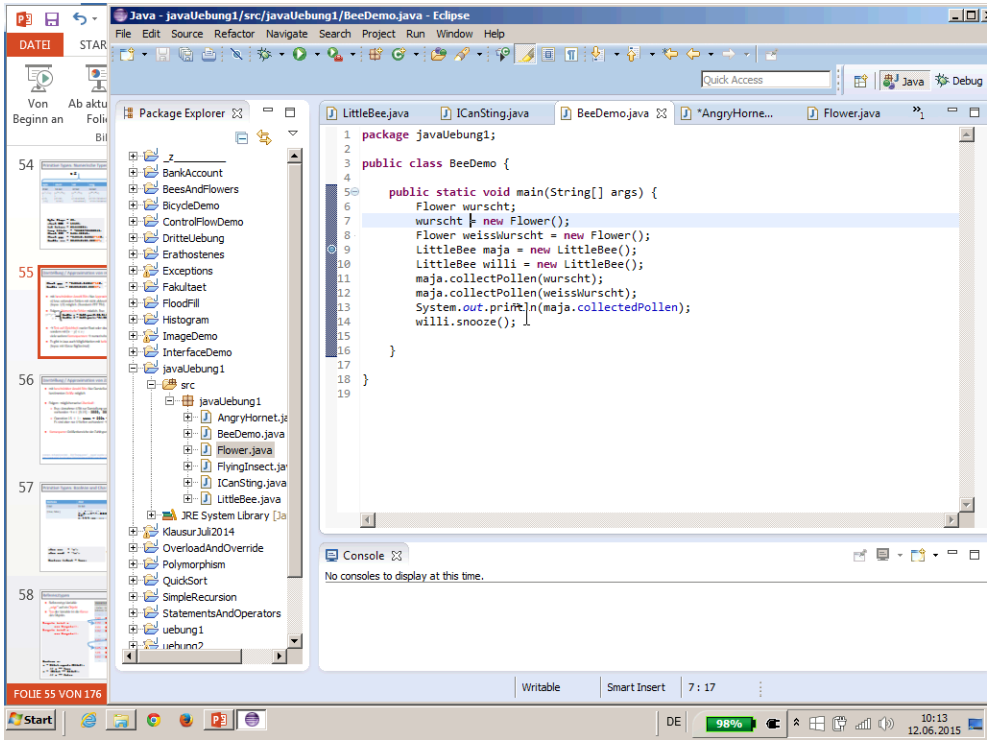

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
*BeeDemo.java
*AngryHorne...
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
Flower
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen();
willi.snooze();
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
6 : 9

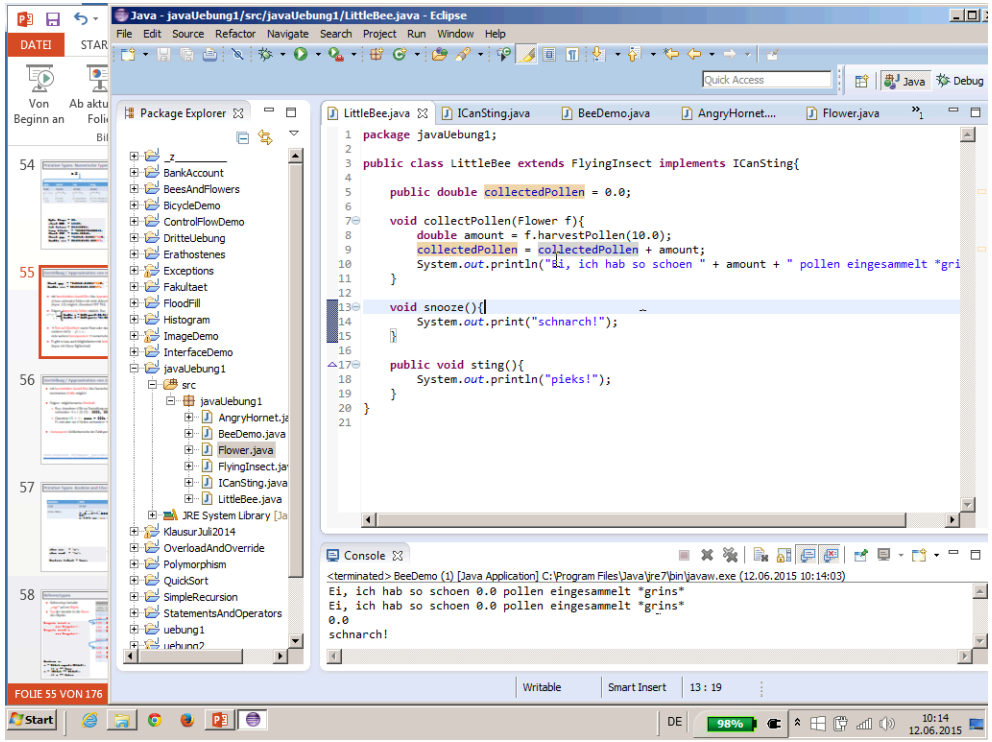

Java - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
*BeeDemo.java
*AngryHorne...
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
Flower wurscht = new Flower();
Flower weisswurscht = new Flower();
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen(wurscht);
maja.collectPollen(weisswurscht);
willi.snooze();
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
5 : 6

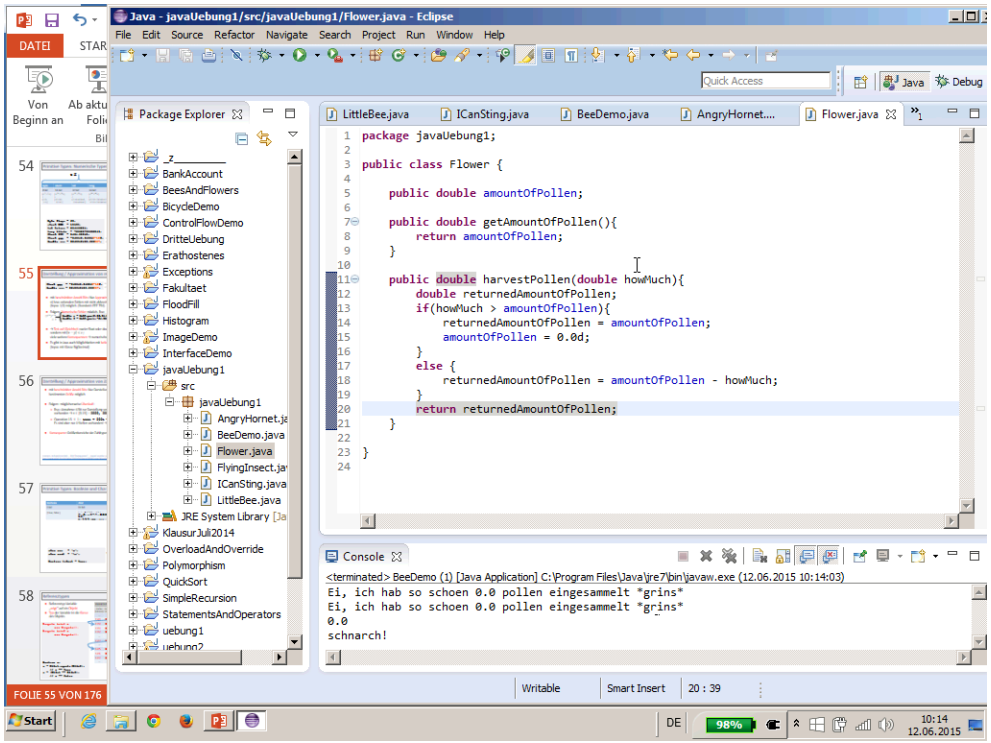

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
*BeeDemo.java
*AngryHorne...
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
Flower wurscht = new Flower();
Flower weisswurscht = new Flower();
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen(wurscht);
maja.collectPollen(weisswurscht);
System.out.println(maja.)
willi.snooze();
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
12 : 33

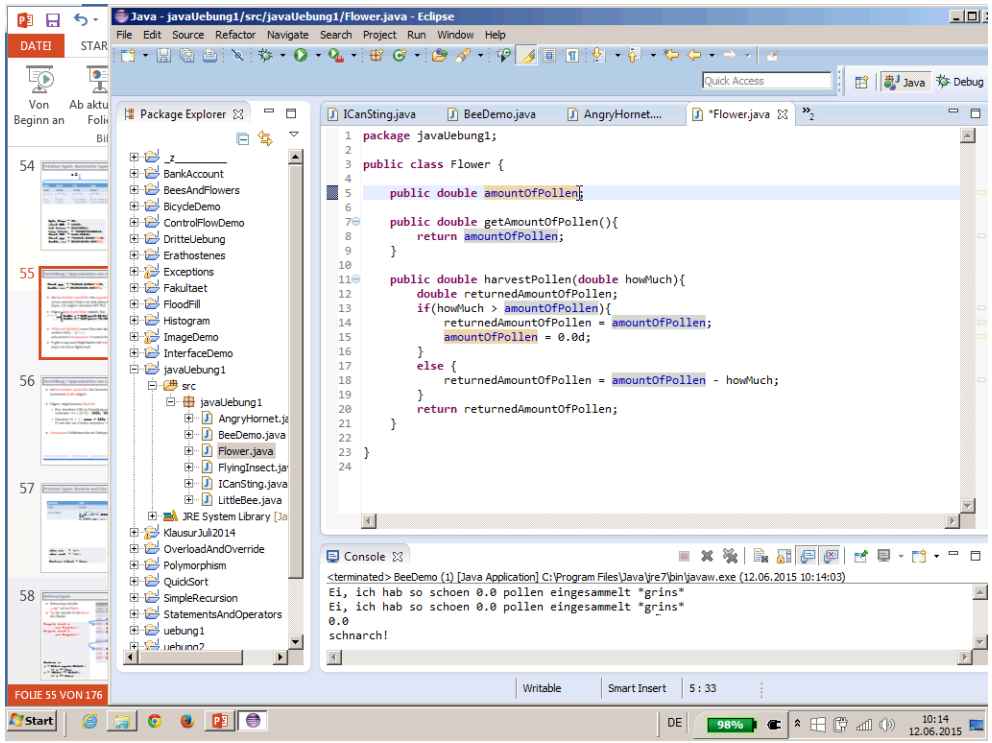

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
*AngryHorne...
Flower.java
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
Flower wurscht = new Flower();
Flower weisswurscht = new Flower();
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen(wurscht);
maja.collectPollen(weisswurscht);
System.out.println(maja.collectedPollen);
willi.snooze();
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
6 : 39

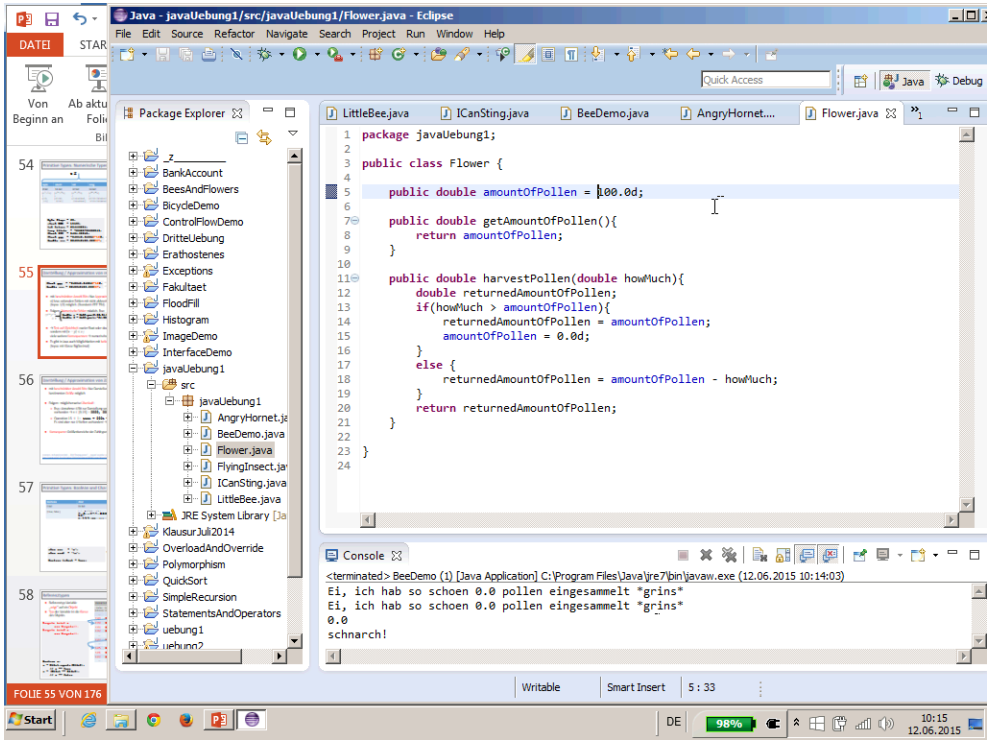

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
*AngryHorne...
Flower.java
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
Flower wurscht
wurscht = new Flower();
Flower weisswurscht = new Flower();
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen(wurscht);
maja.collectPollen(weisswurscht);
System.out.println(maja.collectedPollen);
willi.snooze();
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
7 : 9

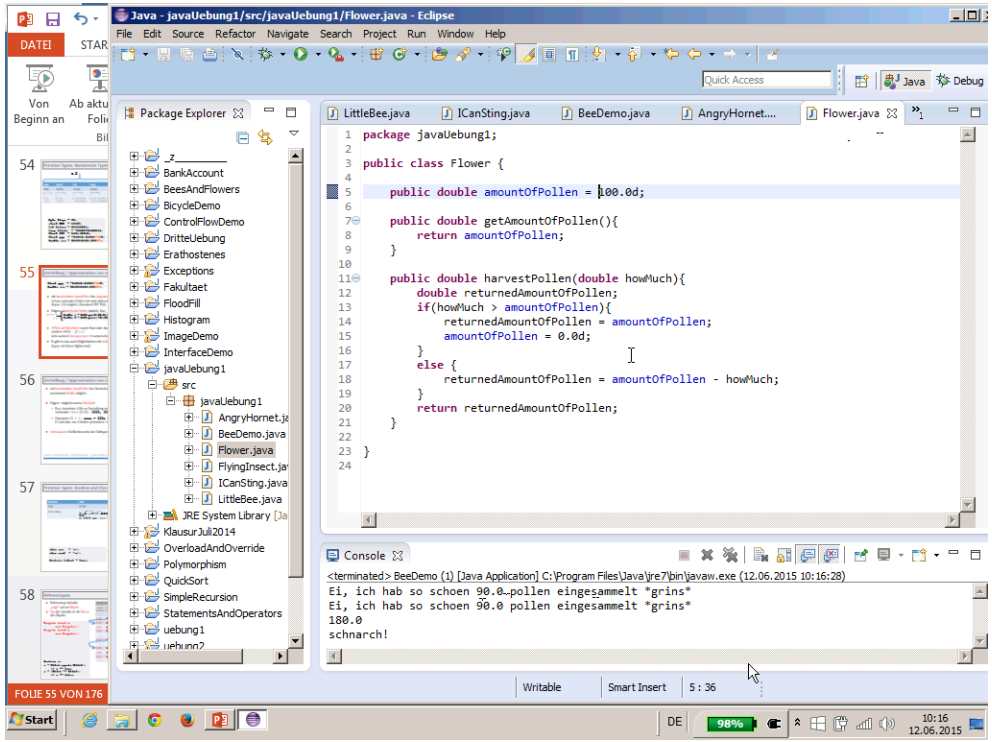

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
LittleBee.java
ICanSting.java
BeeDemo.java
*AngryHorne...
Flower.java
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
Flower wurscht;
wurscht = new Flower();
Flower weisswurscht = new Flower();
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen(wurscht);
maja.collectPollen(weisswurscht);
System.out.println(maja.collectedPollen);
willi.snooze();
}
}
Console
No consoles to display at this time.
Writable
Smart Insert
7 : 17

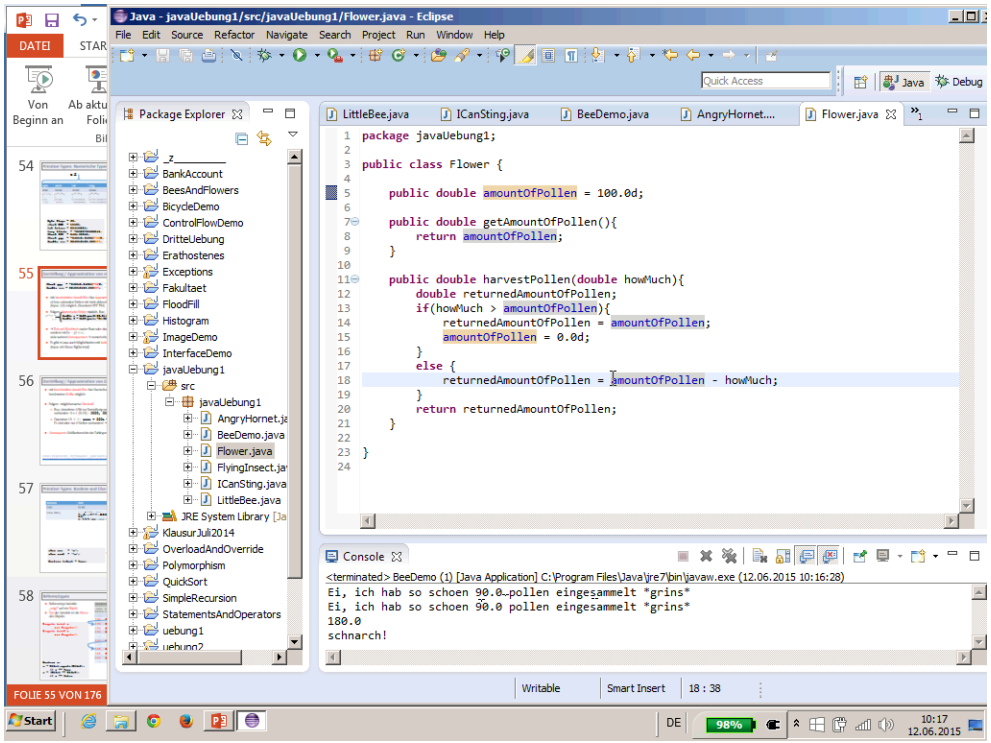

Java - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
package javaUebung1;
public class LittleBee extends FlyingInsect implements ICanSting{
public double collectedPollen = 0.0;
void collectPollen(Flower f){
double amount = f.harvestPollen(10.0);
collectedPollen = collectedPollen + amount;
System.out.println("Ei, ich hab so schoen " + amount + " pollen eingesammelt *gri
}
void snooze(){
System.out.print("schnarch!");
}
public void sting(){
System.out.println("pieks!");
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:14:03)
Ei, ich hab so schoen 0.0 pollen eingesammelt *grins*
Ei, ich hab so schoen 0.0 pollen eingesammelt *grins*
0.0
schnarch!
Writable
Smart Insert
13 : 19

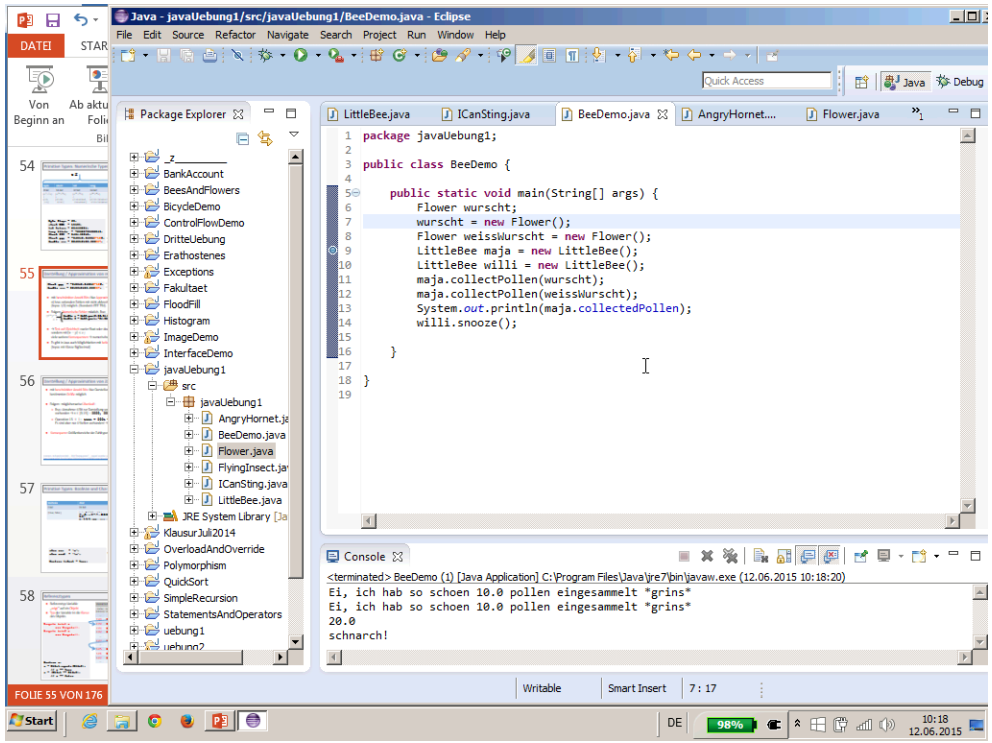

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
package javaUebung1;
public class Flower {
public double amountOfPollen;
public double getAmountOfPollen(){
return amountOfPollen;
}
public double harvestPollen(double howMuch){
double returnedAmountOfPollen;
if(howMuch > amountOfPollen){
returnedAmountOfPollen = amountOfPollen;
amountOfPollen = 0.0d;
}
else {
returnedAmountOfPollen = amountOfPollen - howMuch;
}
return returnedAmountOfPollen;
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:14:03)
Ei, ich hab so schoen 0.0 pollen eingesammelt *grins*
Ei, ich hab so schoen 0.0 pollen eingesammelt *grins*
0.0
schnarch!
Writable
Smart Insert
20 : 39

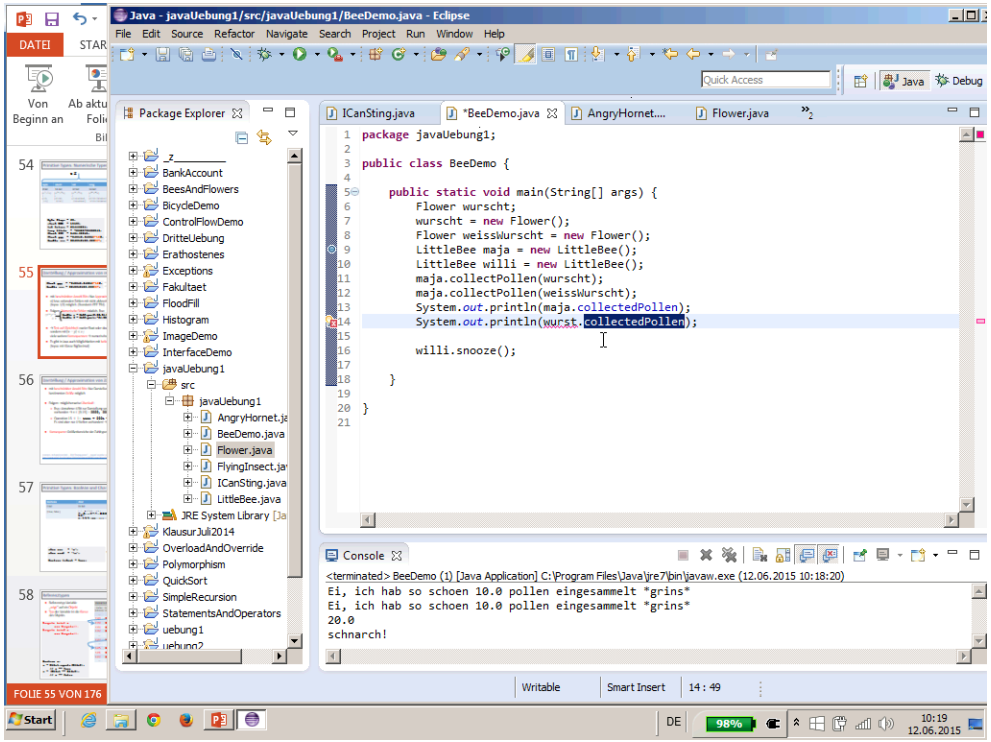

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
ICanSting.java
BeeDemo.java
AngryHornet....
*Flower.java
package javaUebung1;
public class Flower {
public double amountOfPollen;
public double getAmountOfPollen(){
return amountOfPollen;
}
public double harvestPollen(double howMuch){
double returnedAmountOfPollen;
if(howMuch > amountOfPollen){
returnedAmountOfPollen = amountOfPollen;
amountOfPollen = 0.0d;
}
else {
returnedAmountOfPollen = amountOfPollen - howMuch;
}
return returnedAmountOfPollen;
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:14:03)
Ei, ich hab so schoen 0.0 pollen eingesammelt *grins*
Ei, ich hab so schoen 0.0 pollen eingesammelt *grins*
0.0
schnarch!
Writable
Smart Insert
5 : 33

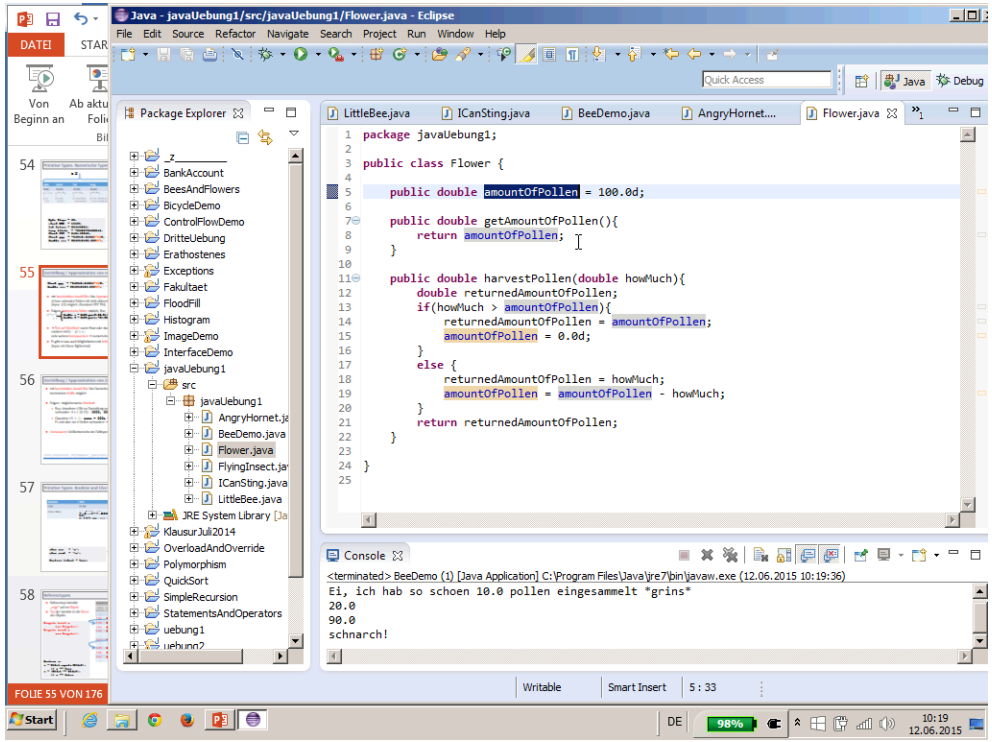

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java ICanSting.java BeeDemo.java AngryHornet.... Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen = 100.0d;
6
7 public double getAmountOfPollen(){
8 return amountOfPollen;
9 }
10
11 public double harvestPollen(double howMuch){
12 double returnedAmountOfPollen;
13 if(howMuch > amountOfPollen){
14 returnedAmountOfPollen = amountOfPollen;
15 amountOfPollen = 0.0d;
16 }
17 else {
18 returnedAmountOfPollen = amountOfPollen - howMuch;
19 }
20 return returnedAmountOfPollen;
21 }
22
23 }
24
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:14:03)
Ei, ich hab so schoen 0.0 pollen eingesammelt *grins*
Ei, ich hab so schoen 0.0 pollen eingesammelt *grins*
0.0
schnarch!
Writable Smart Insert 5 : 33
FOLIE 55 VON 176

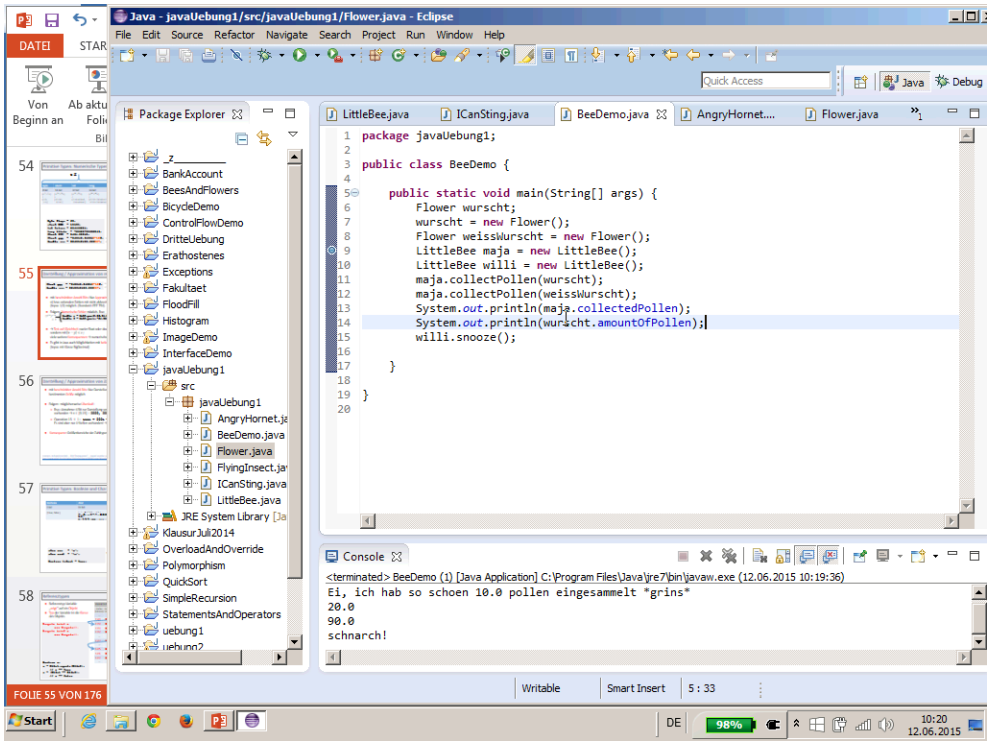

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java ICanSting.java BeeDemo.java AngryHornet.... Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen = 100.0d;
6
7 public double getAmountOfPollen(){
8 return amountOfPollen;
9 }
10
11 public double harvestPollen(double howMuch){
12 double returnedAmountOfPollen;
13 if(howMuch > amountOfPollen){
14 returnedAmountOfPollen = amountOfPollen;
15 amountOfPollen = 0.0d;
16 }
17 else {
18 returnedAmountOfPollen = amountOfPollen - howMuch;
19 }
20 return returnedAmountOfPollen;
21 }
22
23 }
24
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:16:28)
Ei, ich hab so schoen 90.0 pollen eingesammelt *grins*
Ei, ich hab so schoen 90.0 pollen eingesammelt *grins*
180.0
schnarch!
Writable Smart Insert 5 : 36
FOLIE 55 VON 176

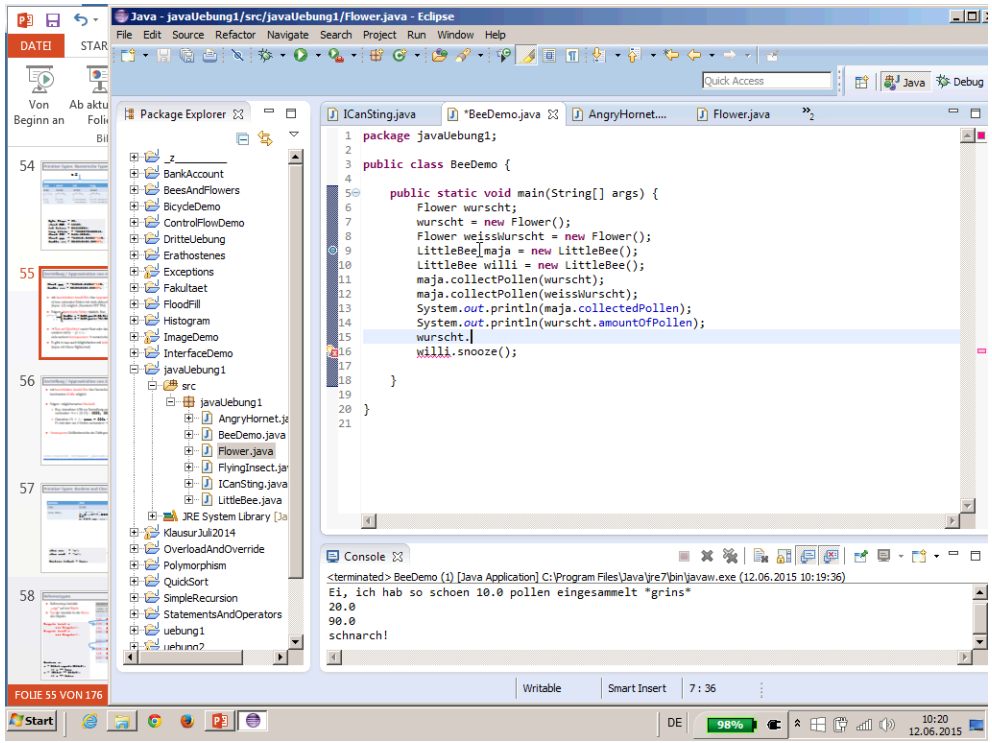

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java ICanSting.java BeeDemo.java AngryHornet.... Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen = 100.0d;
6
7 public double getAmountOfPollen(){
8 return amountOfPollen;
9 }
10
11 public double harvestPollen(double howMuch){
12 double returnedAmountOfPollen;
13 if(howMuch > amountOfPollen){
14 returnedAmountOfPollen = amountOfPollen;
15 amountOfPollen = 0.0d;
16 }
17 else {
18 returnedAmountOfPollen = amountOfPollen - howMuch;
19 }
20 return returnedAmountOfPollen;
21 }
22
23 }
24
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:16:28)
Ei, ich hab so schoen 90.0 pollen eingesammelt *grins*
Ei, ich hab so schoen 90.0 pollen eingesammelt *grins*
180.0
schnarch!
Writable Smart Insert 18 : 38
FOLIE 55 VON 176

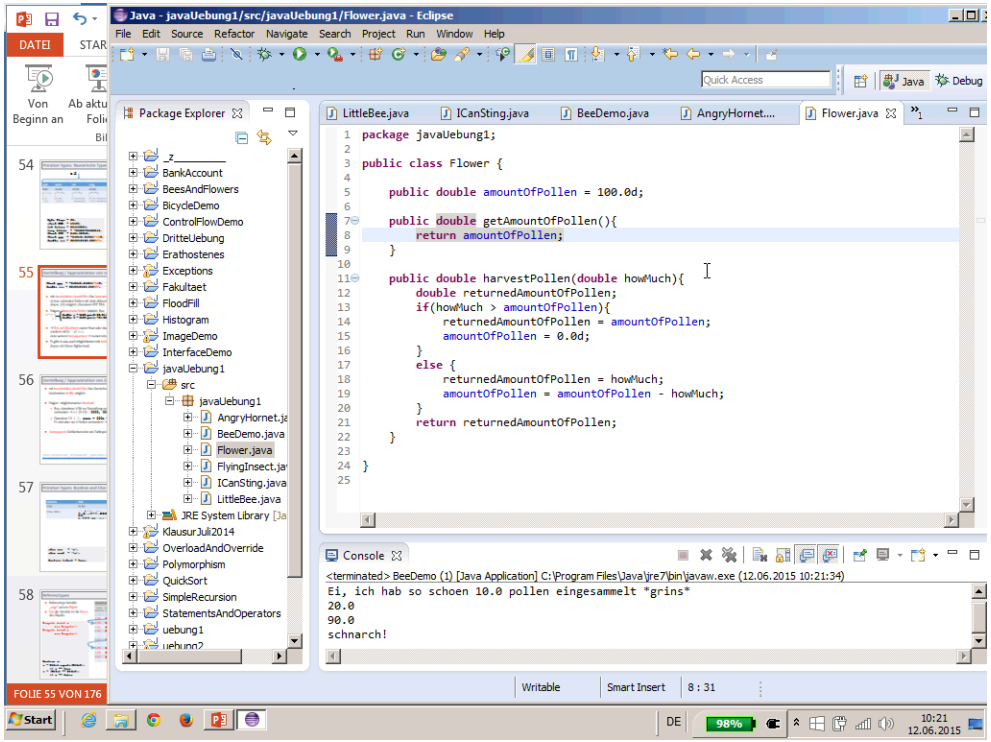

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java ICanSting.java BeeDemo.java AngryHornet.... Flower.java
1 package javaUebung1;
2
3 public class BeeDemo {
4
5 public static void main(String[] args) {
6 Flower wurscht;
7 wurscht = new Flower();
8 Flower weissWurscht = new Flower();
9 LittleBee maja = new LittleBee();
10 LittleBee willi = new LittleBee();
11 maja.collectPollen(wurscht);
12 maja.collectPollen(weissWurscht);
13 System.out.println(maja.collectedPollen);
14 willi.snooze();
15
16 }
17
18 }
19
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:18:20)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
schnarch!
Writable Smart Insert 7 : 17
FOLIE 55 VON 176

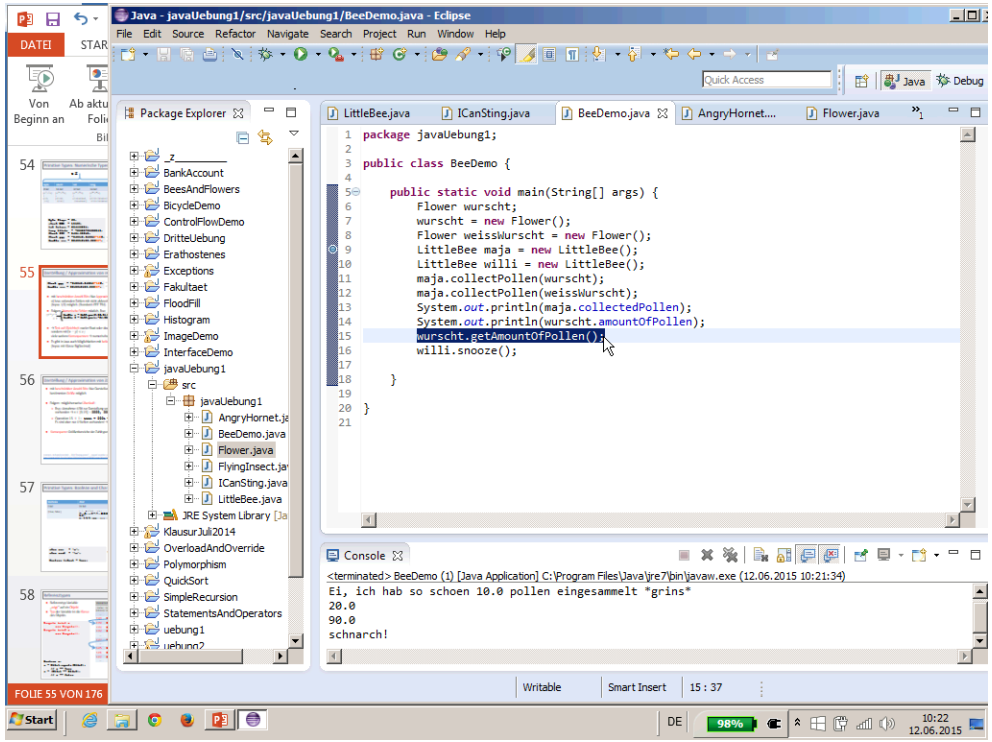

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
*BeeDemo.java
AngryHornet....
Flower.java
1 package javaUebung1;
2
3 public class BeeDemo {
4
5 public static void main(String[] args) {
6 Flower wurscht;
7 wurscht = new Flower();
8 Flower weissWurscht = new Flower();
9 LittleBee maja = new LittleBee();
10 LittleBee willi = new LittleBee();
11 maja.collectPollen(wurscht);
12 maja.collectPollen(weissWurscht);
13 System.out.println(maja.collectedPollen);
14 System.out.println(wurst.collectedPollen);
15
16 willi.snooze();
17
18 }
19
20 }
21
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:18:20)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
schnarch!
Writable
Smart Insert
14 : 49
FOLIE 55 VON 176
10:19
12.06.2015

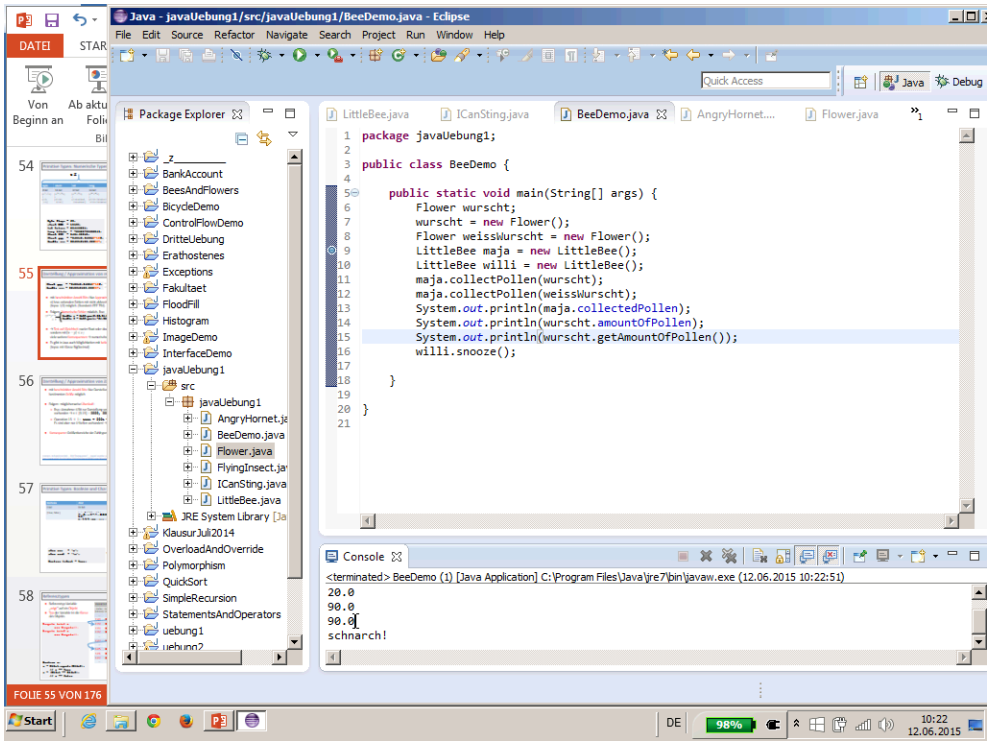

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen = 100.0d;
6
7 public double getAmountOfPollen(){
8 return amountOfPollen;
9 }
10
11 public double harvestPollen(double howMuch){
12 double returnedAmountOfPollen;
13 if(howMuch > amountOfPollen){
14 returnedAmountOfPollen = amountOfPollen;
15 amountOfPollen = 0.0d;
16 }
17 else {
18 returnedAmountOfPollen = howMuch;
19 amountOfPollen = amountOfPollen - howMuch;
20 }
21 return returnedAmountOfPollen;
22 }
23
24 }
25
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:19:36)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
schnarch!
Writable
Smart Insert
5 : 33
FOLIE 55 VON 176
10:19
12.06.2015

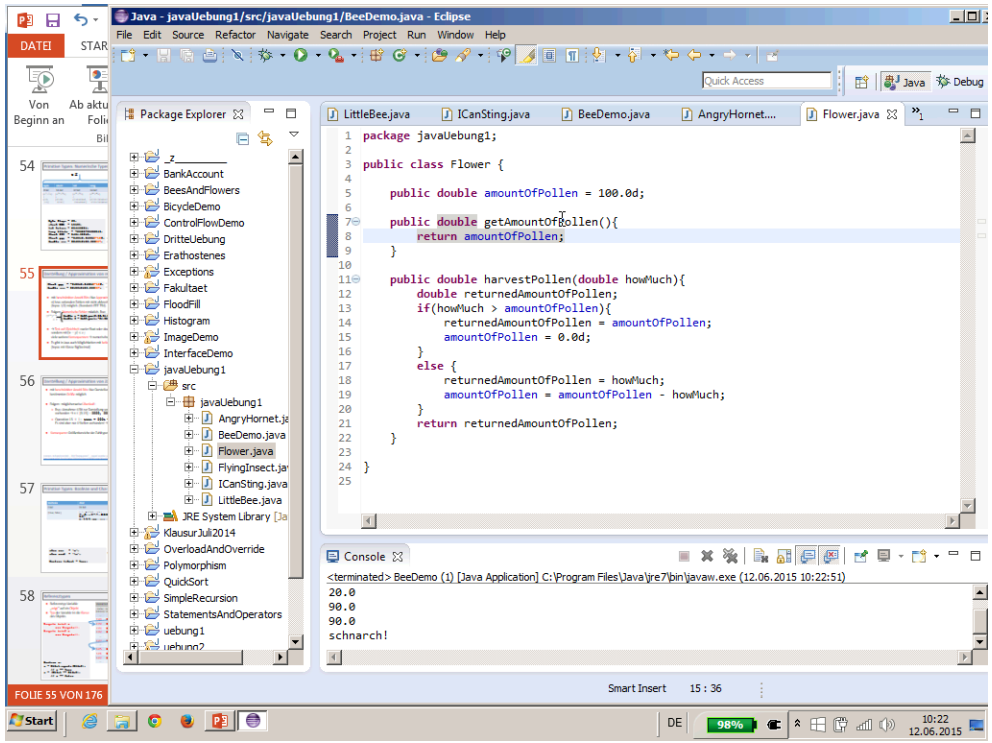

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
1 package javaUebung1;
2
3 public class BeeDemo {
4
5 public static void main(String[] args) {
6 Flower wurscht;
7 wurscht = new Flower();
8 Flower weissWurscht = new Flower();
9 LittleBee maja = new LittleBee();
10 LittleBee willi = new LittleBee();
11 maja.collectPollen(wurscht);
12 maja.collectPollen(weissWurscht);
13 System.out.println(maja.collectedPollen);
14 System.out.println(wurscht.amountOfPollen);
15 willi.snooze();
16
17 }
18
19 }
20
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:19:36)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
schnarch!
Writable
Smart Insert
5 : 33
FOLIE 55 VON 176
10:20
12.06.2015

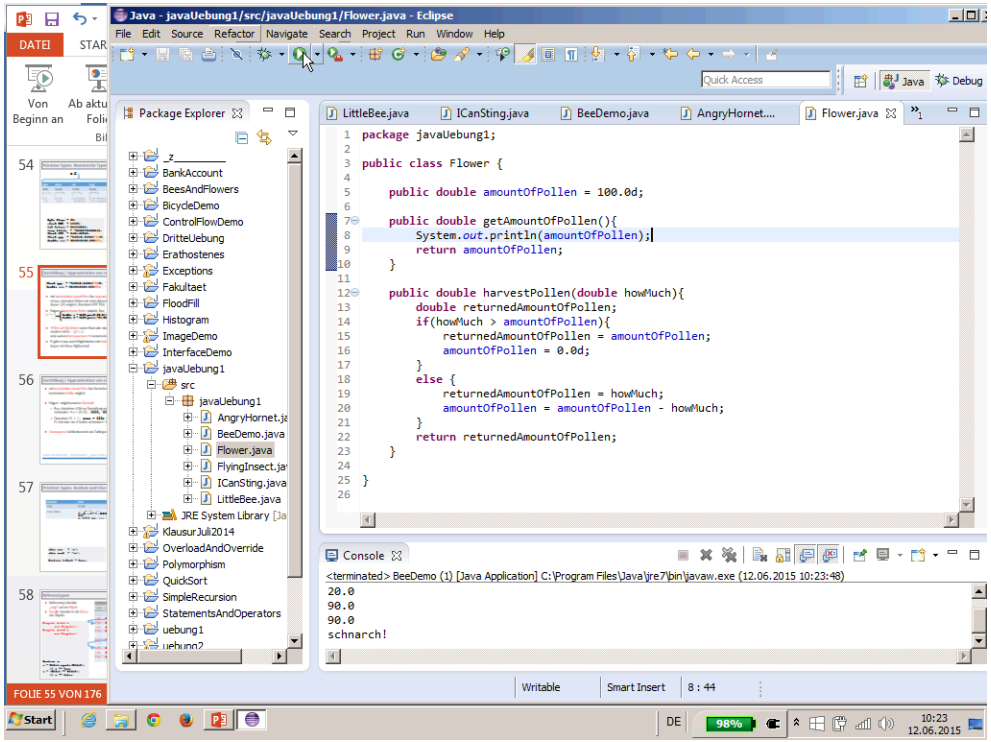

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
*BeeDemo.java
AngryHornet....
Flower.java
1 package javaUebung1;
2
3 public class BeeDemo {
4
5 public static void main(String[] args) {
6 Flower wurscht;
7 wurscht = new Flower();
8 Flower weissWurscht = new Flower();
9 LittleBee maja = new LittleBee();
10 LittleBee willi = new LittleBee();
11 maja.collectPollen(wurscht);
12 maja.collectPollen(weissWurscht);
13 System.out.println(maja.collectedPollen);
14 System.out.println(wurscht.amountOfPollen);
15 wurscht.
16 willi.snooze();
17
18 }
19
20 }
21
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:19:36)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
schnarch!
Writable
Smart Insert
7 : 36
FOLIE 55 VON 176
10:20
12.06.2015

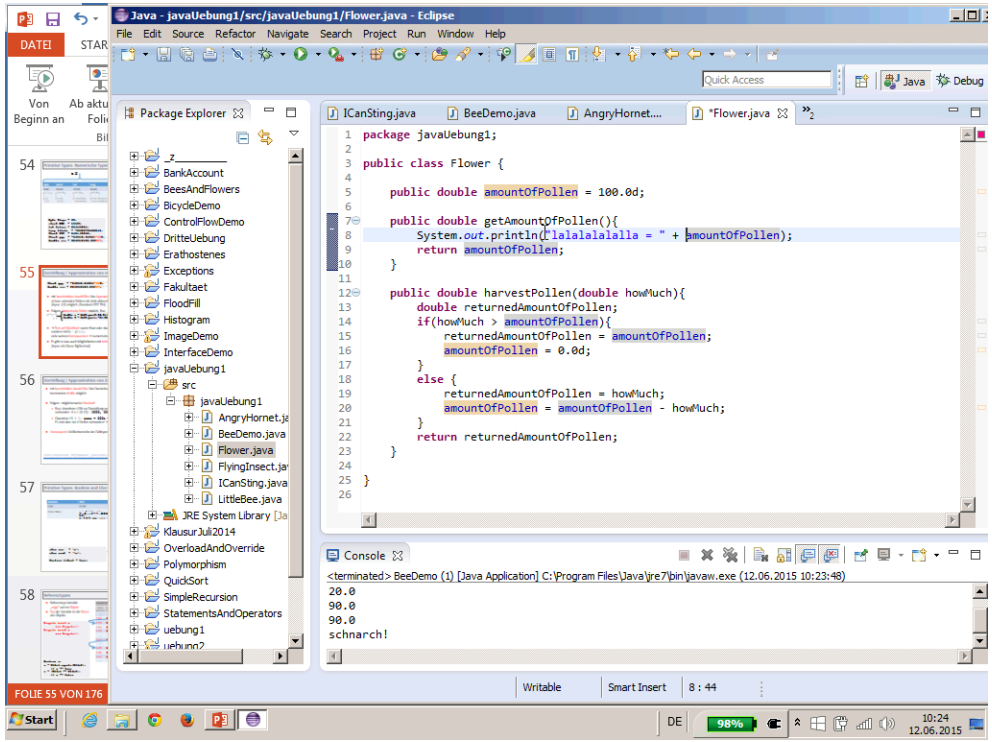

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java ICanSting.java BeeDemo.java AngryHornet.... Flower.java
package javaUebung1;
public class Flower {
public double amountOfPollen = 100.0d;
public double getAmountOfPollen(){
return amountOfPollen;
}
public double harvestPollen(double howMuch){
double returnedAmountOfPollen;
if(howMuch > amountOfPollen){
returnedAmountOfPollen = amountOfPollen;
amountOfPollen = 0.0d;
}
else {
returnedAmountOfPollen = howMuch;
amountOfPollen = amountOfPollen - howMuch;
}
return returnedAmountOfPollen;
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:21:34)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
schnarch!
Writable Smart Insert 8 : 31
FOLIE 55 VON 176
10:21
12.06.2015

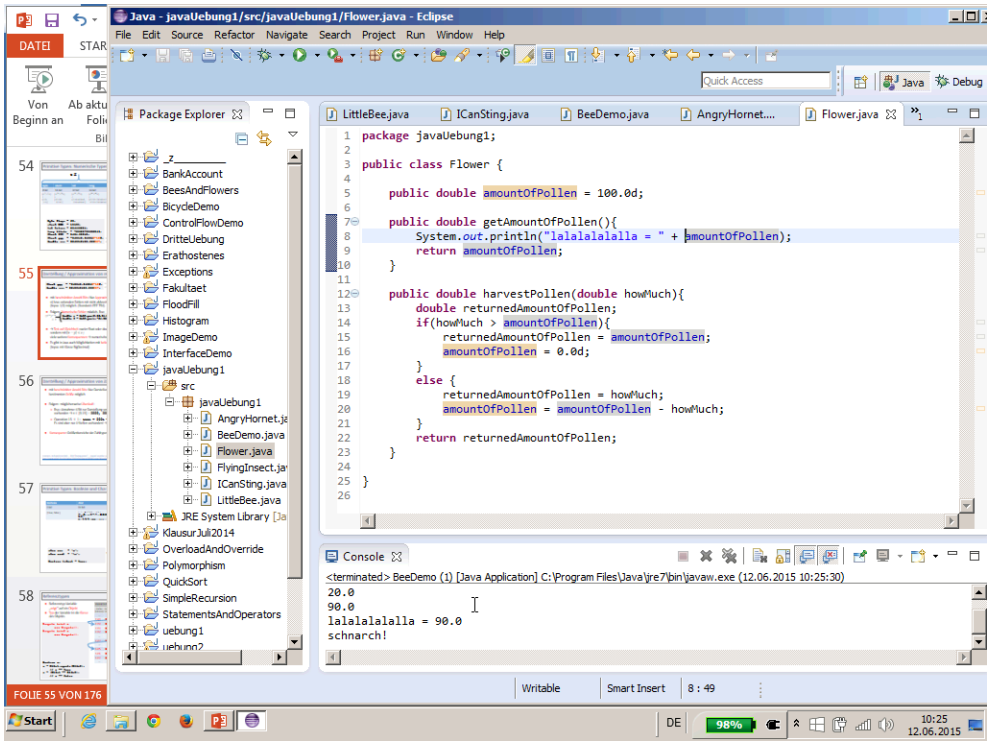

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java ICanSting.java BeeDemo.java AngryHornet.... Flower.java
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
Flower wurscht;
wurscht = new Flower();
Flower weissWurscht = new Flower();
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen(wurscht);
maja.collectPollen(weissWurscht);
System.out.println(maja.collectedPollen);
System.out.println(wurscht.amountOfPollen);
wurscht.getAmountOfPollen();
willi.snooze();
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:21:34)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
schnarch!
Writable Smart Insert 15 : 37
FOLIE 55 VON 176
10:22
12.06.2015

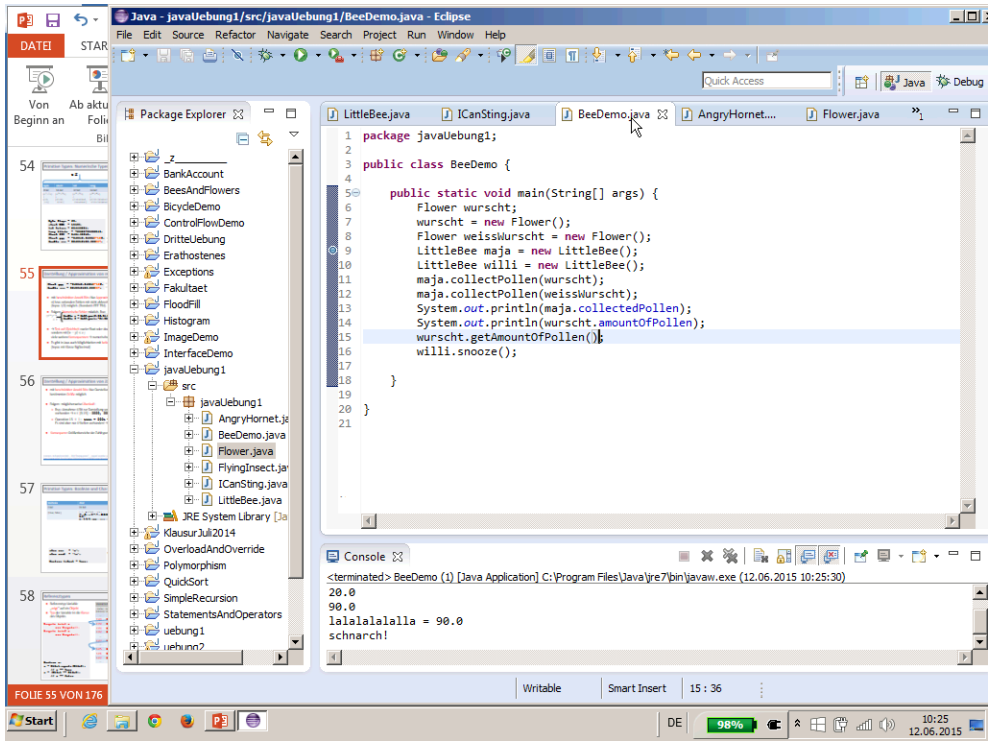

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java ICanSting.java BeeDemo.java AngryHornet.... Flower.java
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
Flower wurscht;
wurscht = new Flower();
Flower weissWurscht = new Flower();
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen(wurscht);
maja.collectPollen(weissWurscht);
System.out.println(maja.collectedPollen);
System.out.println(wurscht.amountOfPollen);
System.out.println(wurscht.getAmountOfPollen());
willi.snooze();
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:22:51)
20.0
90.0
90.0
schnarch!
FOLIE 55 VON 176
10:22
12.06.2015

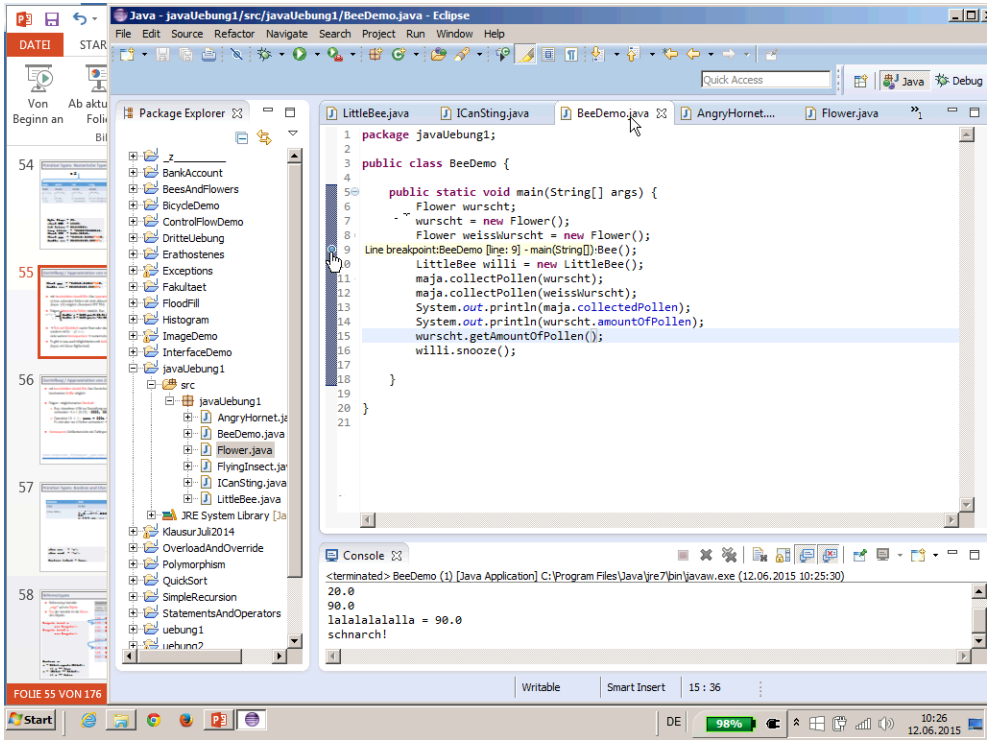

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java ICanSting.java BeeDemo.java AngryHornet.... Flower.java
package javaUebung1;
public class Flower {
public double amountOfPollen = 100.0d;
public double getAmountOfPollen(){
return amountOfPollen;
}
public double harvestPollen(double howMuch){
double returnedAmountOfPollen;
if(howMuch > amountOfPollen){
returnedAmountOfPollen = amountOfPollen;
amountOfPollen = 0.0d;
}
else {
returnedAmountOfPollen = howMuch;
amountOfPollen = amountOfPollen - howMuch;
}
return returnedAmountOfPollen;
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:22:51)
20.0
90.0
90.0
schnarch!
Smart Insert 15 : 36
FOLIE 55 VON 176
10:22
12.06.2015

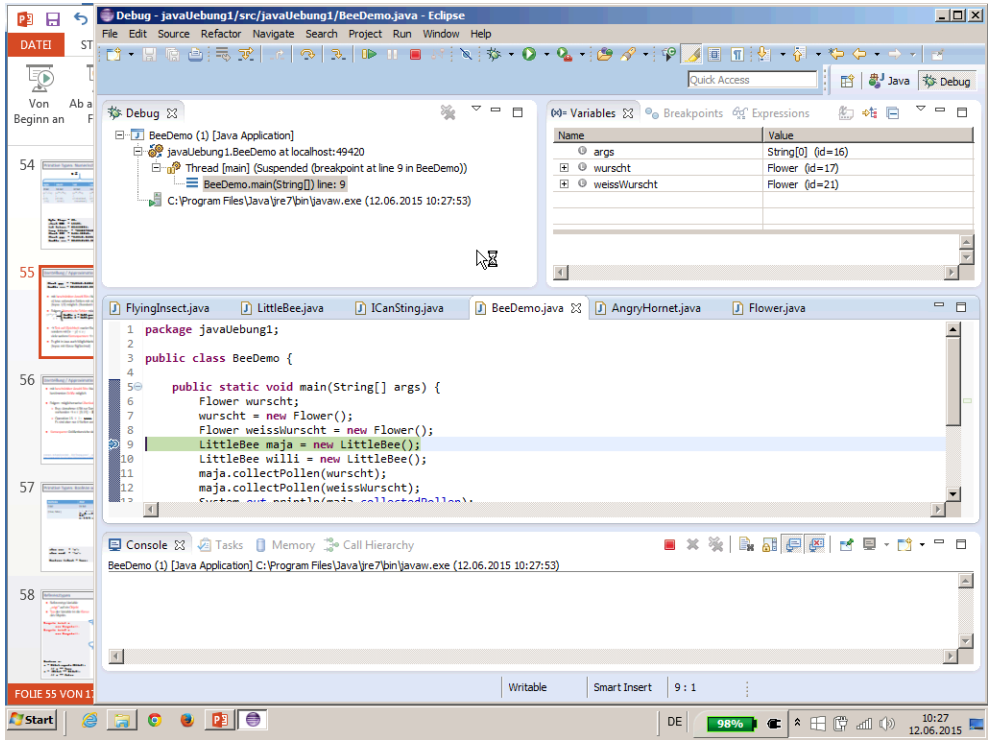

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
package javaUebung1;
public class Flower {
public double amountOfPollen = 100.0d;
public double getAmountOfPollen(){
System.out.println(amountOfPollen);
return amountOfPollen;
}
public double harvestPollen(double howMuch){
double returnedAmountOfPollen;
if(howMuch > amountOfPollen){
returnedAmountOfPollen = amountOfPollen;
amountOfPollen = 0.0d;
}
else {
returnedAmountOfPollen = howMuch;
amountOfPollen = amountOfPollen - howMuch;
}
return returnedAmountOfPollen;
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:23:48)
20.0
90.0
90.0
schnarch!
Writable
Smart Insert
8 : 44
FOLIE 55 VON 176
10:23
12.06.2015

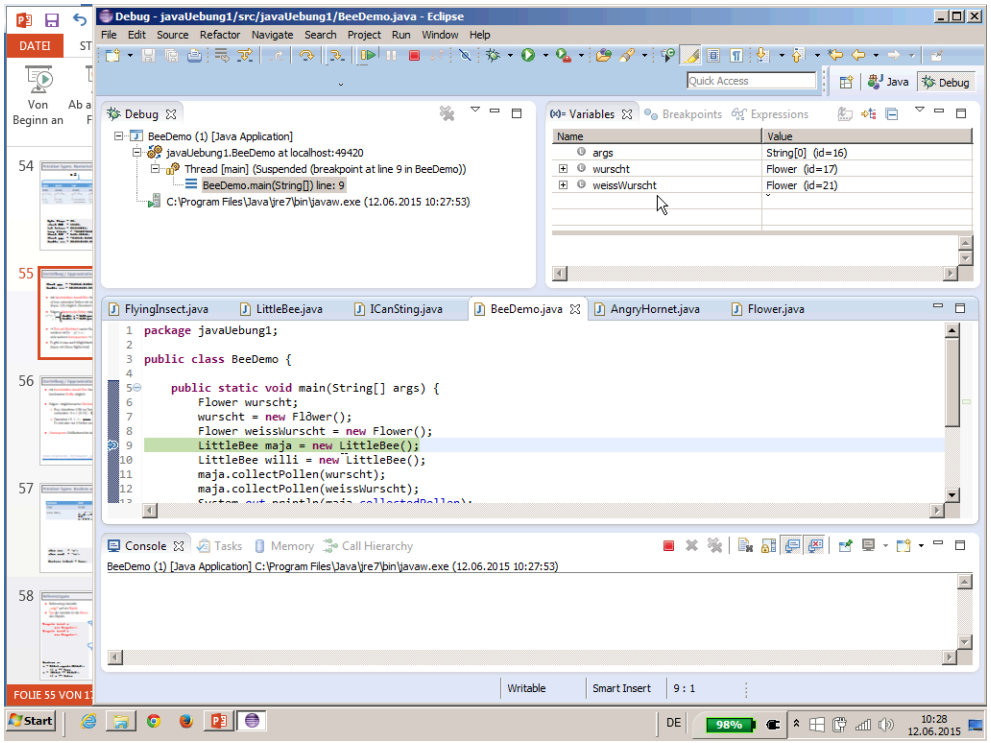

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
*Flower.java
package javaUebung1;
public class Flower {
public double amountOfPollen = 100.0d;
public double getAmountOfPollen(){
System.out.println("lalalalalalla = " + amountOfPollen);
return amountOfPollen;
}
public double harvestPollen(double howMuch){
double returnedAmountOfPollen;
if(howMuch > amountOfPollen){
returnedAmountOfPollen = amountOfPollen;
amountOfPollen = 0.0d;
}
else {
returnedAmountOfPollen = howMuch;
amountOfPollen = amountOfPollen - howMuch;
}
return returnedAmountOfPollen;
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:23:48)
20.0
90.0
90.0
schnarch!
Writable
Smart Insert
8 : 44
FOLIE 55 VON 176
10:24
12.06.2015

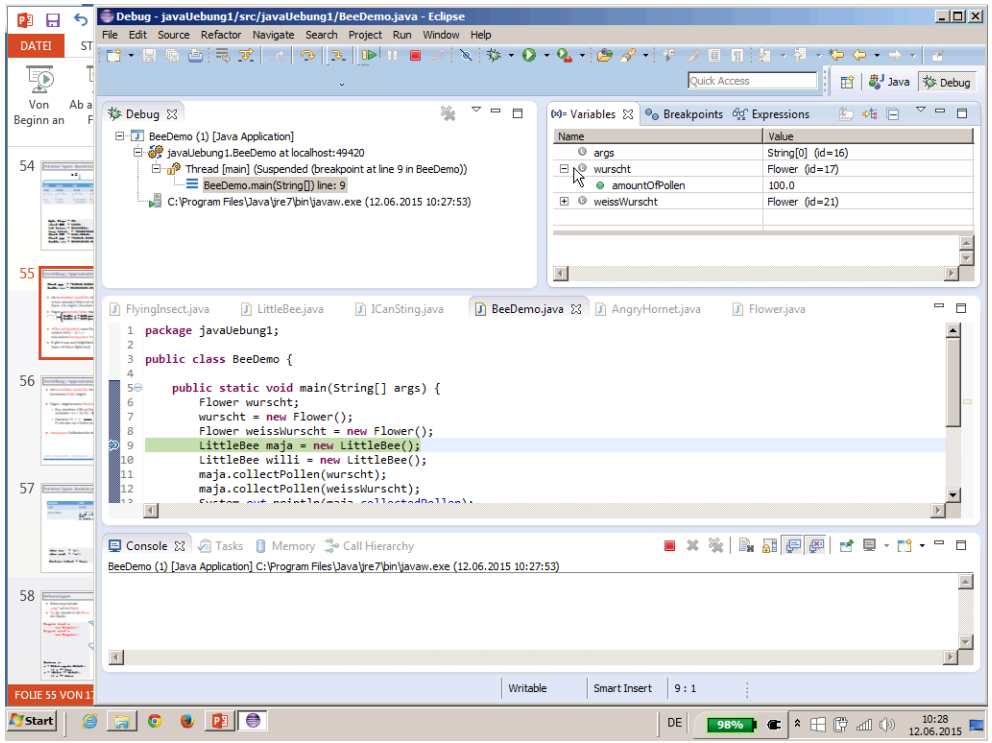

Java - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
package javaUebung1;
public class Flower {
public double amountOfPollen = 100.0d;
public double getAmountOfPollen(){
System.out.println("lalalalalalla = " + amountOfPollen);
return amountOfPollen;
}
public double harvestPollen(double howMuch){
double returnedAmountOfPollen;
if(howMuch > amountOfPollen){
returnedAmountOfPollen = amountOfPollen;
amountOfPollen = 0.0d;
}
else {
returnedAmountOfPollen = howMuch;
amountOfPollen = amountOfPollen - howMuch;
}
return returnedAmountOfPollen;
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:25:30)
20.0
90.0
lalalalalalla = 90.0
schnarch!
Writable
Smart Insert
8 : 49
FOLIE 55 VON 176
10:25
12.06.2015

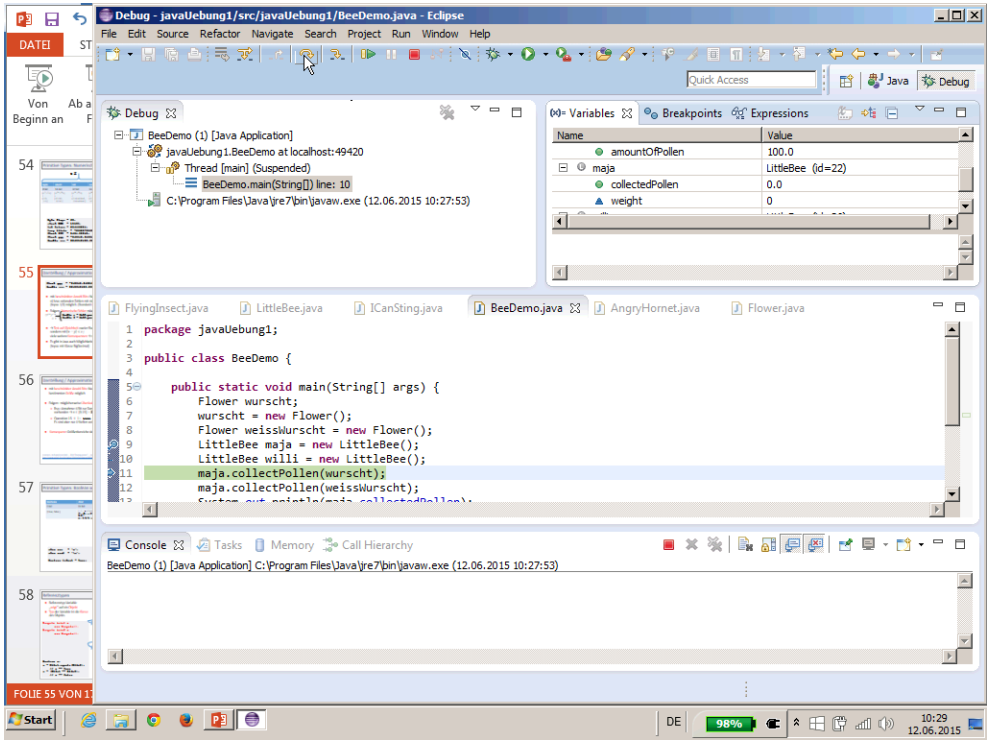

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
Flower wurscht;
wurscht = new Flower();
Flower weissWurscht = new Flower();
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen(wurscht);
maja.collectPollen(weissWurscht);
System.out.println(maja.collectedPollen);
System.out.println(wurscht.amountOfPollen);
wurscht.getAmountOfPollen();
willi.snooze();
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:25:30)
20.0
90.0
lalalalalalla = 90.0
schnarch!
Writable
Smart Insert
15 : 36
FOLIE 55 VON 176
10:25
12.06.2015

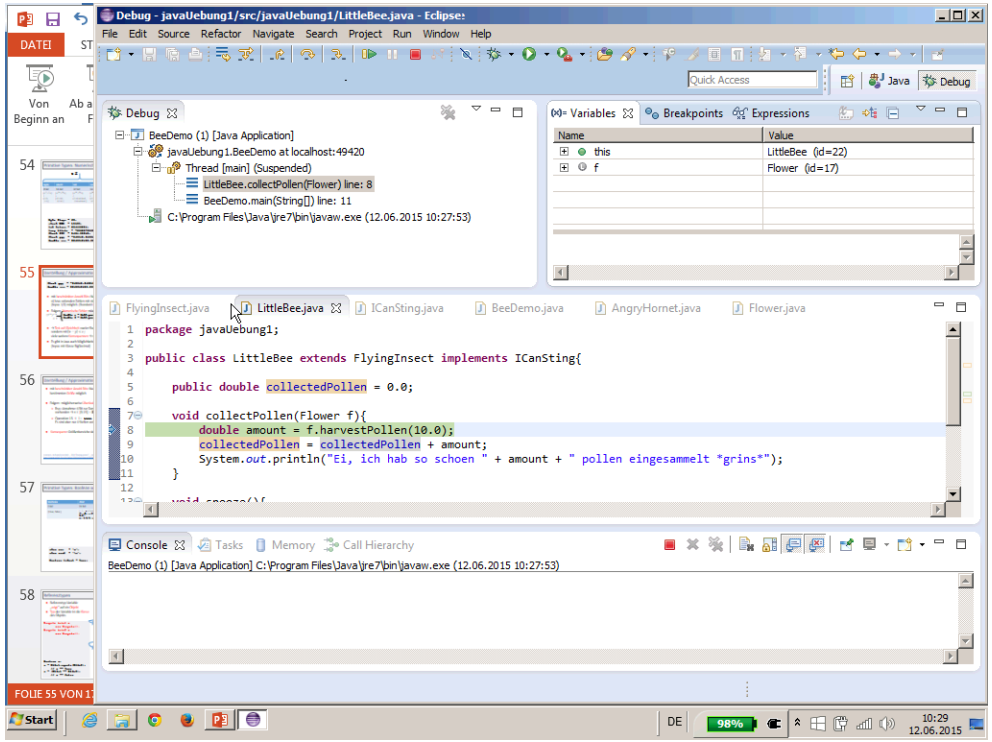

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
1 package javaUebung1;
2
3 public class BeeDemo {
4
5 public static void main(String[] args) {
6 Flower wurscht;
7 wurscht = new Flower();
8 Flower weissWurscht = new Flower();
9 Line breakpoint:BeeDemo [line: 9] - main(String[]):Bee();
10 LittleBee willi = new LittleBee();
11 maja.collectPollen(wurscht);
12 maja.collectPollen(weissWurscht);
13 System.out.println(maja.collectedPollen);
14 System.out.println(wurscht.amountOfPollen);
15 wurscht.getAmountOfPollen();
16 willi.snooze();
17
18 }
19
20 }
21
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:25:30)
20.0
90.0
lalalalalalla = 90.0
schnarch!
Writable
Smart Insert
15 : 36

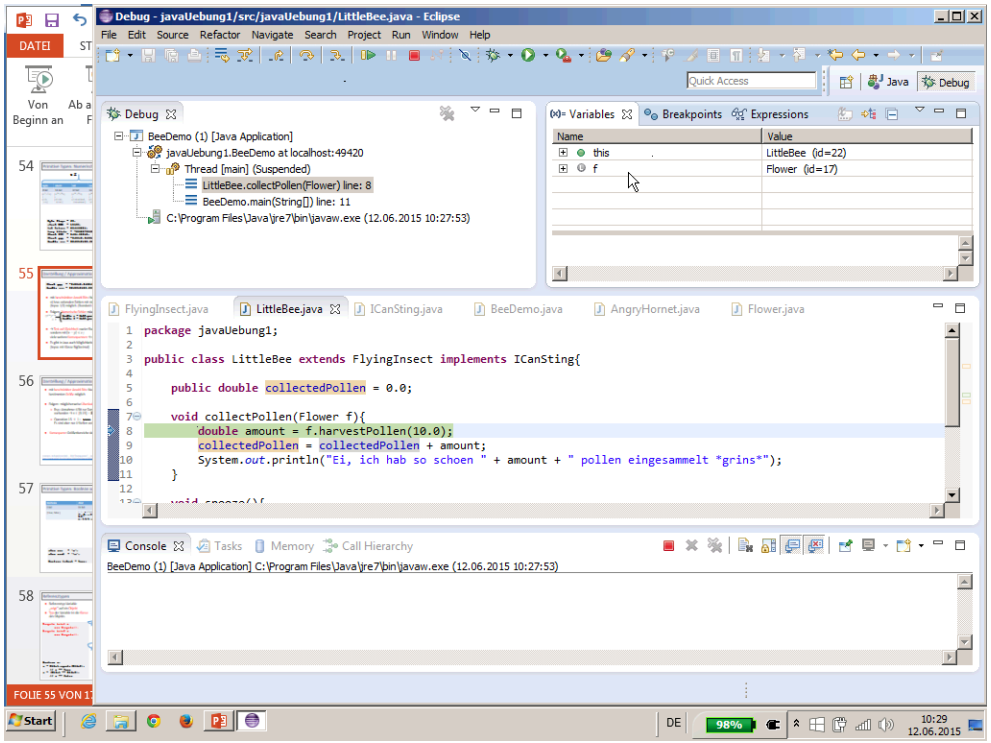

Debug - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
BeeDemo (1) [Java Application]
javaUebung1.BeeDemo at localhost:49420
Thread [main] (Suspended (breakpoint at line 9 in BeeDemo))
BeeDemo.main(String[]) line: 9
C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Variables
Breakpoints
Expressions
Name
Value
args
String[0] (id=16)
wurscht
Flower (id=17)
weissWurscht
Flower (id=21)
FlyingInsect.java
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet.java
Flower.java
1 package javaUebung1;
2
3 public class BeeDemo {
4
5 public static void main(String[] args) {
6 Flower wurscht;
7 wurscht = new Flower();
8 Flower weissWurscht = new Flower();
9 LittleBee maja = new LittleBee();
10 LittleBee willi = new LittleBee();
11 maja.collectPollen(wurscht);
12 maja.collectPollen(weissWurscht);
Console
Tasks
Memory
Call Hierarchy
BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Writable
Smart Insert
9 : 1

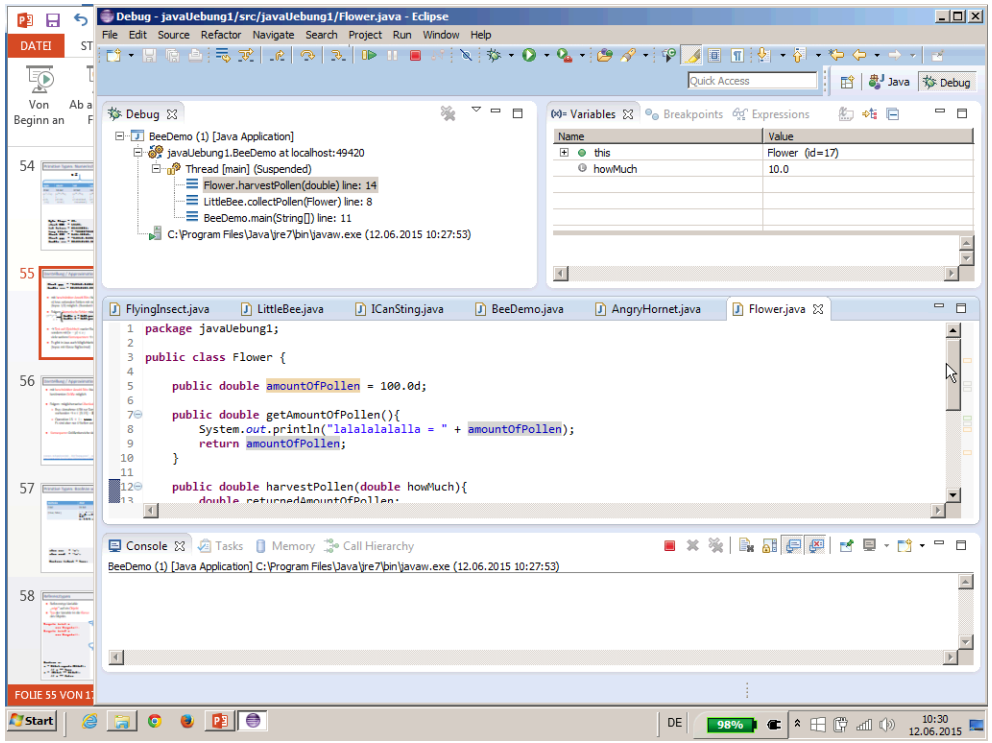

Debug - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
BeeDemo (1) [Java Application]
javaUebung1.BeeDemo at localhost:49420
Thread [main] (Suspended (breakpoint at line 9 in BeeDemo))
BeeDemo.main(String[]) line: 9
C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Variables
Breakpoints
Expressions
Name
Value
args
String[0] (id=16)
wurscht
Flower (id=17)
weissWurscht
Flower (id=21)
FlyingInsect.java
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet.java
Flower.java
1 package javaUebung1;
2
3 public class BeeDemo {
4
5 public static void main(String[] args) {
6 Flower wurscht;
7 wurscht = new Flower();
8 Flower weissWurscht = new Flower();
9 LittleBee maja = new LittleBee();
10 LittleBee willi = new LittleBee();
11 maja.collectPollen(wurscht);
12 maja.collectPollen(weissWurscht);
Console
Tasks
Memory
Call Hierarchy
BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Writable
Smart Insert
9 : 1

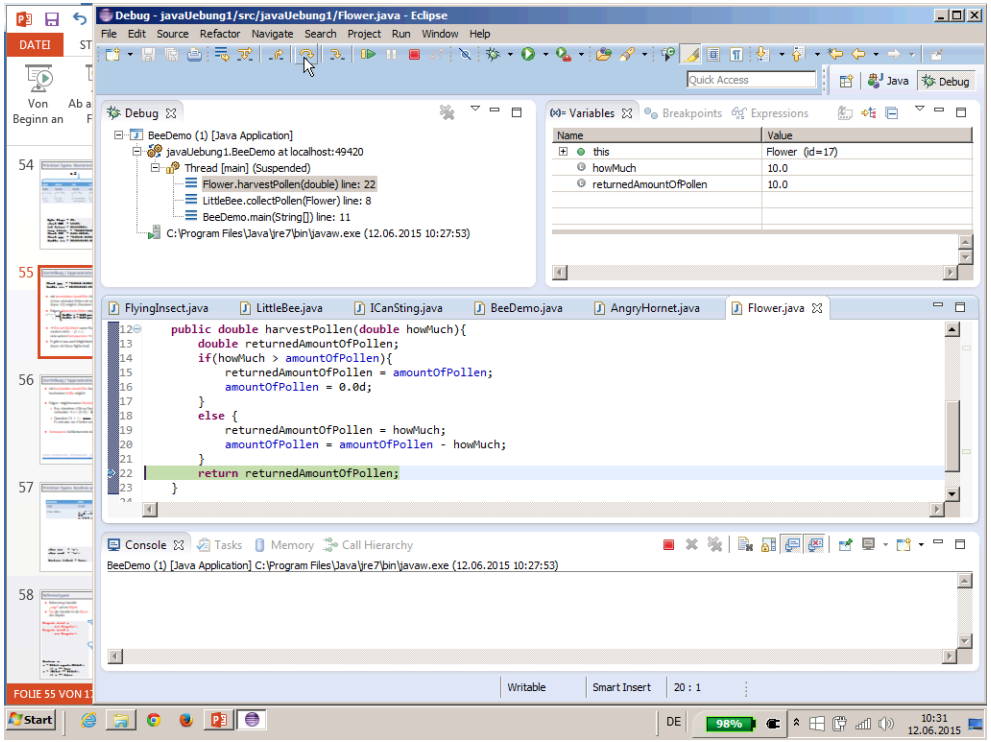

Debug - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
BeeDemo (1) [Java Application]
javaUebung1.BeeDemo at localhost:49420
Thread [main] (Suspended (breakpoint at line 9 in BeeDemo))
BeeDemo.main(String[]) line: 9
C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Variables
Breakpoints
Expressions
Name
Value
args
String[0] (id=16)
wurscht
Flower (id=17)
amountOfPollen
100.0
weissWurscht
Flower (id=21)
FlyingInsect.java
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet.java
Flower.java
1 package javaUebung1;
2
3 public class BeeDemo {
4
5 public static void main(String[] args) {
6 Flower wurscht;
7 wurscht = new Flower();
8 Flower weissWurscht = new Flower();
9 LittleBee maja = new LittleBee();
10 LittleBee willi = new LittleBee();
11 maja.collectPollen(wurscht);
12 maja.collectPollen(weissWurscht);
Console
Tasks
Memory
Call Hierarchy
BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Writable
Smart Insert
9 : 1

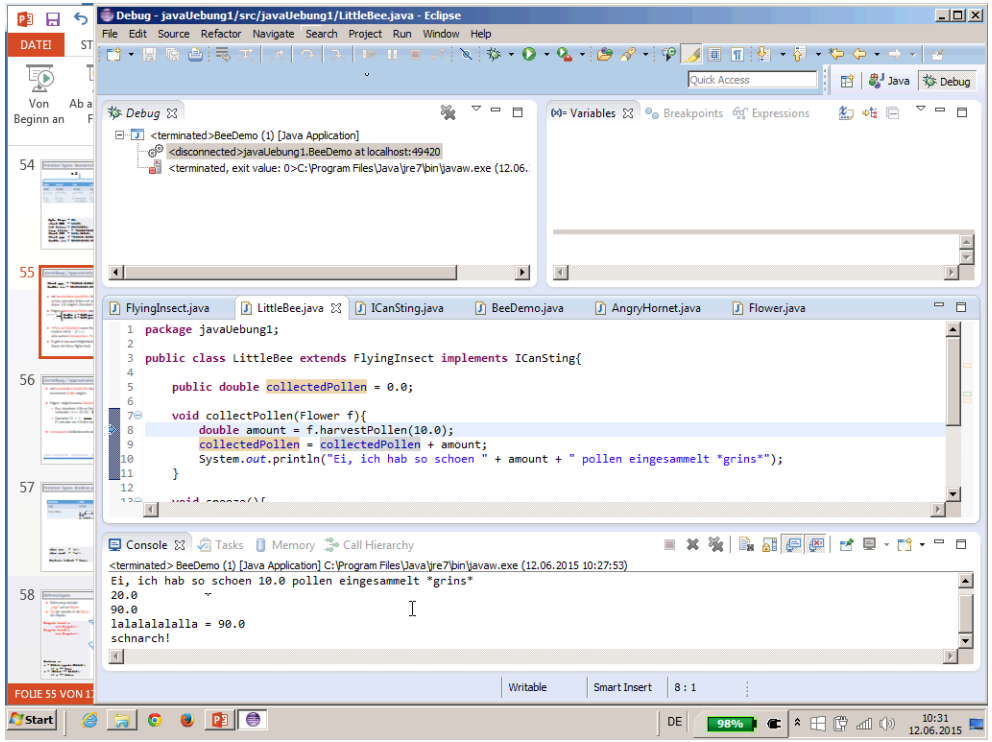

Debug - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
BeeDemo (1) [Java Application]
javaUebung1.BeeDemo at localhost:49420
Thread [main] (Suspended)
BeeDemo.main(String[]) line: 10
C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
amountOfPollen 100.0
maja LittleBee (id=22)
collectedPollen 0.0
weight 0
FlyingInsect.java
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet.java
Flower.java
1 package javaUebung1;
2
3 public class BeeDemo {
4
5 public static void main(String[] args) {
6 Flower wurscht;
7 wurscht = new Flower();
8 Flower weisswurscht = new Flower();
9 LittleBee maja = new LittleBee();
10 LittleBee willi = new LittleBee();
11 maja.collectPollen(wurscht);
12 maja.collectPollen(weisswurscht);
Console
Tasks
Memory
Call Hierarchy
BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)

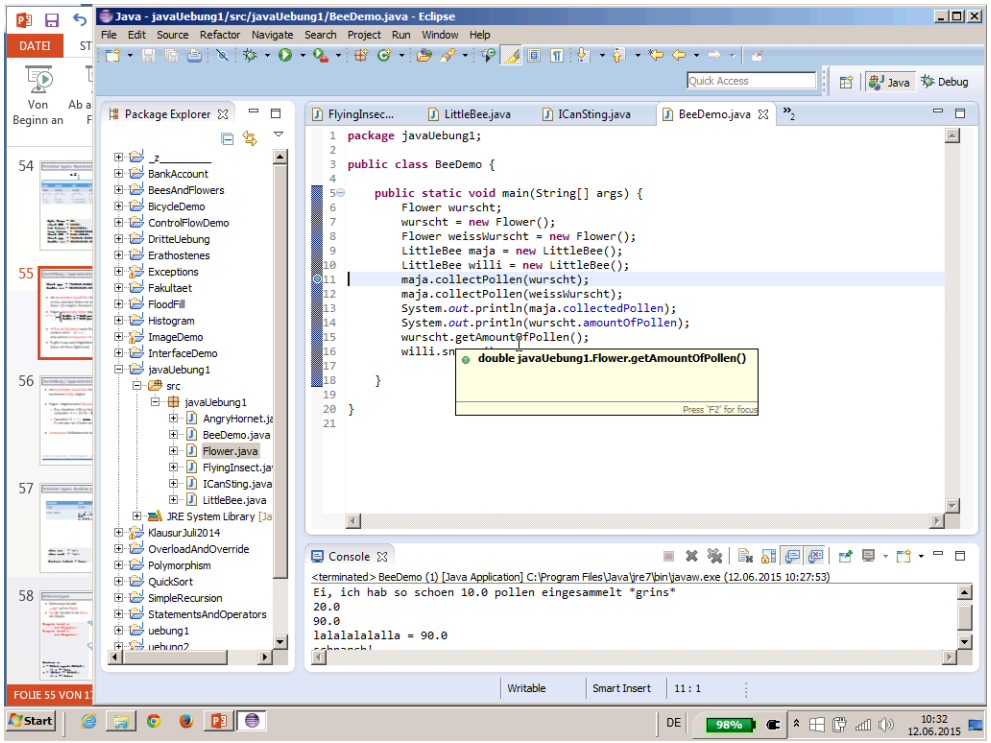

Debug - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
BeeDemo (1) [Java Application]
javaUebung1.BeeDemo at localhost:49420
Thread [main] (Suspended)
LittleBee.collectPollen(Flower) line: 8
BeeDemo.main(String[]) line: 11
C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
this LittleBee (id=22)
f Flower (id=17)
FlyingInsect.java
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet.java
Flower.java
1 package javaUebung1;
2
3 public class LittleBee extends FlyingInsect implements ICanSting{
4
5 public double collectedPollen = 0.0;
6
7 void collectPollen(Flower f){
8 double amount = f.harvestPollen(10.0);
9 collectedPollen = collectedPollen + amount;
10 System.out.println("Ei, ich hab so schoen " + amount + " pollen eingesammelt *grins*");
11 }
Console
Tasks
Memory
Call Hierarchy
BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)

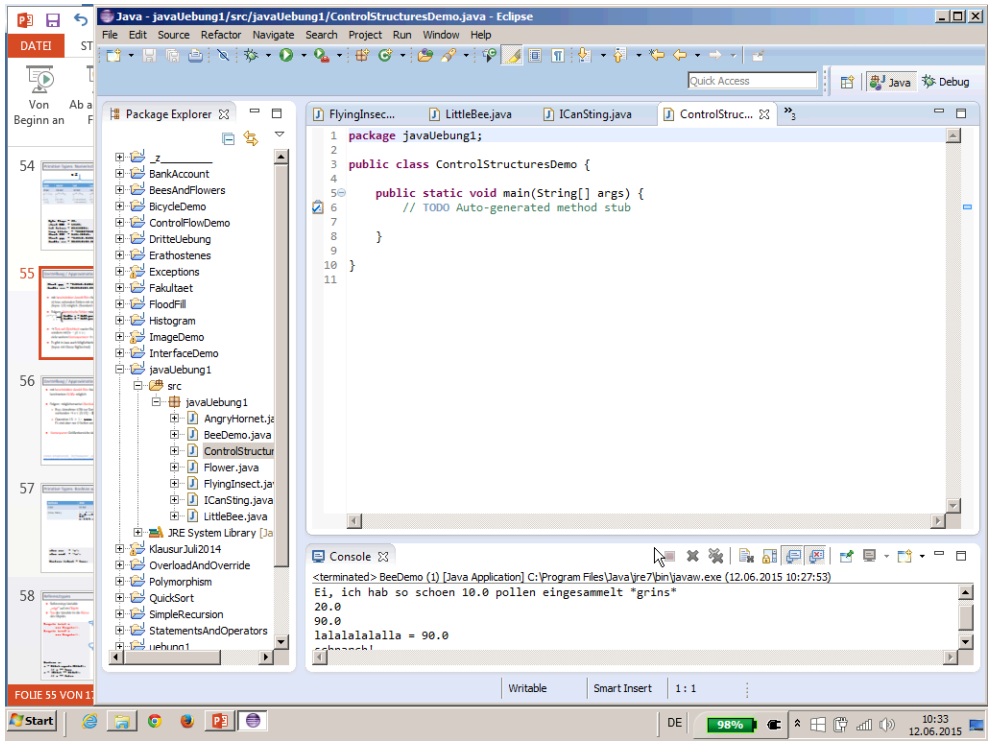

Debug - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
BeeDemo (1) [Java Application]
javaUebung1.BeeDemo at localhost:49420
Thread [main] (Suspended)
LittleBee.collectPollen(Flower) line: 8
BeeDemo.main(String[]) line: 11
C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
this LittleBee (id=22)
f Flower (id=17)
FlyingInsect.java
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet.java
Flower.java
1 package javaUebung1;
2
3 public class LittleBee extends FlyingInsect implements ICanSting{
4
5 public double collectedPollen = 0.0;
6
7 void collectPollen(Flower f){
8 double amount = f.harvestPollen(10.0);
9 collectedPollen = collectedPollen + amount;
10 System.out.println("Ei, ich hab so schoen " + amount + " pollen eingesammelt *grins*");
11 }
Console
Tasks
Memory
Call Hierarchy
BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)

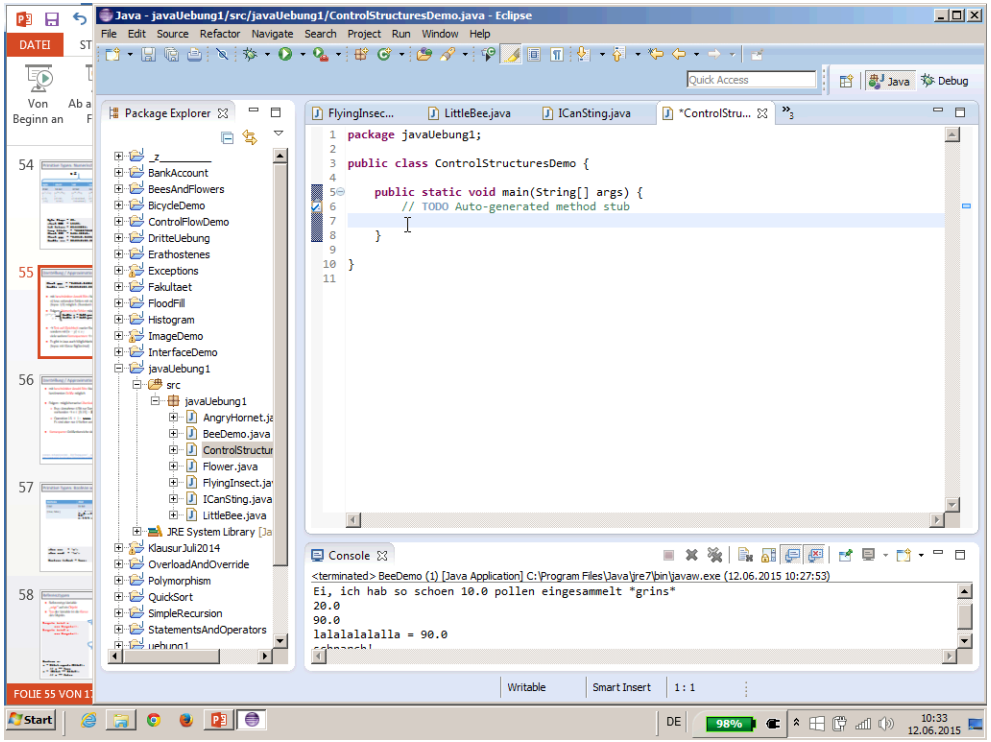

Debug - javaUebung1/src/javaUebung1/Flower.java - Eclipse
BeeDemo (1) [Java Application]
javaUebung1.BeeDemo at localhost:49420
Thread [main] (Suspended)
Flower.harvestPollen(double) line: 14
LittleBee.collectPollen(Flower) line: 8
BeeDemo.main(String[]) line: 11
C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
this Flower (id=17)
howMuch 10.0
FlyingInsect.java
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet.java
Flower.java
1 package javaUebung1;
2
3 public class Flower {
4
5 public double amountOfPollen = 100.0d;
6
7 public double getAmountOfPollen(){
8 System.out.println("lalalalalalla = " + amountOfPollen);
9 return amountOfPollen;
10 }
11
12 public double harvestPollen(double howMuch){
Console
Tasks
Memory
Call Hierarchy
BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)

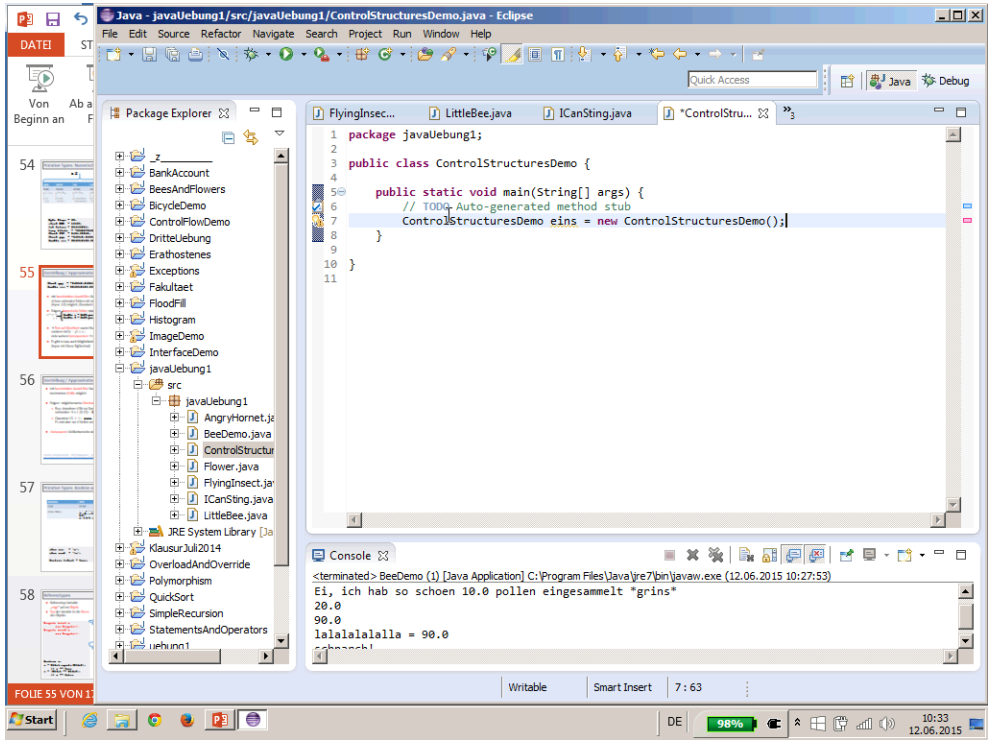

Debug - javaUebung1/src/javaUebung1/Flower.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
BeeDemo (1) [Java Application]
javaUebung1.BeeDemo at localhost:49420
Thread [main] (Suspended)
Flower.harvestPollen(double) line: 22
LittleBee.collectPollen(Flower) line: 8
BeeDemo.main(String[]) line: 11
C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Variables
Breakpoints
Expressions
Name
Value
this
Flower (id=17)
howMuch
10.0
returnedAmountOfPollen
10.0
FlyingInsect.java
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet.java
Flower.java
public double harvestPollen(double howMuch){
double returnedAmountOfPollen;
if(howMuch > amountOfPollen){
returnedAmountOfPollen = amountOfPollen;
amountOfPollen = 0.0d;
}
else {
returnedAmountOfPollen = howMuch;
amountOfPollen = amountOfPollen - howMuch;
}
return returnedAmountOfPollen;
}
Console
Tasks
Memory
Call Hierarchy
BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Writable
Smart Insert
20 : 1

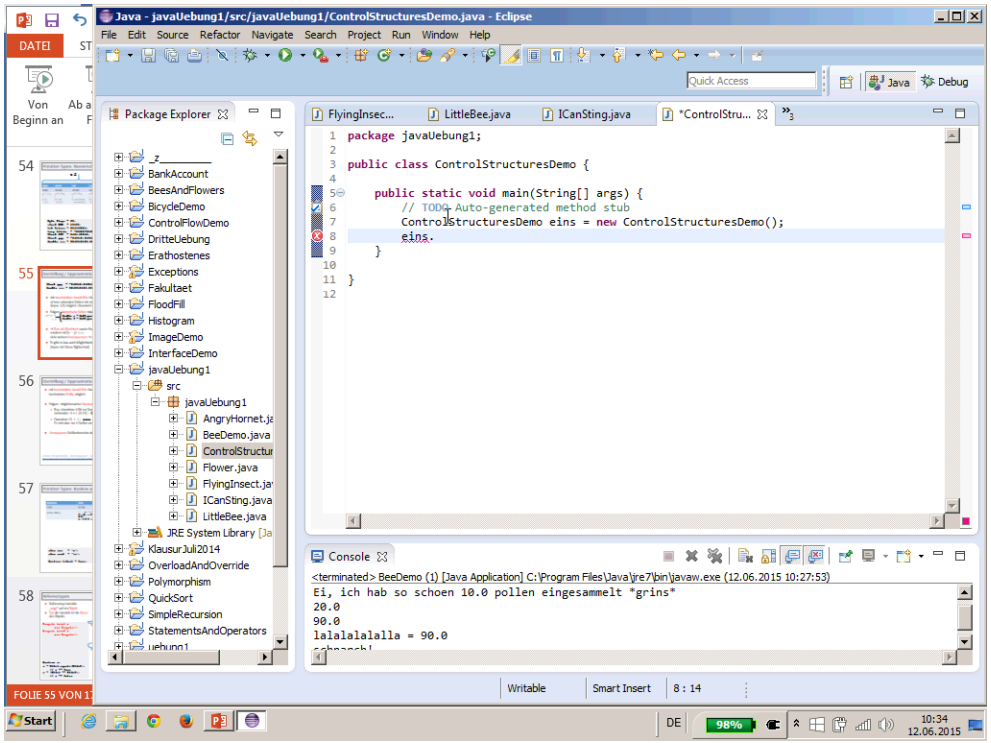

Debug - javaUebung1/src/javaUebung1/LittleBee.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
<terminated>BeeDemo (1) [Java Application]
<disconnected>javaUebung1.BeeDemo at localhost:49420
<terminated, exit value: 0>C:\Program Files\Java\jre7\bin\javaw.exe (12.06.
Variables
Breakpoints
Expressions
FlyingInsect.java
LittleBee.java
ICanSting.java
BeeDemo.java
AngryHornet.java
Flower.java
package javaUebung1;
public class LittleBee extends FlyingInsect implements ICanSting{
public double collectedPollen = 0.0;
void collectPollen(Flower f){
double amount = f.harvestPollen(10.0);
collectedPollen = collectedPollen + amount;
System.out.println("Ei, ich hab so schoen " + amount + " pollen eingesammelt *grins*");
}
Console
Tasks
Memory
Call Hierarchy
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
lalalalalalla = 90.0
schnarch!
Writable
Smart Insert
8 : 1

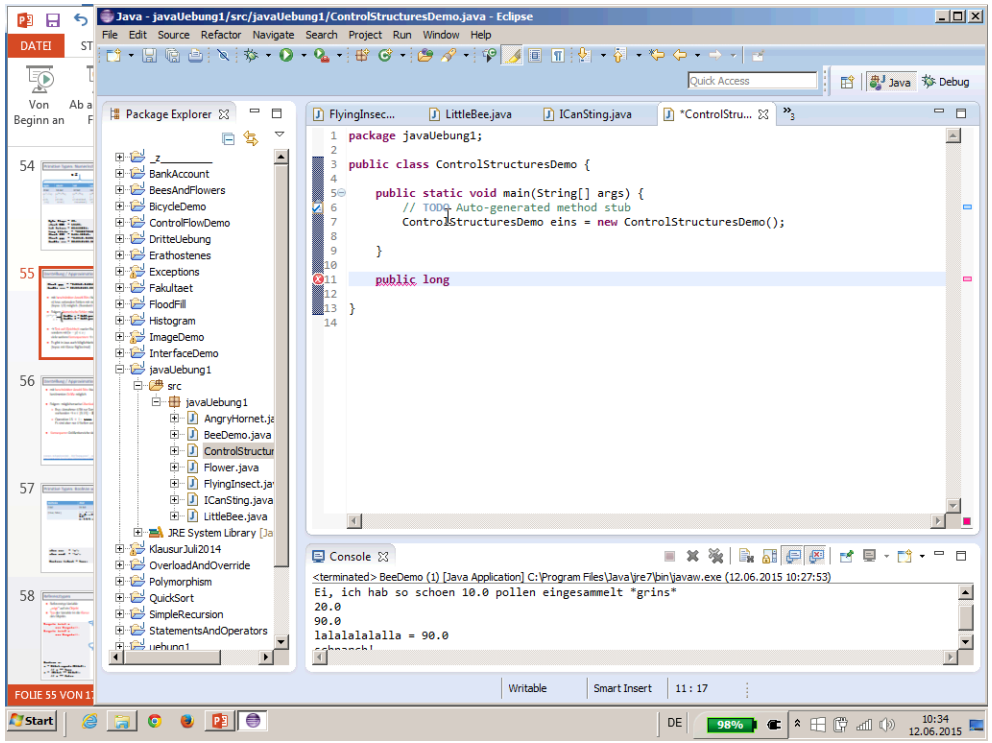

Java - javaUebung1/src/javaUebung1/BeeDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
src
javaUebung1
AngryHornet.ja
BeeDemo.java
Flower.java
FlyingInsect.ja
ICanSting.java
LittleBee.java
JRE System Library [Ja
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
FlyingInsec...
LittleBee.java
ICanSting.java
BeeDemo.java
package javaUebung1;
public class BeeDemo {
public static void main(String[] args) {
Flower wurscht;
wurscht = new Flower();
Flower weissWurscht = new Flower();
LittleBee maja = new LittleBee();
LittleBee willi = new LittleBee();
maja.collectPollen(wurscht);
maja.collectPollen(weissWurscht);
System.out.println(maja.collectedPollen);
System.out.println(wurscht.amountOfPollen);
wurscht.getAmountOfPollen();
willi.sn
double javaUebung1.Flower.getAmountOfPollen()
Press 'F2' for focus
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
lalalalalalla = 90.0
Writable
Smart Insert
11 : 1

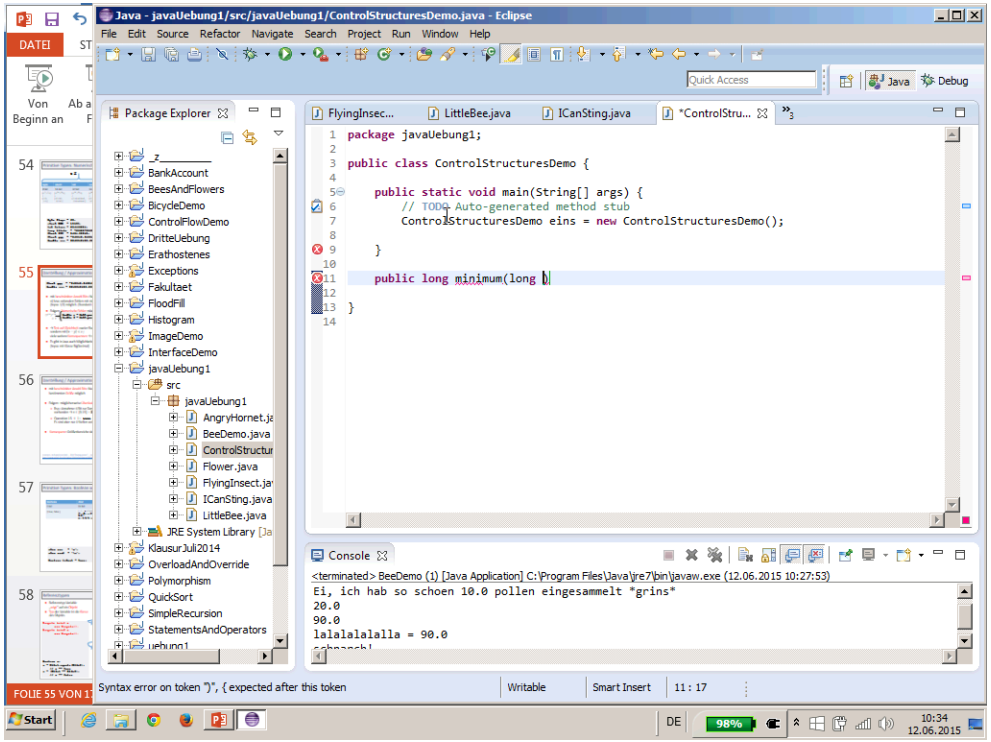

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
src
javaUebung1
AngryHornet.ja
BeeDemo.java
ControlStructur
Flower.java
FlyingInsect.ja
ICanSting.java
LittleBee.java
JRE System Library [Ja
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
FlyingInsec...
LittleBee.java
ICanSting.java
ControlStruc...
package javaUebung1;
public class ControlStructuresDemo {
public static void main(String[] args) {
// TODO Auto-generated method stub
}
}
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
lalalalalalla = 90.0
Writable
Smart Insert
1 : 1

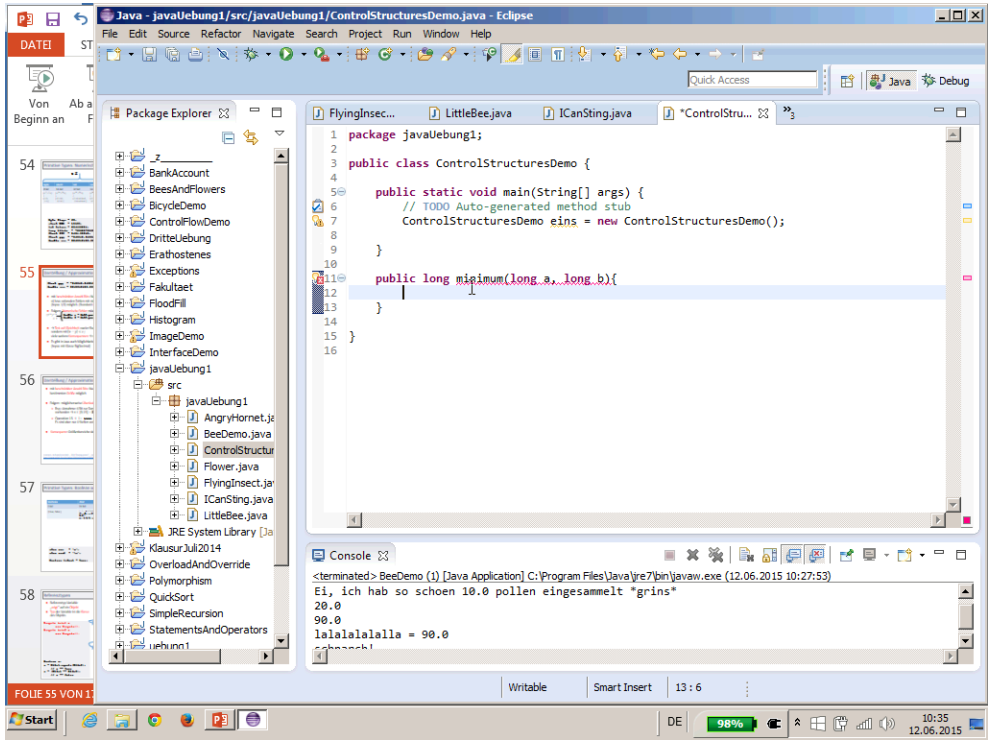

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7
8 }
9
10 }
11
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
lalalalalalla = 90.0
Writable Smart Insert 1 : 1

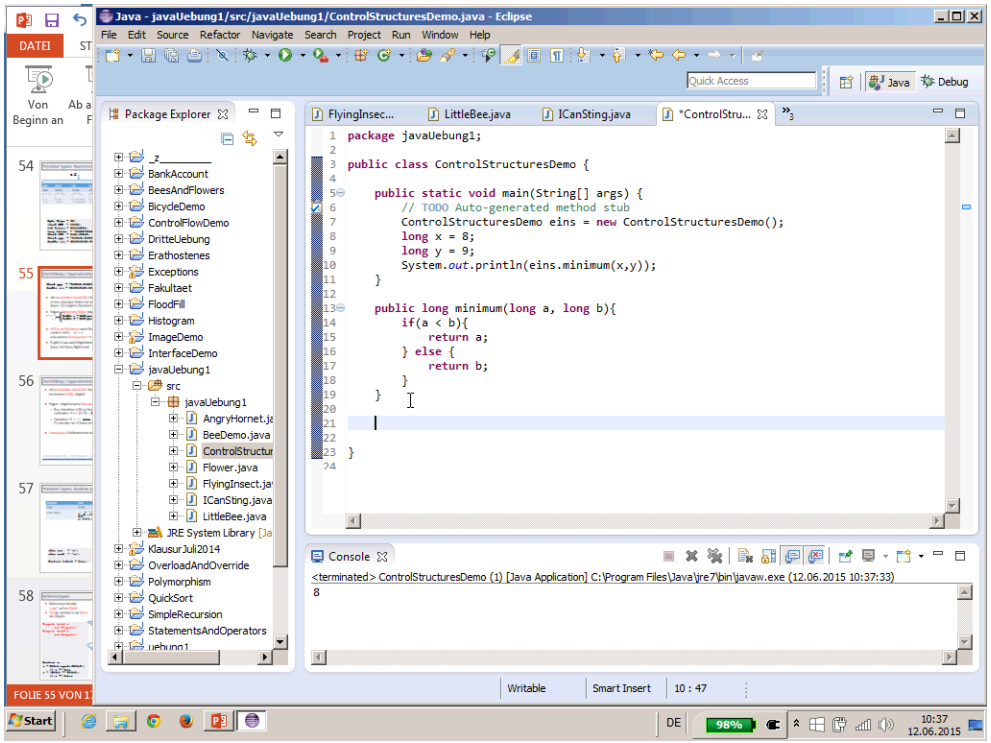

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 }
9
10 }
11
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
lalalalalalla = 90.0
Writable Smart Insert 7 : 63

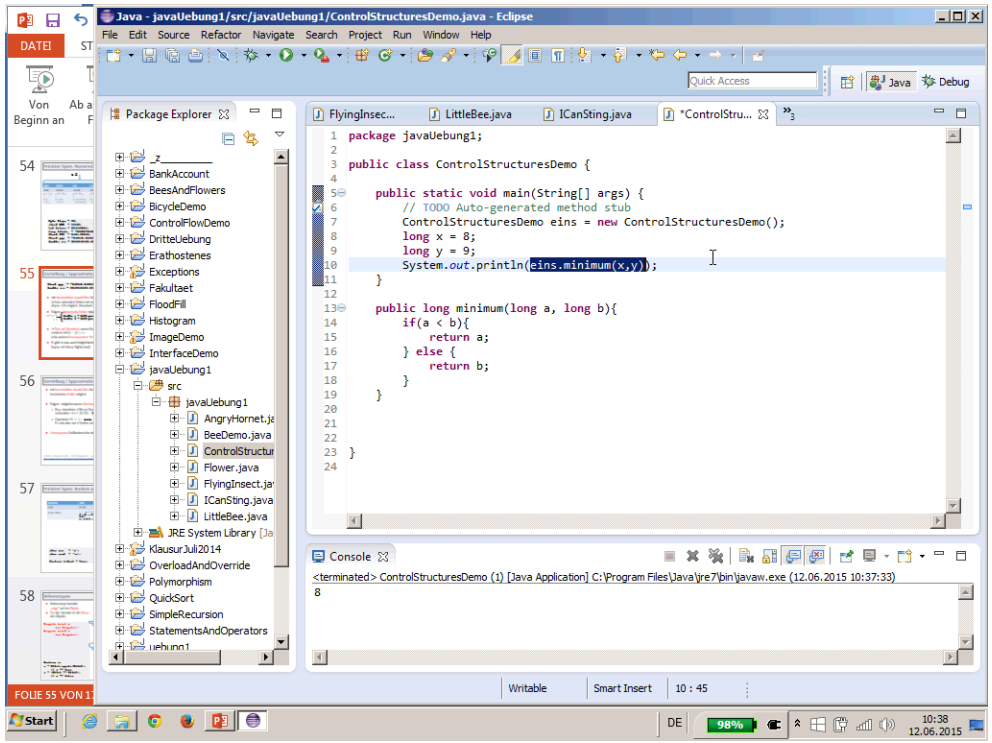

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 eins.
9 }
10
11 }
12
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
lalalalalalla = 90.0
Writable Smart Insert 8 : 14

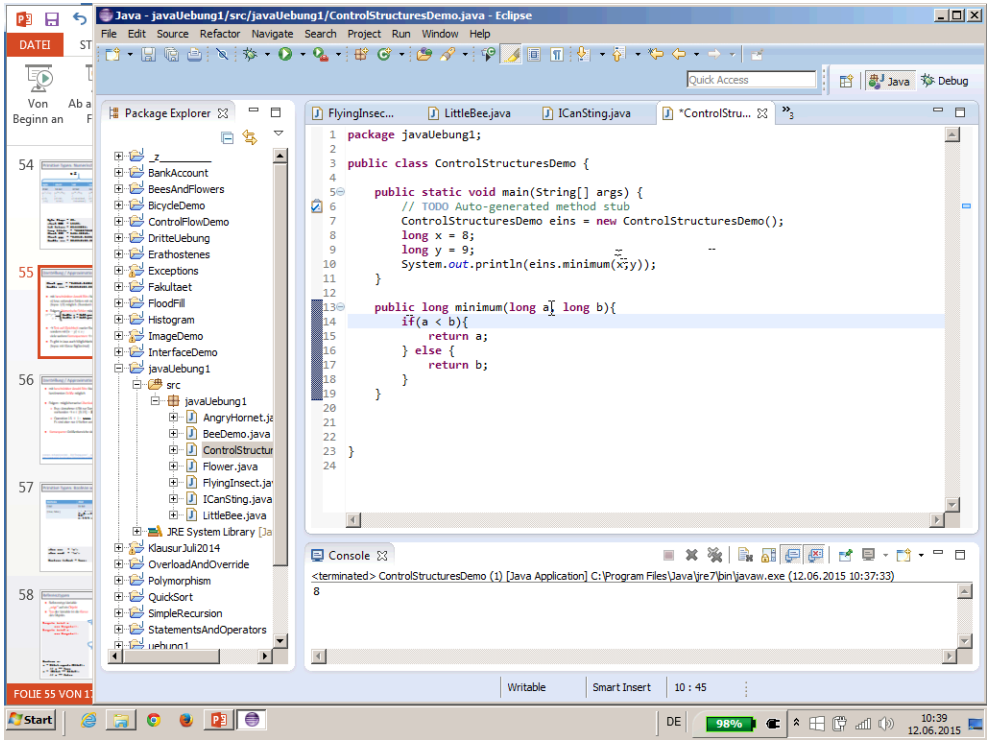

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8
9 }
10
11 public long
12
13 }
14
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
lalalalalalla = 90.0
Writable Smart Insert 11 : 17

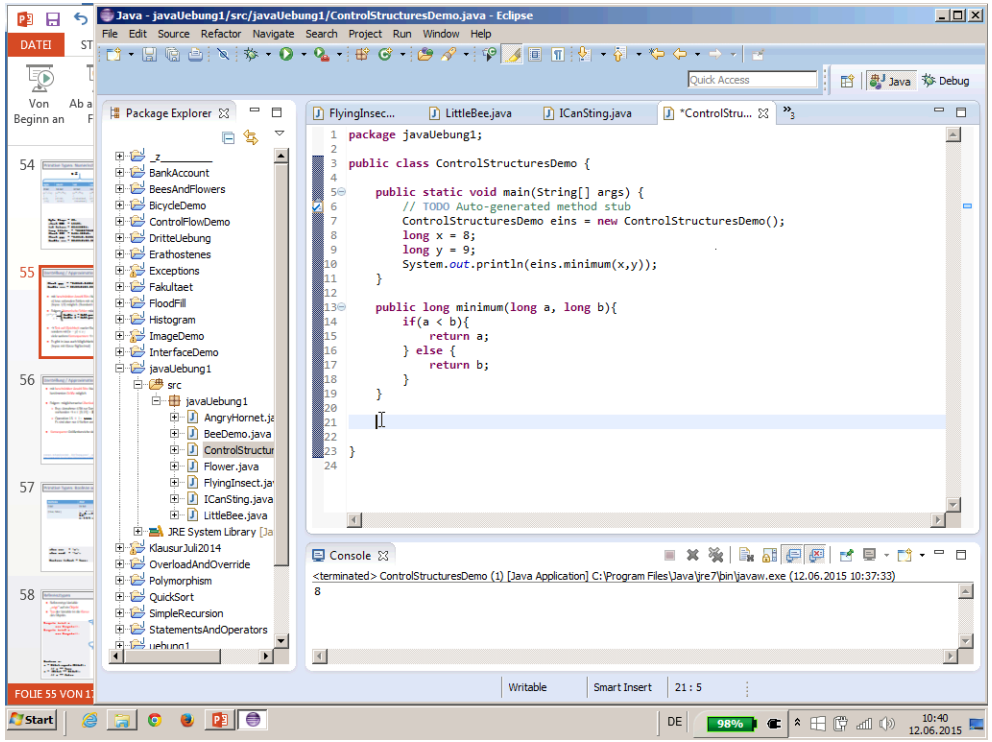

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8
9 }
10
11 public long minimum(long b
12
13 }
14
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
lalalalalalla = 90.0
Syntax error on token ")", { expected after this token
Writable
Smart Insert
11 : 17

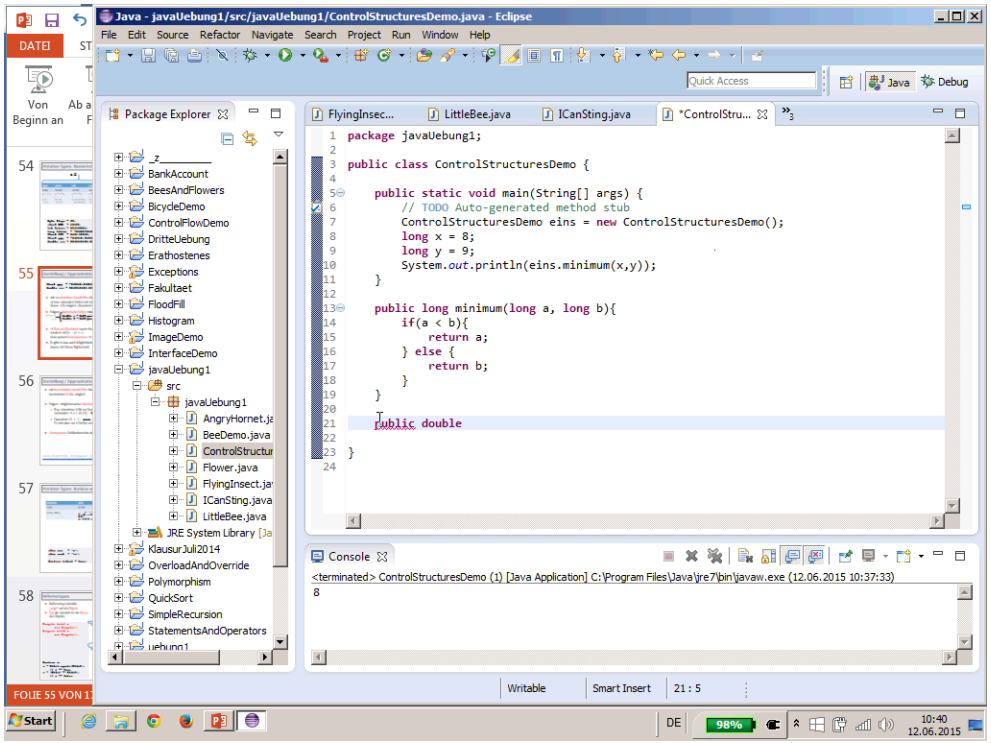

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8
9 }
10
11 public long minimum(long a, long b){
12
13 }
14
15 }
16
Console
<terminated> BeeDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:27:53)
Ei, ich hab so schoen 10.0 pollen eingesammelt *grins*
20.0
90.0
lalalalalalla = 90.0
Writable
Smart Insert
13 : 6

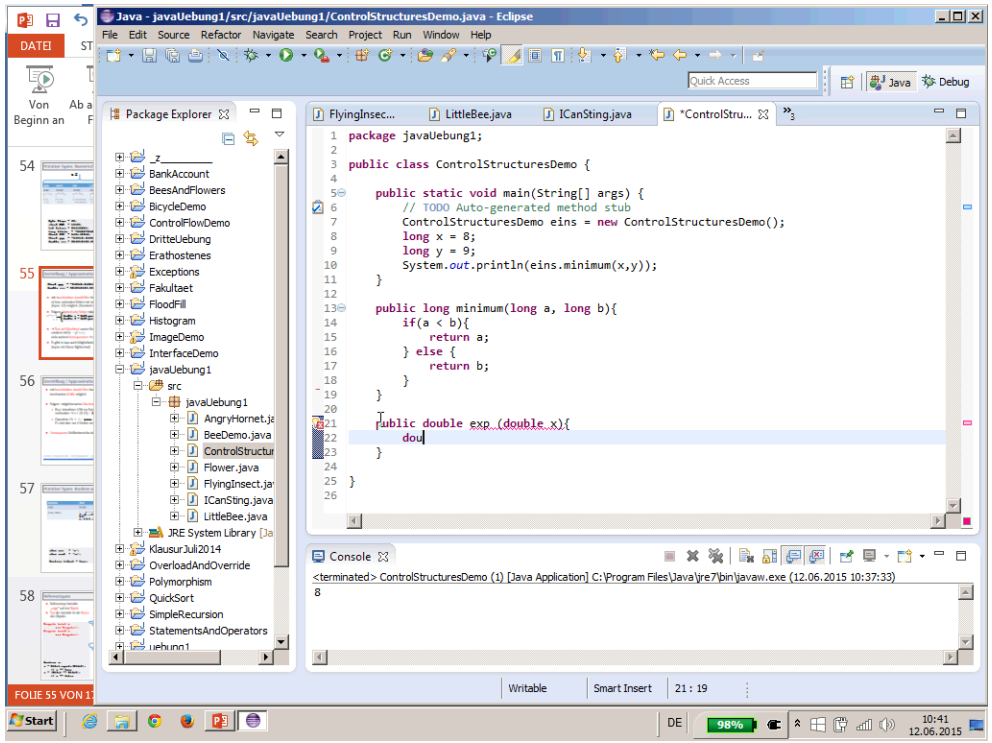

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public long minimum(long a, long b){
14 if(a < b){
15 return a;
16 } else {
17 return b;
18 }
19 }
20
21
22
23 }
24
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable
Smart Insert
10 : 47

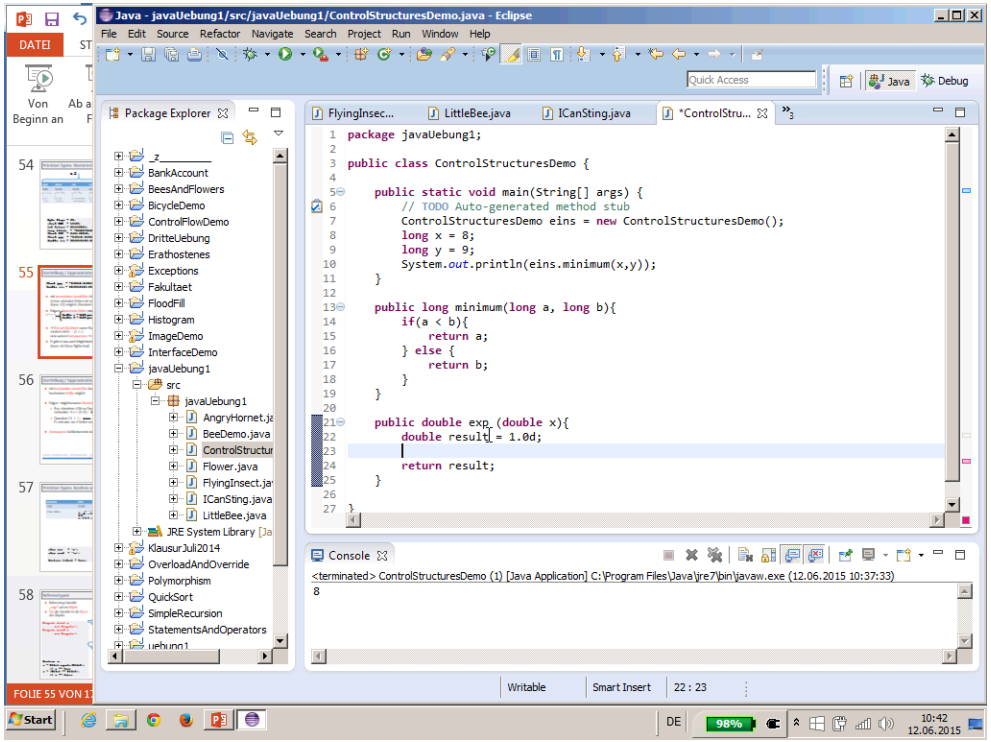

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public long minimum(long a, long b){
14 if(a < b){
15 return a;
16 } else {
17 return b;
18 }
19 }
20
21
22
23 }
24
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable
Smart Insert
10 : 45

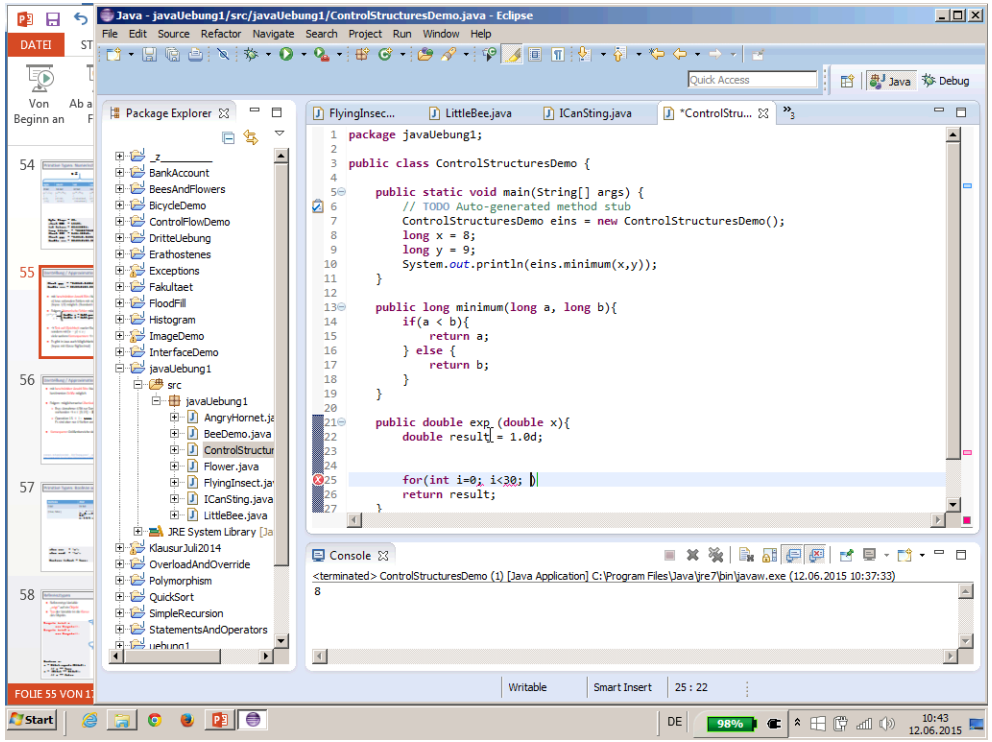

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public long minimum(long a, long b){
14 if(a < b){
15 return a;
16 } else {
17 return b;
18 }
19 }
20
21
22
23 }
24
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable Smart Insert 10 : 45

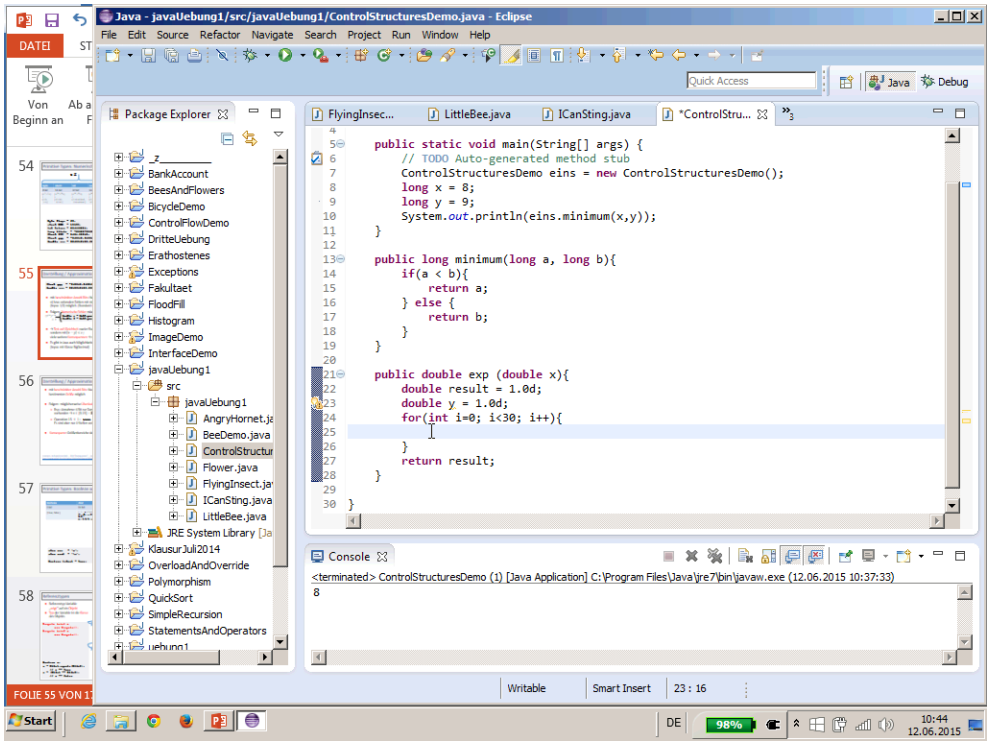

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public long minimum(long a, long b){
14 if(a < b){
15 return a;
16 } else {
17 return b;
18 }
19 }
20
21
22
23 }
24
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable Smart Insert 21 : 5

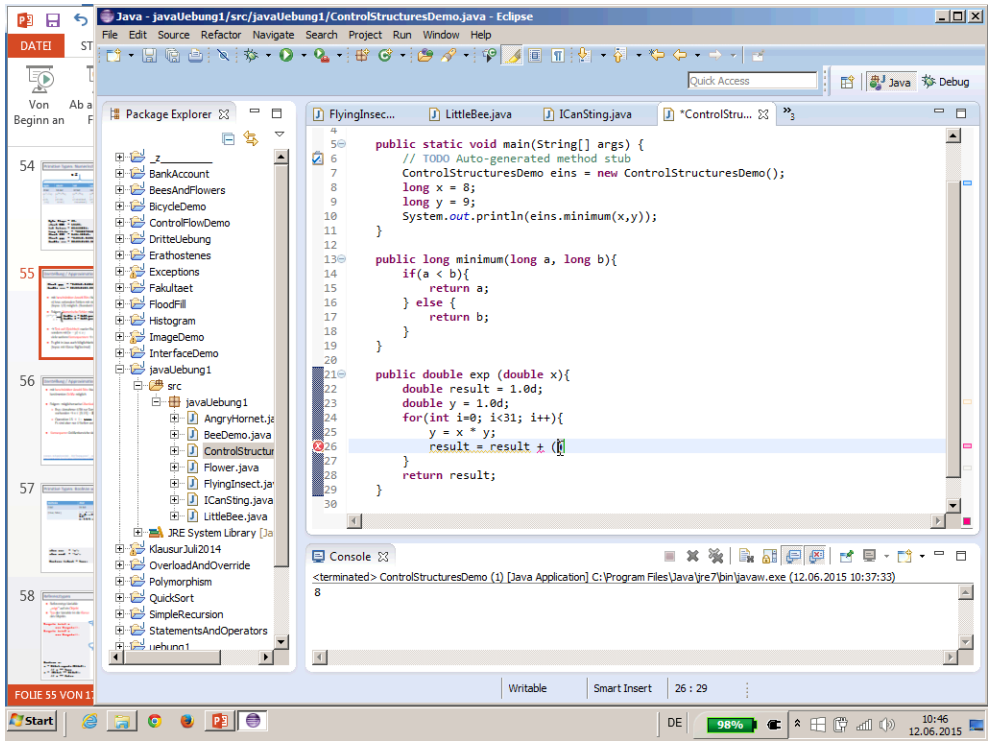

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public long minimum(long a, long b){
14 if(a < b){
15 return a;
16 } else {
17 return b;
18 }
19 }
20
21 public double
22
23 }
24
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable Smart Insert 21 : 5

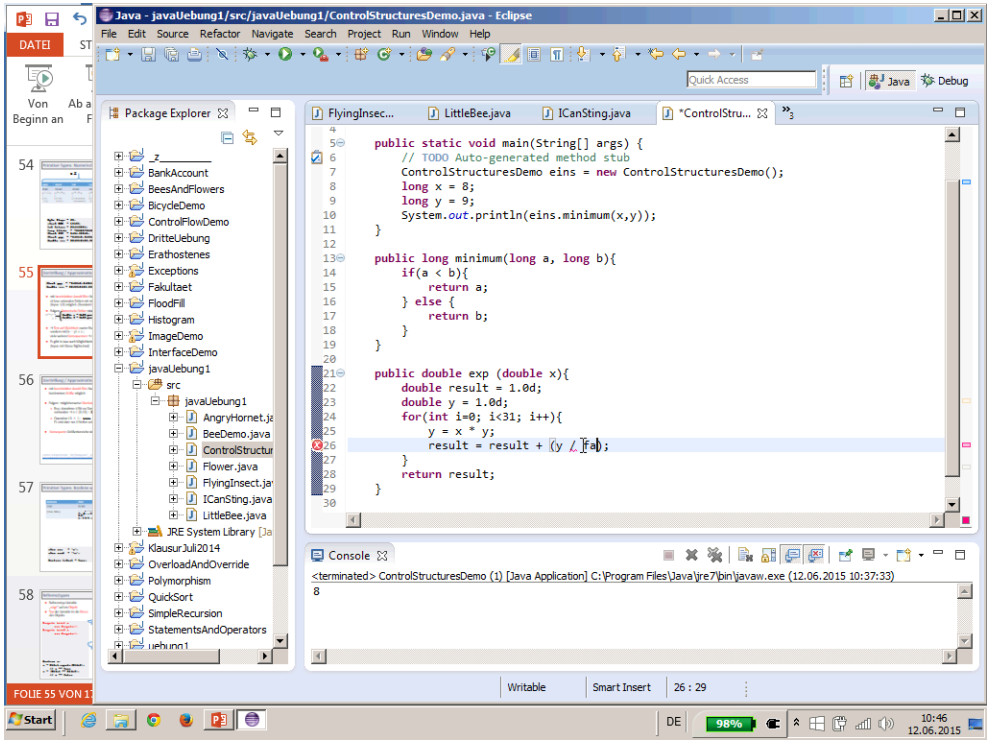

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public long minimum(long a, long b){
14 if(a < b){
15 return a;
16 } else {
17 return b;
18 }
19 }
20
21 public double exp (double x){
22 dou
23 }
24
25 }
26
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable Smart Insert 21 : 19

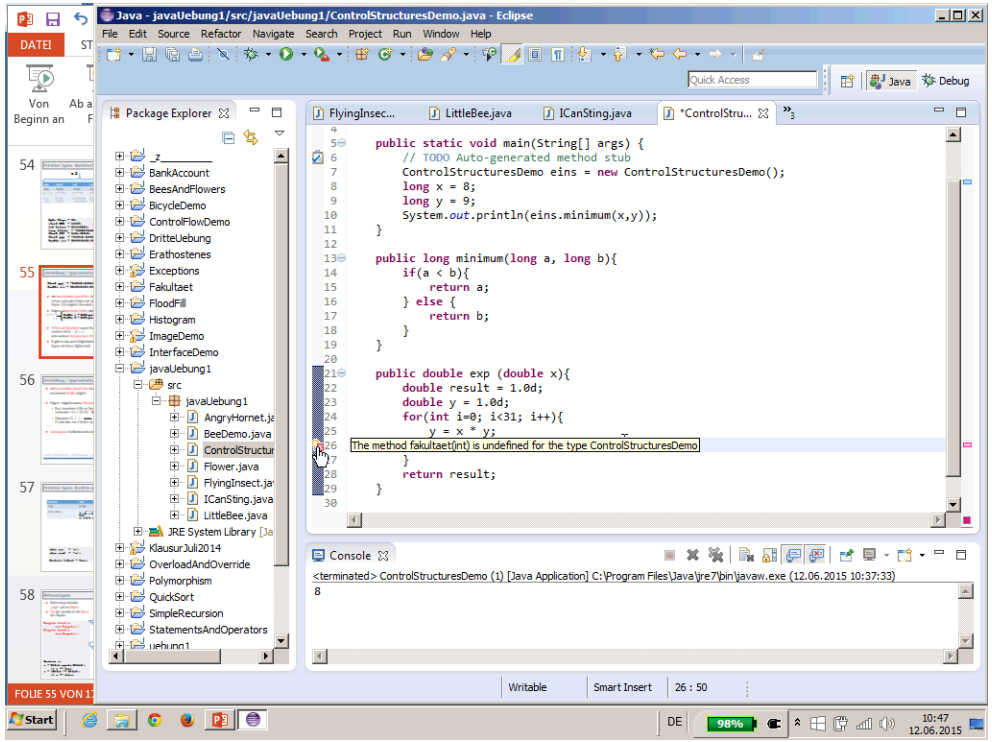

Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public long minimum(long a, long b){
14 if(a < b){
15 return a;
16 } else {
17 return b;
18 }
19 }
20
21 public double exp (double x){
22 double result = 1.0d;
23
24 return result;
25 }
26
27 }
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable Smart Insert 22 : 23
10:42
12.06.2015


Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public long minimum(long a, long b){
14 if(a < b){
15 return a;
16 } else {
17 return b;
18 }
19 }
20
21 public double exp (double x){
22 double result = 1.0d;
23
24
25 for(int i=0; i<30;)
26 return result;
27 }
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable Smart Insert 25 : 22
10:43
12.06.2015


Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public long minimum(long a, long b){
14 if(a < b){
15 return a;
16 } else {
17 return b;
18 }
19 }
20
21 public double exp (double x){
22 double result = 1.0d;
23 double y = 1.0d;
24 for(int i=0; i<30; i++){
25
26 }
27 return result;
28 }
29
30 }
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable Smart Insert 23 : 16
10:44
12.06.2015


Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo eins = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public long minimum(long a, long b){
14 if(a < b){
15 return a;
16 } else {
17 return b;
18 }
19 }
20
21 public double exp (double x){
22 double result = 1.0d;
23 double y = 1.0d;
24 for(int i=0; i<31; i++){
25 y = x * y;
26 result = result + (
27 }
28 return result;
29 }
30
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable Smart Insert 26 : 29
10:46
12.06.2015


Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
public static void main(String[] args) {
// TODO Auto-generated method stub
ControlStructuresDemo eins = new ControlStructuresDemo();
long x = 8;
long y = 9;
System.out.println(eins.minimum(x,y));
}
public long minimum(long a, long b){
if(a < b){
return a;
} else {
return b;
}
}
public double exp (double x){
double result = 1.0d;
double y = 1.0d;
for(int i=0; i<31; i++){
y = x * y;
result = result + (y / Fa);
}
return result;
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable Smart Insert 26 : 29
10:46
12.06.2015


Java - javaUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FlyingInsec... LittleBee.java ICanSting.java *ControlStru...
public static void main(String[] args) {
// TODO Auto-generated method stub
ControlStructuresDemo eins = new ControlStructuresDemo();
long x = 8;
long y = 9;
System.out.println(eins.minimum(x,y));
}
public long minimum(long a, long b){
if(a < b){
return a;
} else {
return b;
}
}
public double exp (double x){
double result = 1.0d;
double y = 1.0d;
for(int i=0; i<31; i++){
y = x * y;
The method fakultaet(int) is undefined for the type ControlStructuresDemo
}
return result;
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12.06.2015 10:37:33)
8
Writable Smart Insert 26 : 50
10:47
12.06.2015