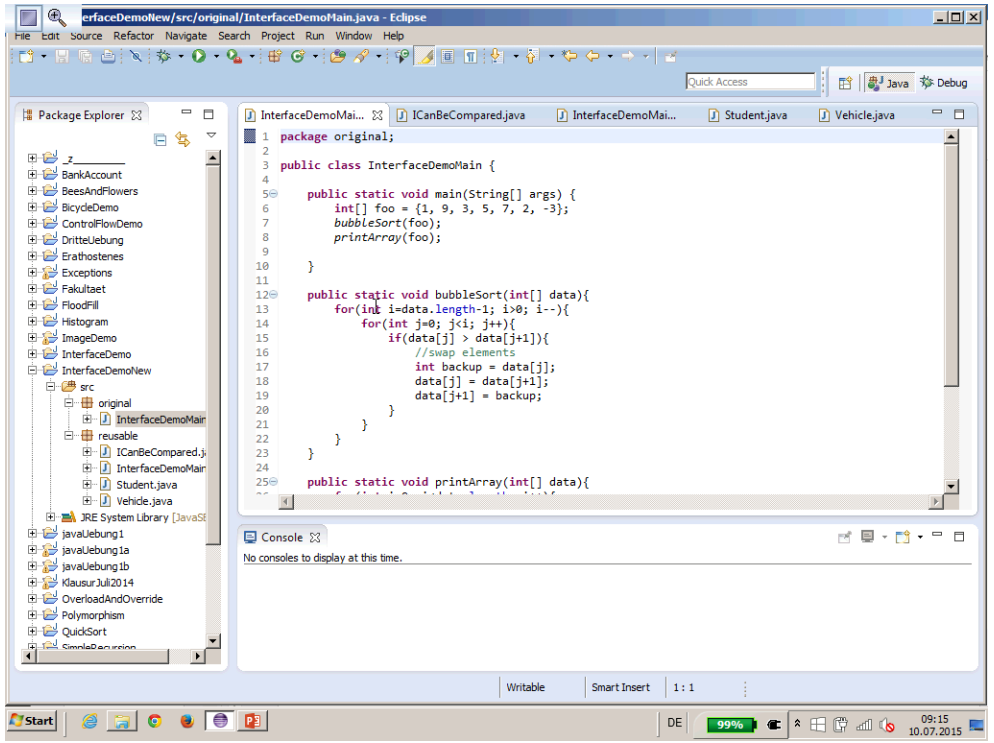
Script **generated by TTT**

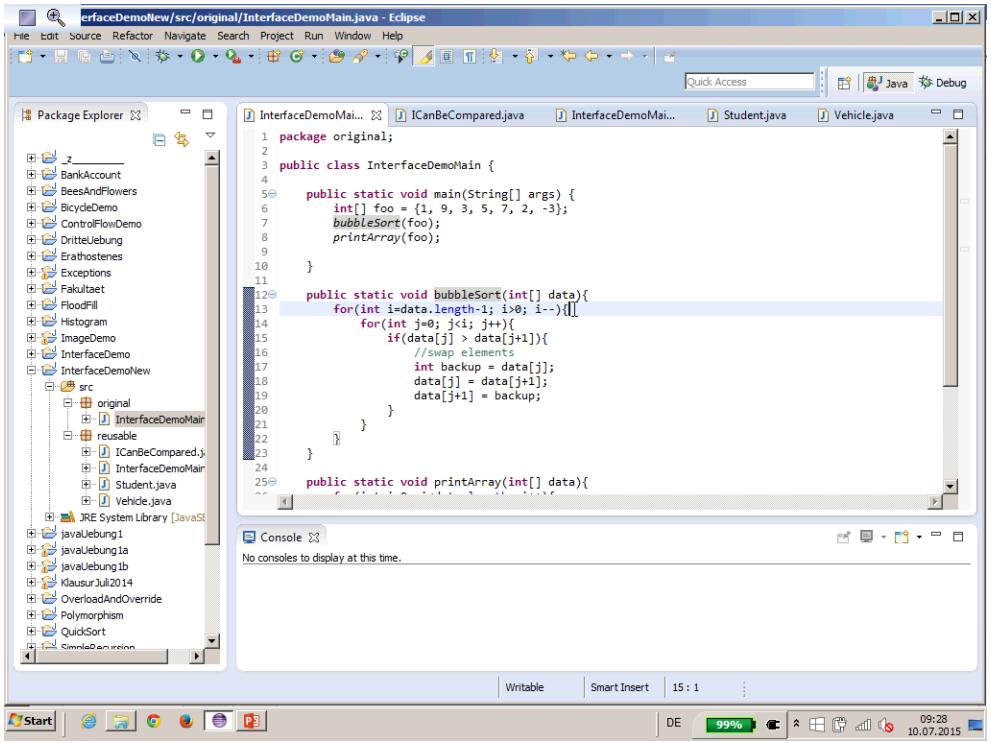
Title: groh: profile1 (10.07.2015)

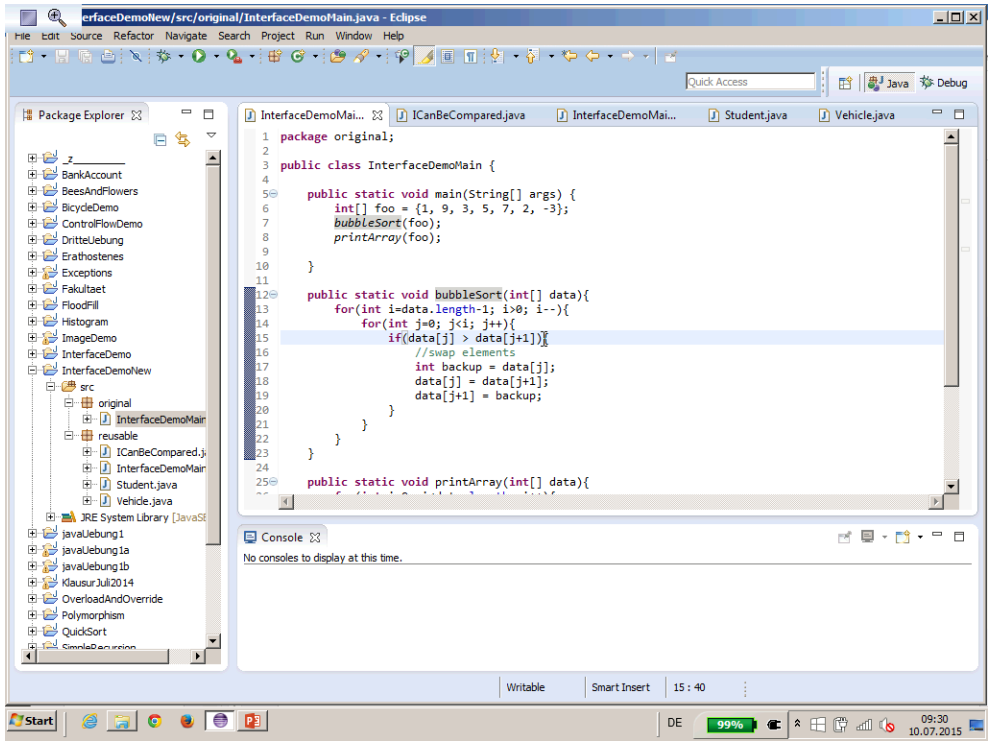
Date: Fri Jul 10 09:15:10 CEST 2015

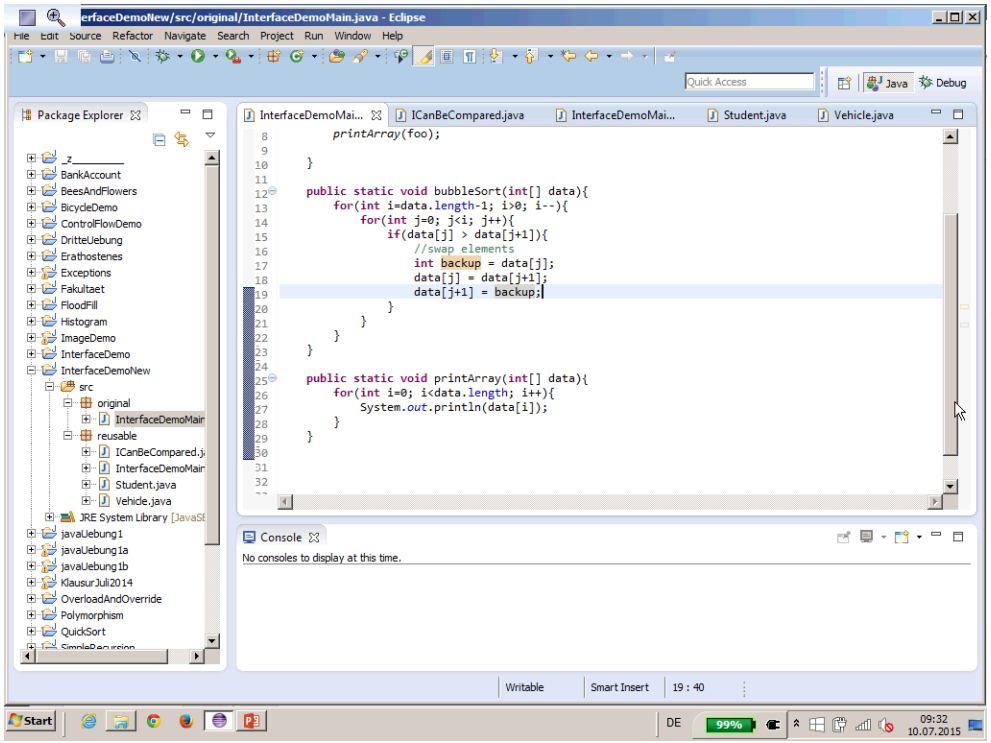
Duration: 90:17 min

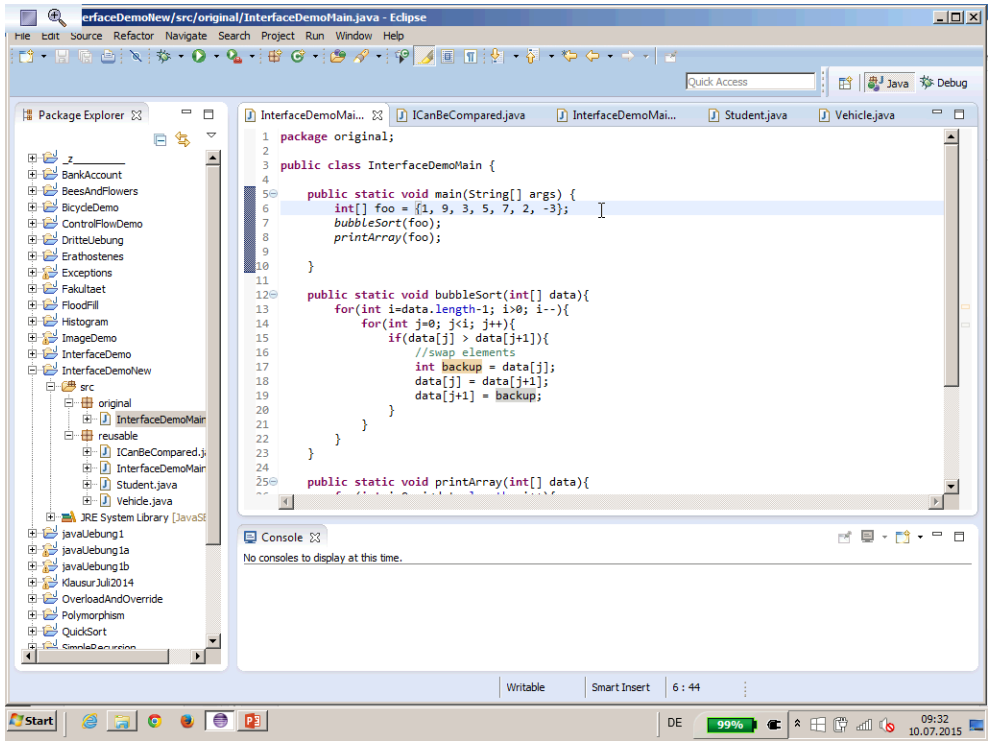
Pages: 62

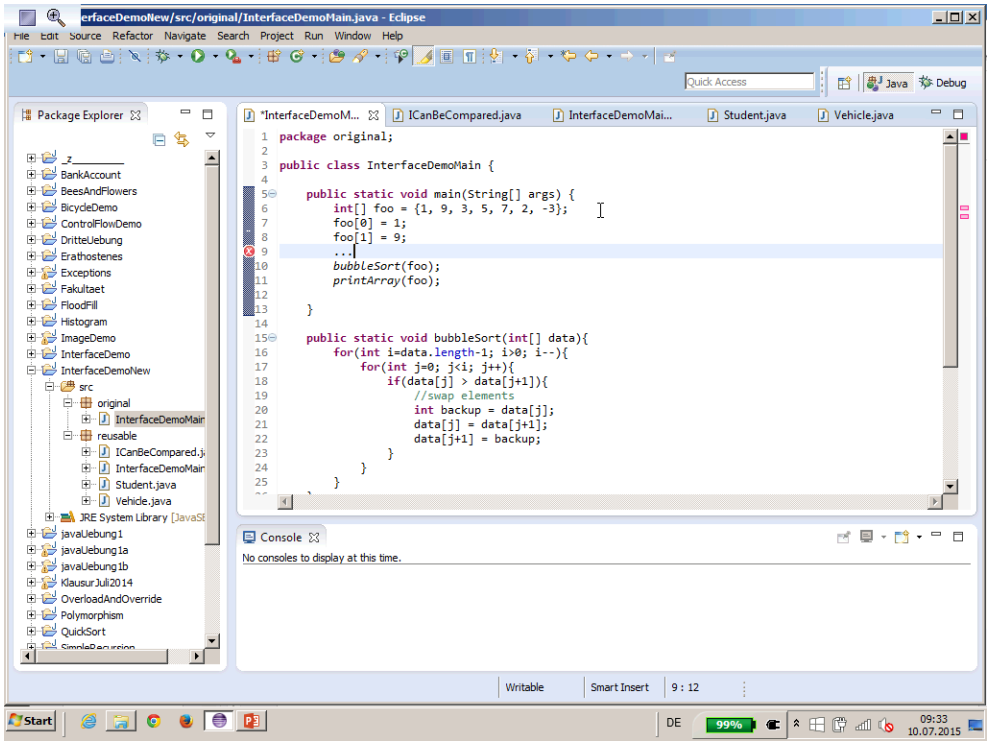


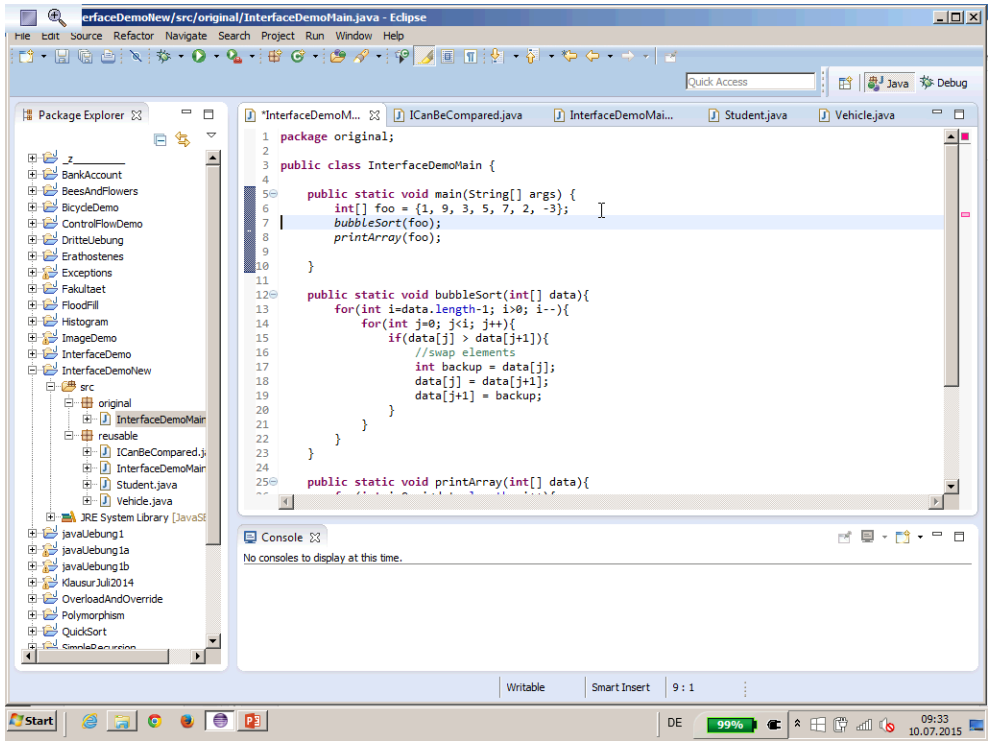


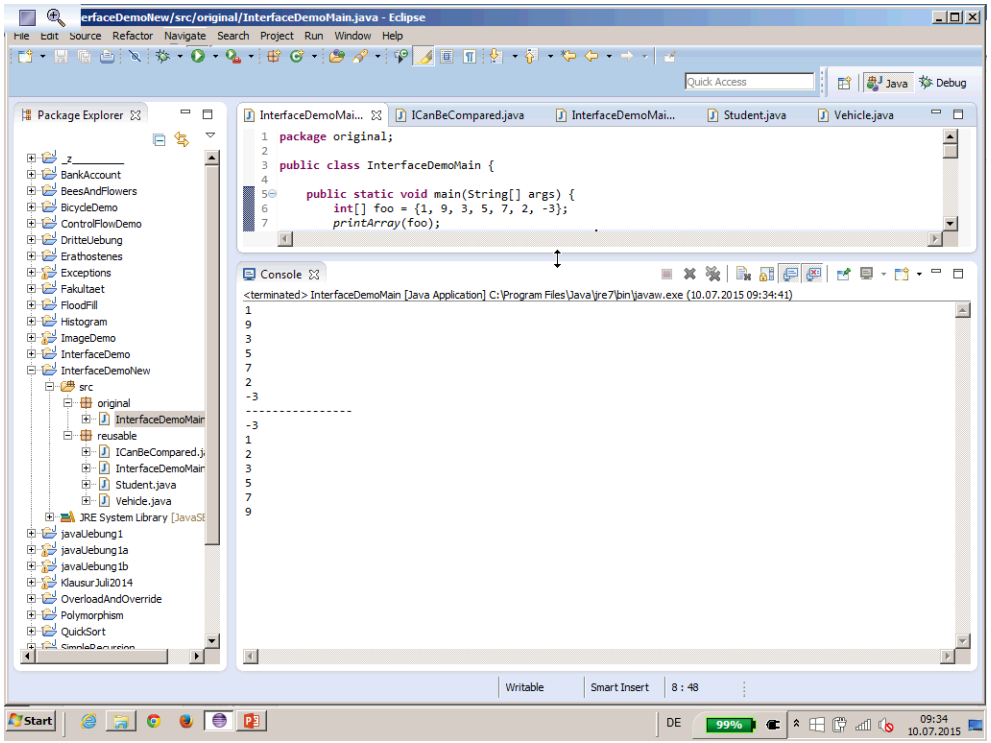



erfaceDemoNew/src/original/InterfaceDemoMain.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
8 printArray(foo);
9
10 }
11
12 public static void bubbleSort(int[] data){
13 for(int i=data.length-1; i>0; i--){
14 for(int j=0; j<i; j++){
15 if(data[j] > data[j+1]){
16 //swap elements
17 int backup = data[j];
18 data[j] = data[j+1];
19 data[j+1] = backup;
20 }
21 }
22 }
23
24
25 public static void printArray(int[] data){
26 for(int i=0; i<data.length; i++){
27 System.out.println(data[i]);
28 }
29 }
30
31
32
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
InterfaceDemoNew
src
original
InterfaceDemoMain
reusable
ICanBeCompared.j
InterfaceDemoMain
Student.java
Vehicle.java
JRE System Library [JavaSt
javaUebung1
javaUebung1a
javaUebung1b
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
Console
No consoles to display at this time.
Writable
Smart Insert
19 : 40
Start
DE
99%
09:32
10.07.2015

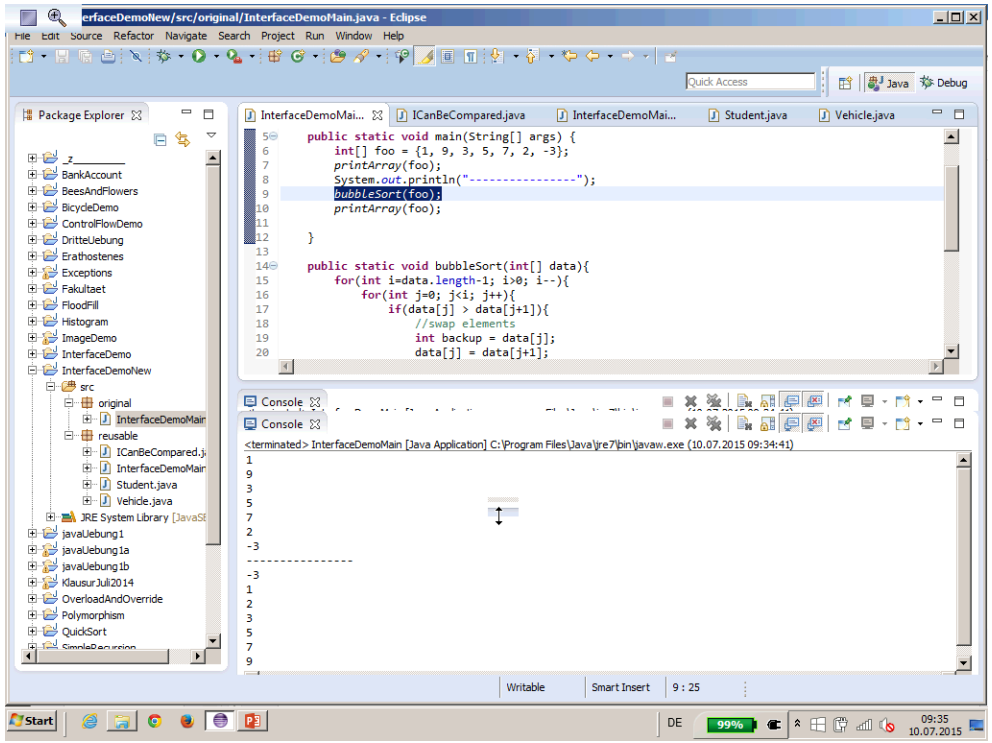

erfaceDemoNew/src/original/InterfaceDemoMain.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
1 package original;
2
3 public class InterfaceDemoMain {
4
5 public static void main(String[] args) {
6 int[] foo = {1, 9, 3, 5, 7, 2, -3};
7 bubbleSort(foo);
8 printArray(foo);
9
10 }
11
12 public static void bubbleSort(int[] data){
13 for(int i=data.length-1; i>0; i--){
14 for(int j=0; j<i; j++){
15 if(data[j] > data[j+1]){
16 //swap elements
17 int backup = data[j];
18 data[j] = data[j+1];
19 data[j+1] = backup;
20 }
21 }
22 }
23
24
25 public static void printArray(int[] data){
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
InterfaceDemoNew
src
original
InterfaceDemoMain
reusable
ICanBeCompared.j
InterfaceDemoMain
Student.java
Vehicle.java
JRE System Library [JavaSt
javaUebung1
javaUebung1a
javaUebung1b
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
Console
No consoles to display at this time.
Writable
Smart Insert
6 : 44
Start
DE
99%
09:32
10.07.2015


erfaceDemoNew/src/original/InterfaceDemoMain.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
*InterfaceDemoM...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
1 package original;
2
3 public class InterfaceDemoMain {
4
5 public static void main(String[] args) {
6 int[] foo = {1, 9, 3, 5, 7, 2, -3};
7 foo[0] = 1;
8 foo[1] = 9;
9 ...
10 bubbleSort(foo);
11 printArray(foo);
12
13 }
14
15 public static void bubbleSort(int[] data){
16 for(int i=data.length-1; i>0; i--){
17 for(int j=0; j<i; j++){
18 if(data[j] > data[j+1]){
19 //swap elements
20 int backup = data[j];
21 data[j] = data[j+1];
22 data[j+1] = backup;
23 }
24 }
25 }
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
InterfaceDemoNew
src
original
InterfaceDemoMain
reusable
ICanBeCompared.j
InterfaceDemoMain
Student.java
Vehicle.java
JRE System Library [JavaSt
javaUebung1
javaUebung1a
javaUebung1b
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
Console
No consoles to display at this time.
Writable
Smart Insert
9 : 12
Start
DE
99%
09:33
10.07.2015

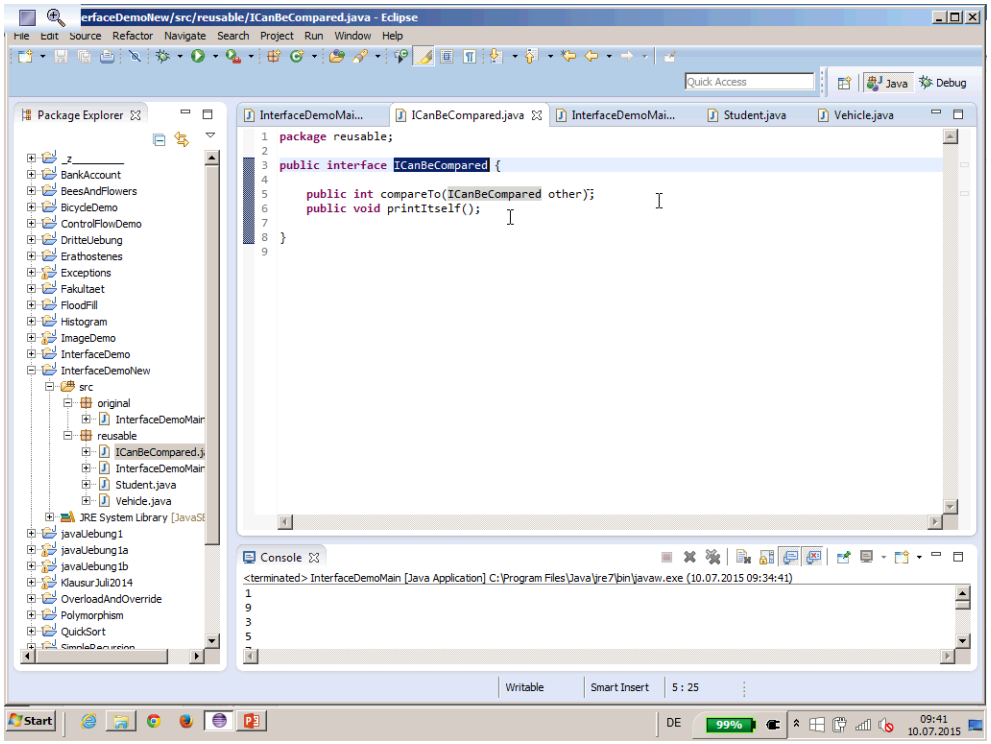

erfaceDemoNew/src/original/InterfaceDemoMain.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
*InterfaceDemoM...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
1 package original;
2
3 public class InterfaceDemoMain {
4
5 public static void main(String[] args) {
6 int[] foo = {1, 9, 3, 5, 7, 2, -3};
7 bubbleSort(foo);
8 printArray(foo);
9
10 }
11
12 public static void bubbleSort(int[] data){
13 for(int i=data.length-1; i>0; i--){
14 for(int j=0; j<i; j++){
15 if(data[j] > data[j+1]){
16 //swap elements
17 int backup = data[j];
18 data[j] = data[j+1];
19 data[j+1] = backup;
20 }
21 }
22 }
23
24
25 public static void printArray(int[] data){
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
InterfaceDemoNew
src
original
InterfaceDemoMain
reusable
ICanBeCompared.j
InterfaceDemoMain
Student.java
Vehicle.java
JRE System Library [JavaSt
javaUebung1
javaUebung1a
javaUebung1b
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
Console
No consoles to display at this time.
Writable
Smart Insert
9 : 1
Start
DE
99%
09:33
10.07.2015

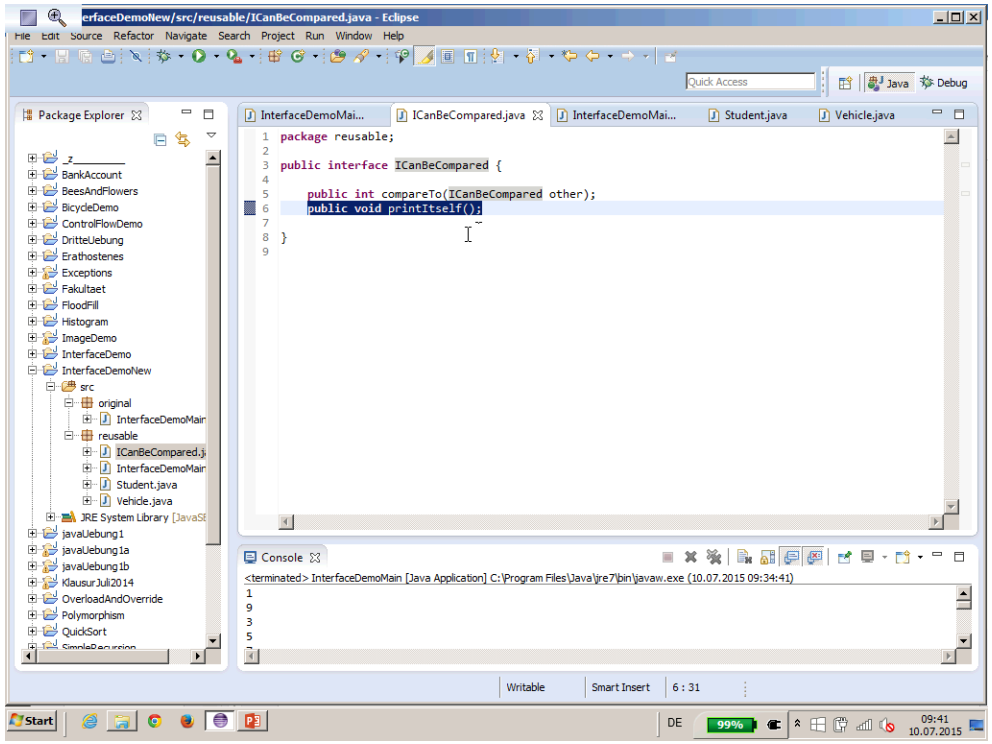

erfaceDemoNew/src/original/InterfaceDemoMain.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
1 package original;
2
3 public class InterfaceDemoMain {
4
5 public static void main(String[] args) {
6 int[] foo = {1, 9, 3, 5, 7, 2, -3};
7 printArray(foo);
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
1
9
3
5
7
2
-3

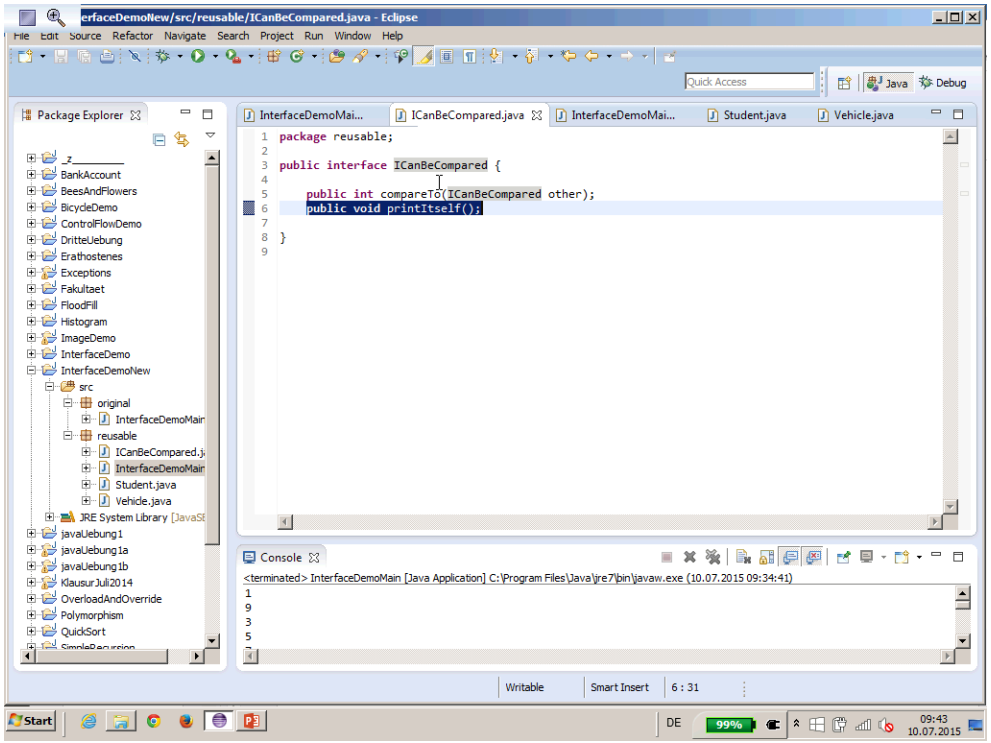
-3
1
2
3
5
7
9
Writable
Smart Insert
8 : 48
Start
DE
99%
09:34
10.07.2015

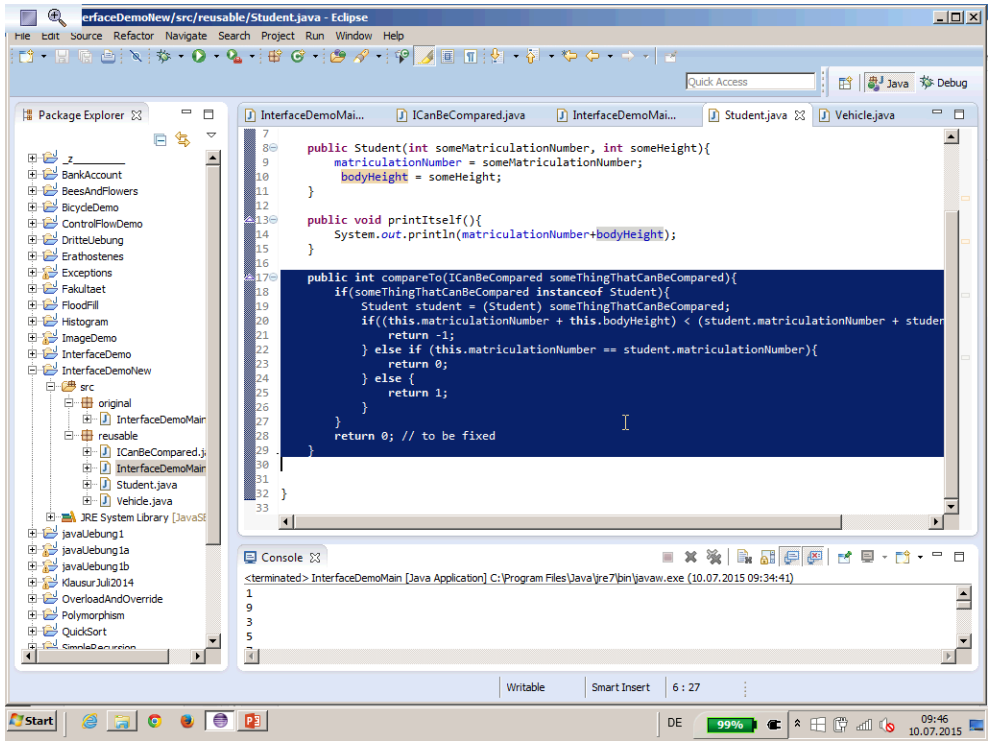

erfaceDemoNew/src/original/InterfaceDemoMain.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
5 public static void main(String[] args) {
6 int[] foo = {1, 9, 3, 5, 7, 2, -3};
7 printArray(foo);
8 System.out.println("----------");
9 bubbleSort(foo);
10 printArray(foo);
11
12 }
13
14 public static void bubbleSort(int[] data){
15 for(int i=data.length-1; i>0; i--){
16 for(int j=0; j<i; j++){
17 if(data[j] > data[j+1]){
18 //swap elements
19 int backup = data[j];
20 data[j] = data[j+1];
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
1
9
3
5
7
2
-3

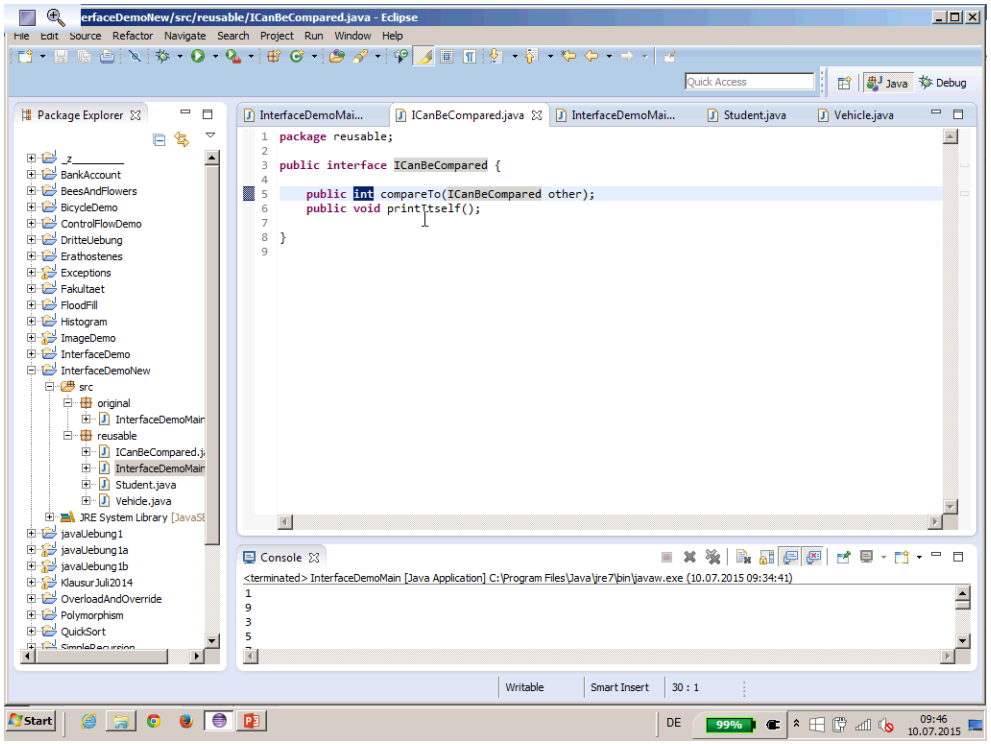
-3
1
2
3
5
7
9
Writable
Smart Insert
9 : 25
Start
DE
99%
09:35
10.07.2015

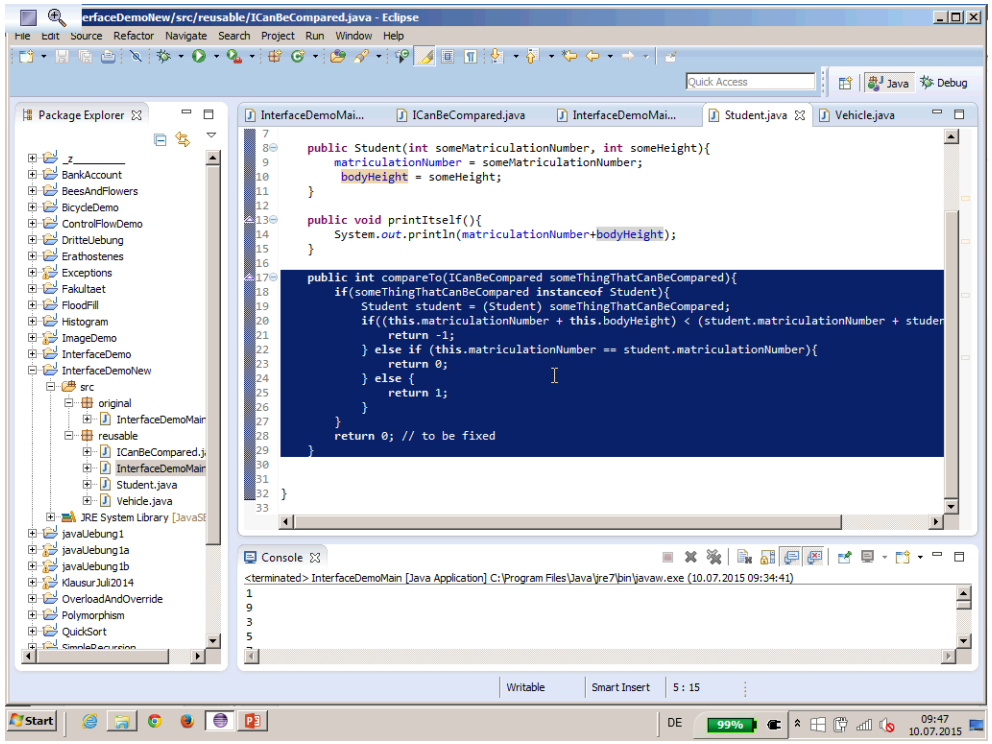

erfaceDemoNew/src/reusable/ICanBeCompared.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
1 package reusable;
2
3 public interface ICanBeCompared {
4
5 public int compareTo(ICanBeCompared other);
6 public void printItself();
7
8 }
9
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
1
9
3
5
Writable
Smart Insert
5 : 25
Start
DE
99%
09:41
10.07.2015

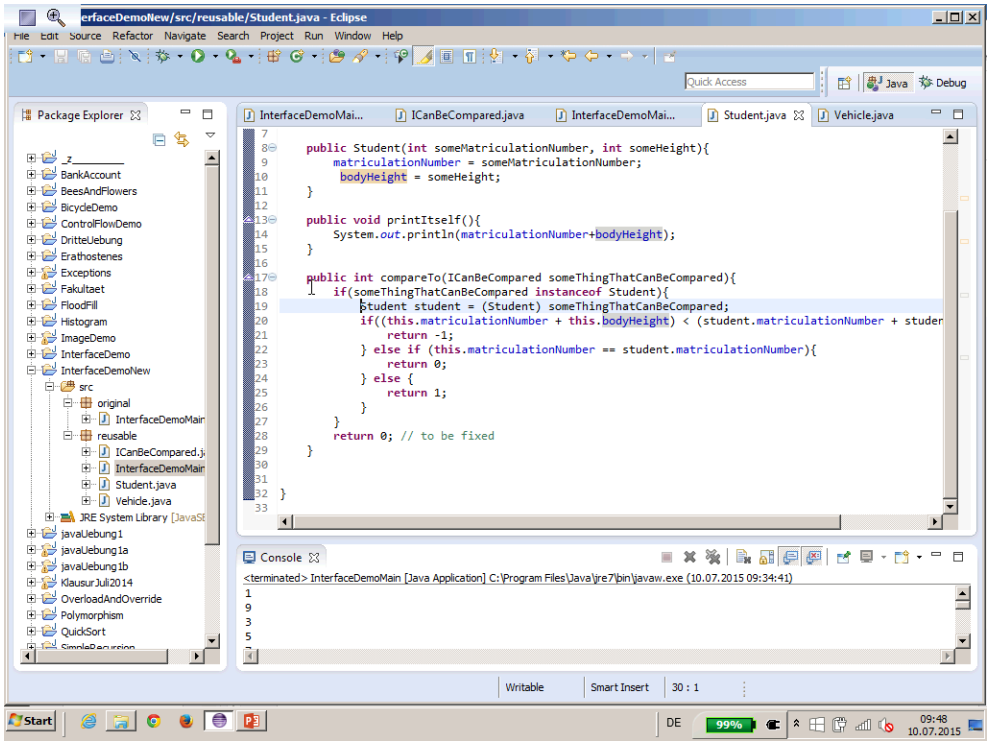

erfaceDemoNew/src/reusable/ICanBeCompared.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
1 package reusable;
2
3 public interface ICanBeCompared {
4
5 public int compareTo(ICanBeCompared other);
6 public void printItself();
7
8 }
9
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
1
9
3
5
Writable
Smart Insert
6 : 31
Start
DE
99%
09:41
10.07.2015

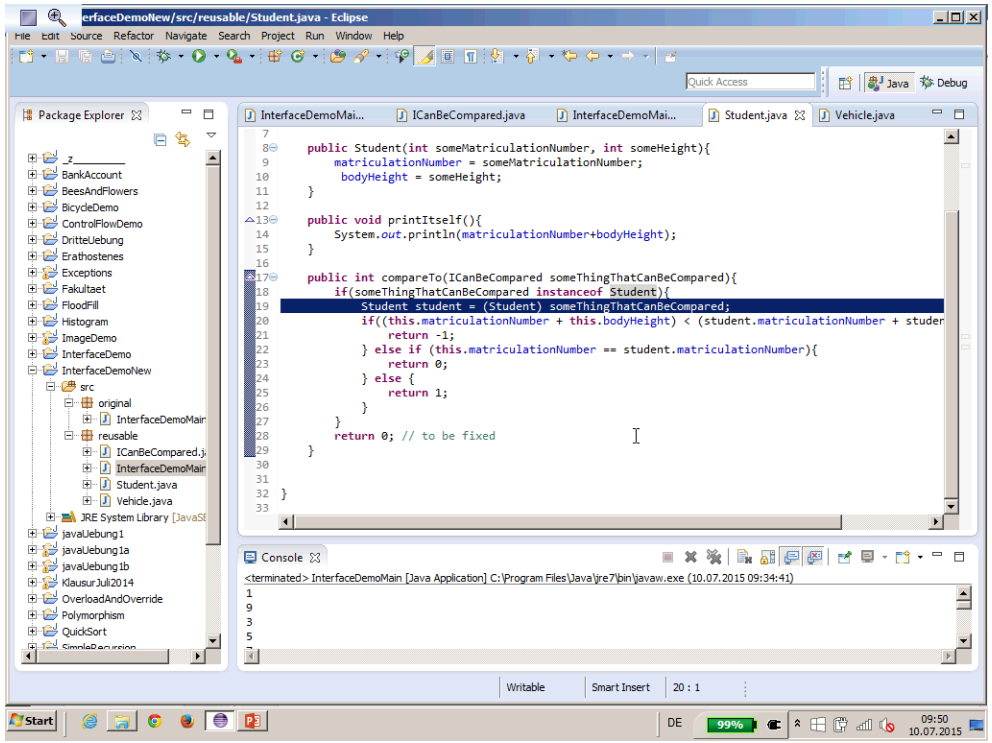

erfaceDemoNew/src/reusable/ICanBeCompared.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
1 package reusable;
2
3 public interface ICanBeCompared {
4
5 public int compareTo(ICanBeCompared other);
6 public void printItself();
7
8 }
9
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
Writable
Smart Insert
6 : 31
Start
DE
99%
09:43
10.07.2015

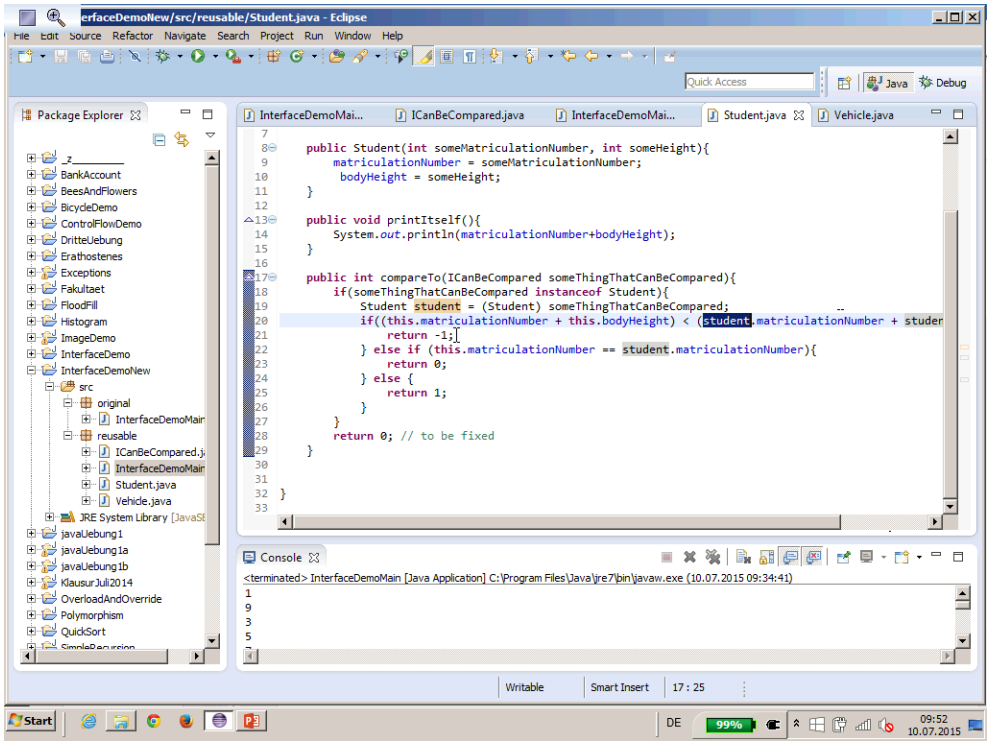

erfaceDemoNew/src/reusable/Student.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
8 public Student(int someMatriculationNumber, int someHeight){
9 matriculationNumber = someMatriculationNumber;
10 bodyHeight = someHeight;
11 }
12
13 public void printItself(){
14 System.out.println(matriculationNumber+bodyHeight);
15 }
16
17 public int compareTo(ICanBeCompared someThingThatCanBeCompared){
18 if(someThingThatCanBeCompared instanceof Student){
19 Student student = (Student) someThingThatCanBeCompared;
20 if((this.matriculationNumber + this.bodyHeight) < (student.matriculationNumber + studer
21 return -1;
22 } else if (this.matriculationNumber == student.matriculationNumber){
23 return 0;
24 } else {
25 return 1;
26 }
27 }
28 return 0; // to be fixed
29 }
30
31
32 }
33
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
Writable
Smart Insert
6 : 27
Start
DE
99%
09:46
10.07.2015

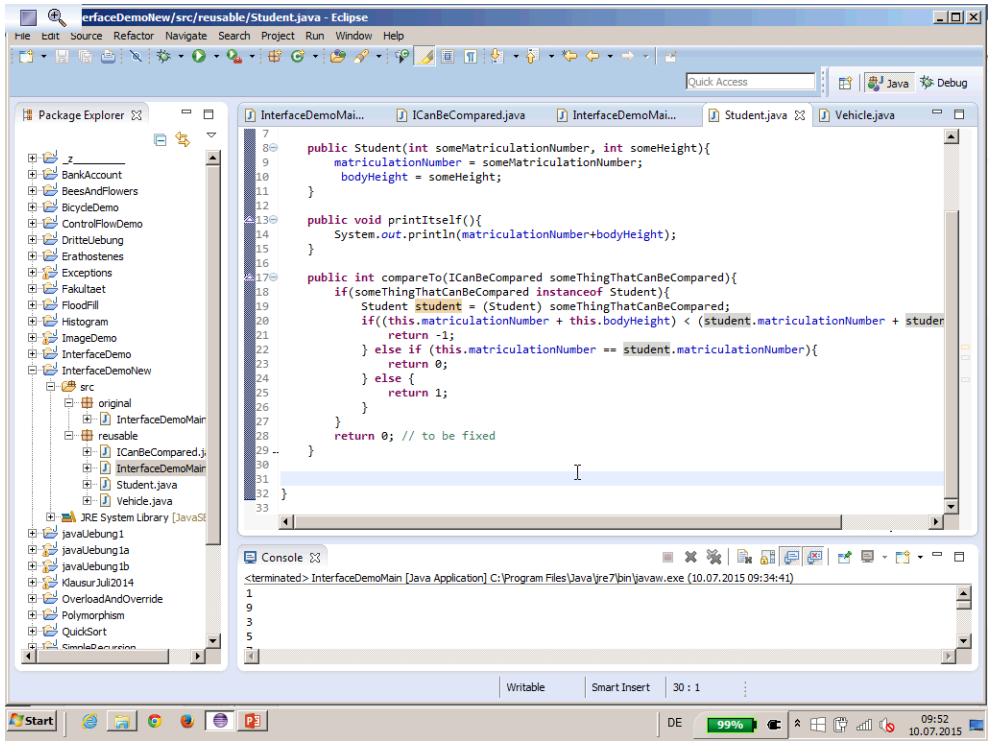

erfaceDemoNew/src/reusable/ICanBeCompared.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
1 package reusable;
2
3 public interface ICanBeCompared {
4
5 public int compareTo(ICanBeCompared other);
6 public void printItself();
7
8 }
9
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
Writable
Smart Insert
30 : 1
Start
DE
99%
09:46
10.07.2015

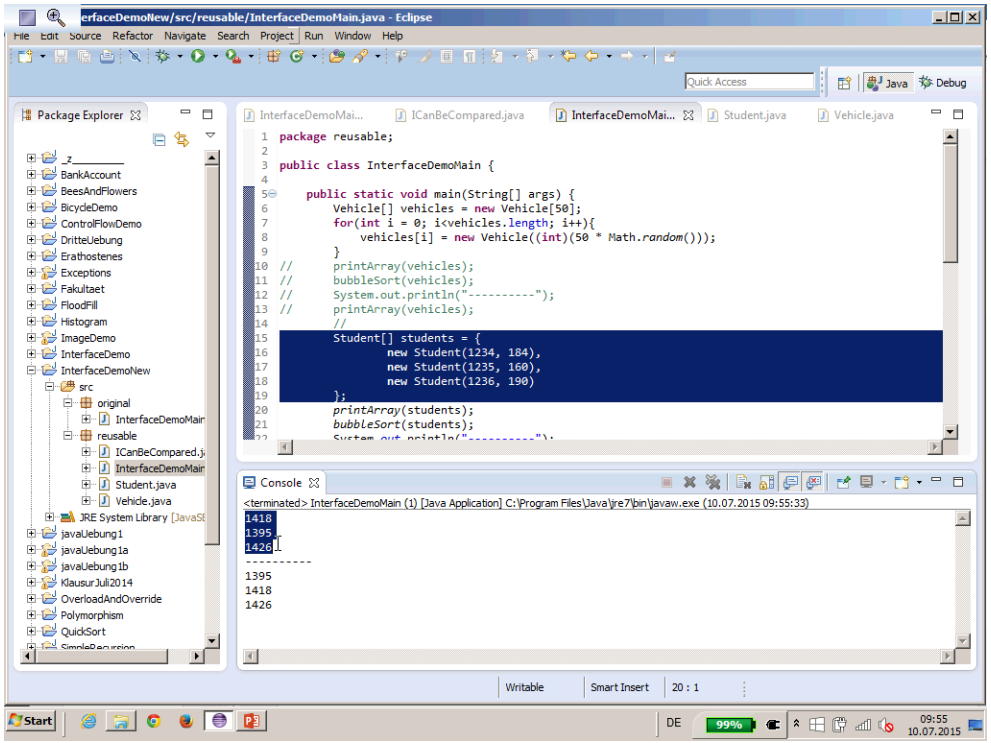

erfaceDemoNew/src/reusable/ICanBeCompared.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
8 public Student(int someMatriculationNumber, int someHeight){
9 matriculationNumber = someMatriculationNumber;
10 bodyHeight = someHeight;
11 }
12
13 public void printItself(){
14 System.out.println(matriculationNumber+bodyHeight);
15 }
16
17 public int compareTo(ICanBeCompared someThingThatCanBeCompared){
18 if(someThingThatCanBeCompared instanceof Student){
19 Student student = (Student) someThingThatCanBeCompared;
20 if((this.matriculationNumber + this.bodyHeight) < (student.matriculationNumber + studer
21 return -1;
22 } else if (this.matriculationNumber == student.matriculationNumber){
23 return 0;
24 } else {
25 return 1;
26 }
27 }
28 return 0; // to be fixed
29 }
30
31
32 }
33
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
Writable
Smart Insert
5 : 15
Start
DE
99%
09:47
10.07.2015

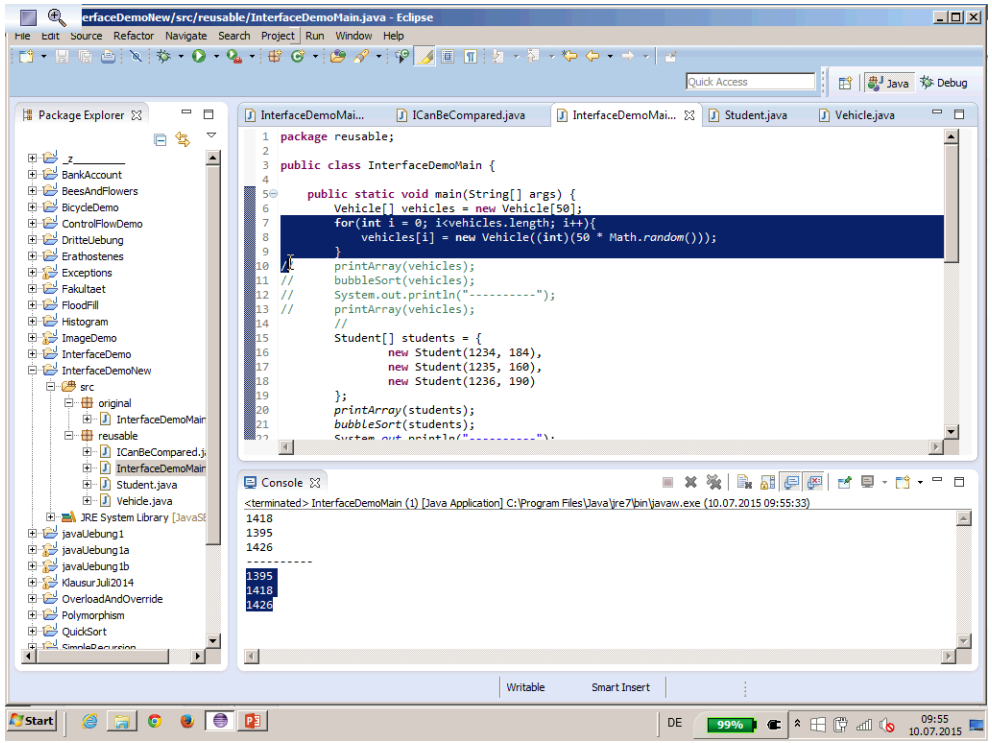

erfaceDemoNew/src/reusable/Student.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
public Student(int someMatriculationNumber, int someHeight){
matriculationNumber = someMatriculationNumber;
bodyHeight = someHeight;
}
public void printItself(){
System.out.println(matriculationNumber+bodyHeight);
}
public int compareTo(ICanBeCompared someThingThatCanBeCompared){
if(someThingThatCanBeCompared instanceof Student){
Student student = (Student) someThingThatCanBeCompared;
if((this.matriculationNumber + this.bodyHeight) < (student.matriculationNumber + studer
return -1;
} else if (this.matriculationNumber == student.matriculationNumber){
return 0;
} else {
return 1;
}
}
return 0; // to be fixed
}
}
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
1
9
3
5
Writable
Smart Insert
30 : 1
Start
DE
99%
09:48
10.07.2015

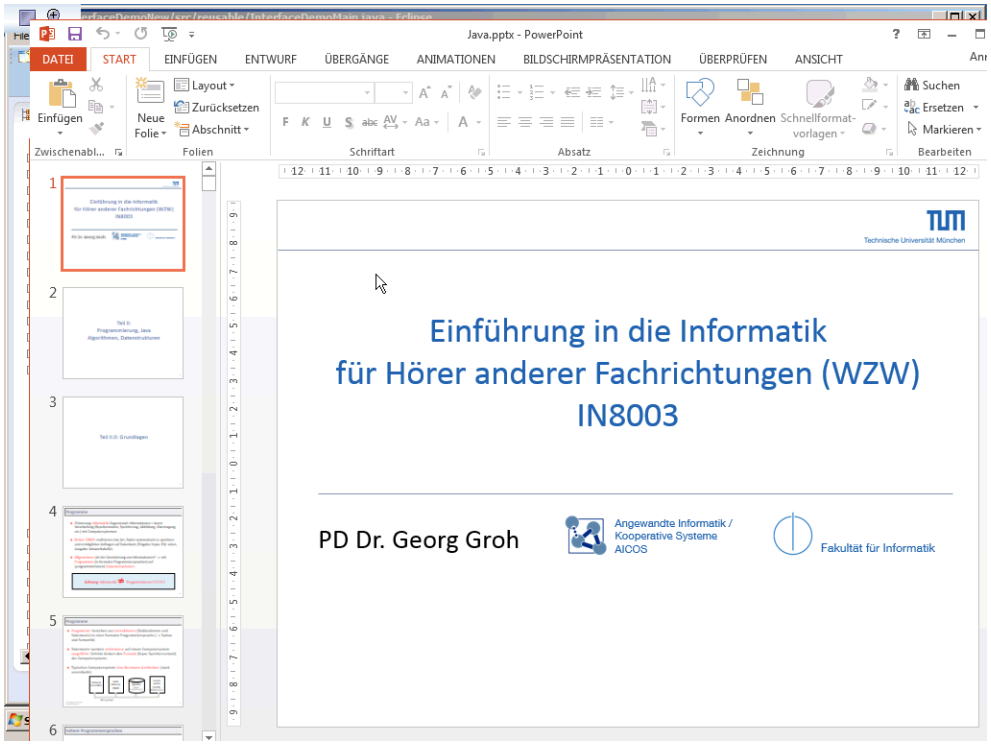

erfaceDemoNew/src/reusable/Student.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
public Student(int someMatriculationNumber, int someHeight){
matriculationNumber = someMatriculationNumber;
bodyHeight = someHeight;
}
public void printItself(){
System.out.println(matriculationNumber+bodyHeight);
}
public int compareTo(ICanBeCompared someThingThatCanBeCompared){
if(someThingThatCanBeCompared instanceof Student){
Student student = (Student) someThingThatCanBeCompared;
if((this.matriculationNumber + this.bodyHeight) < (student.matriculationNumber + studer
return -1;
} else if (this.matriculationNumber == student.matriculationNumber){
return 0;
} else {
return 1;
}
}
return 0; // to be fixed
}
}
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
1
9
3
5
Writable
Smart Insert
20 : 1
Start
DE
99%
09:50
10.07.2015


erfaceDemoNew/src/reusable/Student.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
public Student(int someMatriculationNumber, int someHeight){
matriculationNumber = someMatriculationNumber;
bodyHeight = someHeight;
}
public void printItself(){
System.out.println(matriculationNumber+bodyHeight);
}
public int compareTo(ICanBeCompared someThingThatCanBeCompared){
if(someThingThatCanBeCompared instanceof Student){
Student student = (Student) someThingThatCanBeCompared;
if((this.matriculationNumber + this.bodyHeight) < (student.matriculationNumber + studer
return -1;
} else if (this.matriculationNumber == student.matriculationNumber){
return 0;
} else {
return 1;
}
}
return 0; // to be fixed
}
}
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
1
9
3
5
Writable
Smart Insert
17 : 25
Start
DE
99%
09:52
10.07.2015


erfaceDemoNew/src/reusable/Student.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
InterfaceDemoMai...
ICanBeCompared.java
InterfaceDemoMai...
Student.java
Vehicle.java
public Student(int someMatriculationNumber, int someHeight){
matriculationNumber = someMatriculationNumber;
bodyHeight = someHeight;
}
public void printItself(){
System.out.println(matriculationNumber+bodyHeight);
}
public int compareTo(ICanBeCompared someThingThatCanBeCompared){
if(someThingThatCanBeCompared instanceof Student){
Student student = (Student) someThingThatCanBeCompared;
if((this.matriculationNumber + this.bodyHeight) < (student.matriculationNumber + studer
return -1;
} else if (this.matriculationNumber == student.matriculationNumber){
return 0;
} else {
return 1;
}
}
return 0; // to be fixed
}
}
Console
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (10.07.2015 09:34:41)
1
9
3
5
Writable
Smart Insert
30 : 1
Start
DE
99%
09:52
10.07.2015







Rekursion

- **Rekursion**:
  - Definiere eine Funktion durch sich selbst bzw.
  - formuliere Lösung eines Problems durch Rückgriff auf die Lösung selbst: Gebe Lösung für Basisfälle an und gib Regeln an, wie sich allg. Problem in Richtung der Basisfälle aufteilen lässt.
  - d.h. bspw. durch eine Methode, die sich selbst wieder aufruft

Bsp:

- Fakultät

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

- Fibonacci

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

Rekursion

- **Rekursion**:
  - Definiere eine Funktion durch sich selbst bzw.
  - formuliere Lösung eines Problems durch Rückgriff auf die Lösung selbst: Gebe Lösung für Basisfälle an und gib Regeln an, wie sich allg. Problem in Richtung der Basisfälle aufteilen lässt.
  - d.h. bspw. durch eine Methode, die sich selbst wieder aufruft

Bsp:

- Fakultät $$n! = \begin{cases} 1 & \text{if } n = 0, \\ (n-1)! \cdot n & \text{if } n > 0. \end{cases}$$
- Fibonacci $$f(n) = \begin{cases} 1 & \text{if } n = 1, \\ 1 & \text{if } n = 2, \\ f(n-1) + f(n-2) & \text{if } n > 2. \end{cases}$$

Rekursion

- **Rekursion**:
  - Definiere eine Funktion durch sich selbst bzw.
  - formuliere Lösung eines Problems durch Rückgriff auf die Lösung selbst: Gebe Lösung für Basisfälle an und gib Regeln an, wie sich allg. Problem in Richtung der Basisfälle aufteilen lässt.
  - d.h. bspw. durch eine Methode, die sich selbst wieder aufruft

Fakultät

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

```
long factorial(int n) {
    long temp;
    if (n == 0) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}
```


Rekursion

- **Rekursion**:
  - Definiere eine Funktion durch sich selbst bzw.
  - formuliere Lösung eines Problems durch Rückgriff auf die Lösung selbst: Gebe Lösung für Basisfälle an und gib Regeln an, wie sich allg. Problem in Richtung der Basisfälle aufteilen lässt.
  - d.h. bspw. durch eine Methode, die sich selbst wieder aufruft

Fakultät

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

```
long factorial(int n) {
    long temp;
    if (n == 0) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}
```


Call Stack

Für jeden Methodenaufruf (auch Klassenmethoden oder nicht rekursive Methodenaufrufe) werden lokale Variablen + Parameter + Rücksprungadresse („von wem (von wo aus) wurde die Methode aufgerufen?“) auf Call-Stack gespeichert

```
public class someClass {

  int a;
  int b;

  int someMethodOne(int paramOne1, int paramOne2){
    int localOne = 5;
    int result = someMethodTwo(17) + localOne;
    return result;
  }

  int someMethodTwo(int paramTwo){
    int localTwo = 8;
    return paramTwo * localTwo;
  }
}
```

```
…
    SomeClass someObject = new SomeClass();
    int bbb = someObject.someMethodOne(12, 34);
…
```

…	…	…
2345	someObject	<2346>
2346	someObject.a	
2347	someObject.b	
…		

…	…	…
5467		int result = someMethodTwo(17) + localOne;
…		
5899		int bbb = someObject.someMethodOne(12, 34);
…		

…	…	…
6000	(Rücksprung-Adresse)	<5467>
6001	(Aufruf-Objekt)	<2346>
6002	paramTwo	17
6003	localTwo	8
6004	(Rücksprung-Adresse)	<5899>
6005	(Aufruf-Objekt)	<…>
6006	paramOne1	12
6007	paramOne2	34
6008	localOne	5
…	…	…

Call Stack

Für jeden Methodenaufruf (auch Klassenmethoden oder nicht rekursive Methodenaufrufe) werden lokale Variablen + Parameter + Rücksprungadresse („von wem (von wo aus) wurde die Methode aufgerufen?“) auf Call-Stack gespeichert

```
public class someClass {

  int a;
  int b;

  int someMethodOne(int paramOne1, int paramOne2){
    int localOne = 5;
    int result = someMethodTwo(17) + localOne;
    return result;
  }

  int someMethodTwo(int paramTwo){
    int localTwo = 8;
    return paramTwo * localTwo;
  }
}
```

```
…
    SomeClass someObject = new SomeClass();
    int bbb = someObject.someMethodOne(12, 34);
…
```

…	…	…
2345	someObject	<2346>
2346	someObject.a	
2347	someObject.b	
…		

…	…	…
5467		int result = someMethodTwo(17) + localOne;
…		
5899		int bbb = someObject.someMethodOne(12, 34);
…		

…	…	…
6000	(Rücksprung-Adresse)	<5467>
6001	(Aufruf-Objekt)	<2346>
6002	paramTwo	17
6003	localTwo	8
6004	(Rücksprung-Adresse)	<5899>
6005	(Aufruf-Objekt)	<…>
6006	paramOne1	12
6007	paramOne2	34
6008	localOne	5
…	…	…

Call Stack

Für jeden Methodenaufruf (auch Klassenmethoden oder nicht rekursive Methodenaufrufe) werden lokale Variablen + Parameter + Rücksprungadresse („von wem (von wo aus) wurde die Methode aufgerufen?“) auf Call-Stack gespeichert

Speicherbereich für Objekte: „Heap“

Einfügeende des Call Stacks →

„Call Stack“

Rekursive Methodenaufrufe – Call Stack

```
long factorial(int n) {
    long temp;
    if (n == 0) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}
```

```
…
 int ccc = factorial(3);
…
```

	n	3
	temp	0
	ccc	…
	…	…
…	…	…

Rekursive Methodenaufrufe – Call Stack

```
long factorial(int n) {
    long temp;
    if (n == 0) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}
```

```
…
 int ccc = factorial(3);
…
```

	n	2
	temp	0
	n	3
	temp	0
	ccc	…
	…	…
…	…	…

Rekursive Methodenaufrufe – Call Stack

```
long factorial(int n) {
    long temp;
    if (n == 0) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}
```

```
…
 int ccc = factorial(3);
…
```

	n	1
	temp	0
	n	2
	temp	0
	n	3
	temp	0
	ccc	…
	…	…
…	…	…

Rekursive Methodenaufrufe – Call Stack

```
long factorial(int n) {
    long temp;
    if (n == 0) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}
```

```
…
 int ccc = factorial(3);
…
```

	n	0
	temp	0
	n	1
	temp	0
	n	2
	temp	0
	n	3
	temp	0
	ccc	…
	…	…
…	…	…

Rekursive Methodenaufrufe – Call Stack

```
long factorial(int n) {
    long temp;
    if (n == 0) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}
```

```
…
 int ccc = factorial(3);
…
```

	n	0
	temp	0
	n	1
	temp	0
	n	2
	temp	0
	n	3
	temp	0
	ccc	…
	…	…
…	…	…

Rekursive Methodenaufrufe – Call Stack

```
long factorial(int n) {
    long temp;
    if (n == 0) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}
```

```
…
 int ccc = factorial(3);
…
```

	n	3
	temp	2
	ccc	…
	…	…
…	…	…

Rekursive Methodenaufrufe – Call Stack

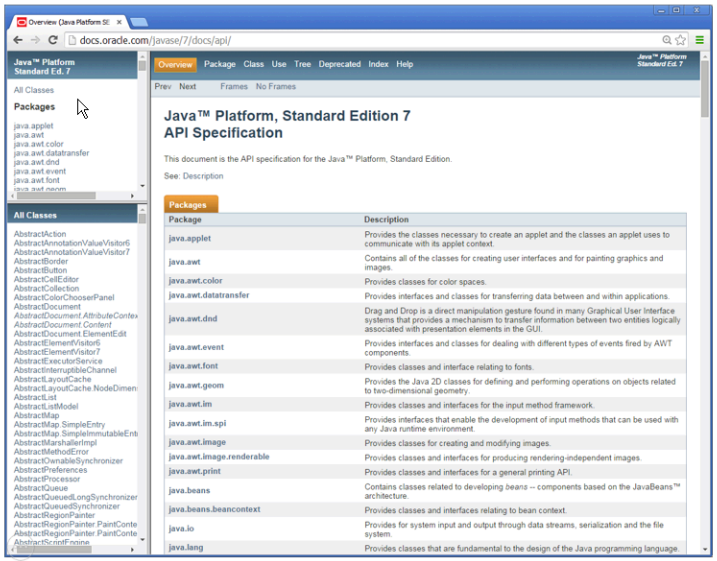
```
long factorial(int n) {
    long temp;
    if (n == 0) {
        return 1;
    } else {
        temp = factorial(n-1);
        return n * temp;
    }
}
```

```
…
 int ccc = factorial(3);
…
```

	n	3
	temp	2
	ccc	…
	…	…
…	…	…

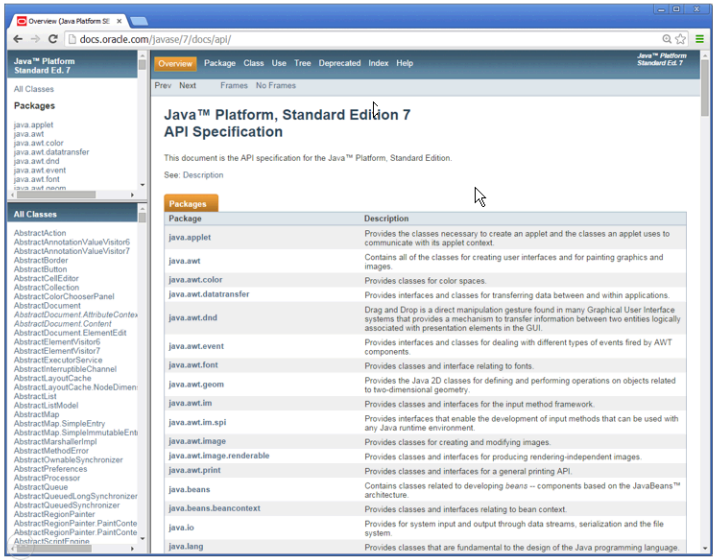
Java Klassenbibliothek

Java Klassenbibliothek: http://java.sun.com/javase/7/docs/api/
(Äquivalentes gibt es auch für andere Sprachen (bspw. .NET mit C#)



Java Klassenbibliothek

Java Klassenbibliothek: http://java.sun.com/javase/7/docs/api/
(Äquivalentes gibt es auch für andere Sprachen (bspw. .NET mit C#)



Packages

- Java Klassen sind in Hierarchie von packages organisiert

```
java.lang.String
java.net.URLConnection
java.util.Collection
javax.xml.parsers.SAXParser
org.w3c.dom.events.DocumentEvent
```

- Klassen die man nutzen möchte, muss man importieren.
  Alle Klassen aus java.lang werden automatisch importiert

```
import java.net.URLConnection;
import java.util.Collection;
import some.other.package.*;    // * means all classes

class SomeClass{
  …
}
```

Packages

- Java Klassen sind in Hierarchie von packages organisiert

```
java.lang.String
java.net.URLConnection
java.util.Collection
javax.xml.parsers.SAXParser
org.w3c.dom.events.DocumentEvent
```

- Klassen die man nutzen möchte, muss man importieren.
  Alle Klassen aus java.lang werden automatisch importiert

```
import java.net.URLConnection;
import java.util.Collection;
import some.other.package.*;     // * means all classes

class SomeClass{
  …
}
```


Packages

- Java Klassen sind in Hierarchie von packages organisiert

```
java.lang.String
java.net.URLConnection
java.util.Collection
javax.xml.parsers.SAXParser
org.w3c.dom.events.DocumentEvent
```

- Klassen die man nutzen möchte, muss man importieren.
  Alle Klassen aus java.lang werden automatisch importiert

```
import java.net.URLConnection;
import java.util.Collection;
import some.other.package.*;     // * means all classes

class SomeClass{
  …
}
```

173

Packages

- Java Klassen sind in Hierarchie von packages organisiert

```
java.lang.String
java.net.URLConnection
java.util.Collection
javax.xml.parsers.SAXParser
org.w3c.dom.events.DocumentEvent
```

- Klassen die man nutzen möchte, muss man importieren.
  Alle Klassen aus java.lang werden automatisch importiert

```
import java.net.URLConnection;
import java.util.Collection;
import some.other.package.*;     // * means all classes

class SomeClass{
  …
}
```

173

Packages

- Java Klassen sind in Hierarchie von packages organisiert

```
java.lang.String
java.net.URLConnection
java.util.Collection
javax.xml.parsers.SAXParser
org.w3c.dom.events.DocumentEvent
```

- Klassen die man nutzen möchte, muss man importieren.
  Alle Klassen aus java.lang werden automatisch importiert

```
import java.net.URLConnection;
import java.util.Collection;
import some.other.package.*;     // * means all classes

class SomeClass{
  …
}
```

173

Generics

- Manche Klassen kann man mit anderen Klassen parametrisieren
- Typisches Beispiel: Datenstrukturen
- Vorteil: Algorithmen und Datenstrukturen können generisch implementiert und genutzt werden.

ohne Generics:

```
Vector nonGenericVector = new Vector();
nonGenericVector.add( 1234 );
nonGenericVector.add( "Hello world" );
Object typeUnknown = nonGenericVector.get(0);
```

mit Generics:

```
Vector<Bicycle> bicycles = new Vector<Bicycle>();
bicycles.add( new Bicycle() );
bicycles.add( 123 ); // Compile time error!
Bicycle typeKnown = bicycles.get(0);
```


Generics

- Manche Klassen kann man mit anderen Klassen parametrisieren
- Typisches Beispiel: Datenstrukturen
- Vorteil: Algorithmen und Datenstrukturen können generisch implementiert und genutzt werden.

ohne Generics:

```
Vector nonGenericVector = new Vector();
nonGenericVector.add( 1234 );
nonGenericVector.add( "Hello world" );
Object typeUnknown = nonGenericVector.get(0);
```

mit Generics:

```
Vector<Bicycle> bicycles = new Vector<Bicycle>();
bicycles.add( new Bicycle() );
bicycles.add( 123 ); // Compile time error!
Bicycle typeKnown = bicycles.get(0);
```


Generics

- Manche Klassen kann man mit anderen Klassen parametrisieren
- Typisches Beispiel: Datenstrukturen
- Vorteil: Algorithmen und Datenstrukturen können generisch implementiert und genutzt werden.

ohne Generics:

```
Vector nonGenericVector = new Vector();
nonGenericVector.add( 1234 );
nonGenericVector.add( "Hello world" );
Object typeUnknown = nonGenericVector.get(0);
```

mit Generics:

```
Vector<Bicycle> bicycles = new Vector<Bicycle>();
bicycles.add( new Bicycle() );
bicycles.add( 123 ); // Compile time error!
Bicycle typeKnown = bicycles.get(0);
```


Wrapper Classes

- Für jeden primitiven Typ gibt es eine entsprechende Wrapper-Klasse

```
java.lang.Short
java.lang.Integer
java.lang.Long
java.lang.Float
java.lang.Double
java.lang.Boolean
 …
```

- Zur Bequemlichkeit im Umgang mit diesen Wrapper Klassen gibt es Boxing:

  statt:

  ```
  Integer someInteger = new Integer(723);
  ```

  kann man schreiben:

  ```
  Integer someInteger = 723;
  ```

Wrapper Classes

- Für jeden primitiven Typ gibt es eine entsprechende Wrapper-Klasse

```
java.lang.Short
java.lang.Integer
java.lang.Long
java.lang.Float
java.lang.Double
java.lang.Boolean
 …
```

- Zur Bequemlichkeit im Umgang mit diesen Wrapper Klassen gibt es Boxing:

  statt:

```
Integer someInteger = new Integer(723);
```

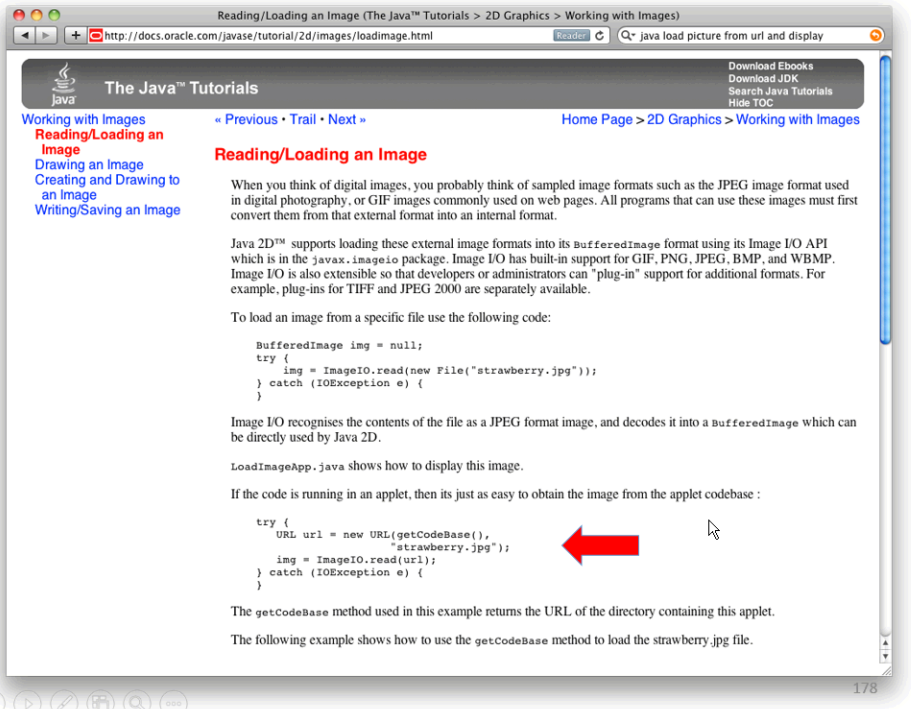
  kann man schreiben:

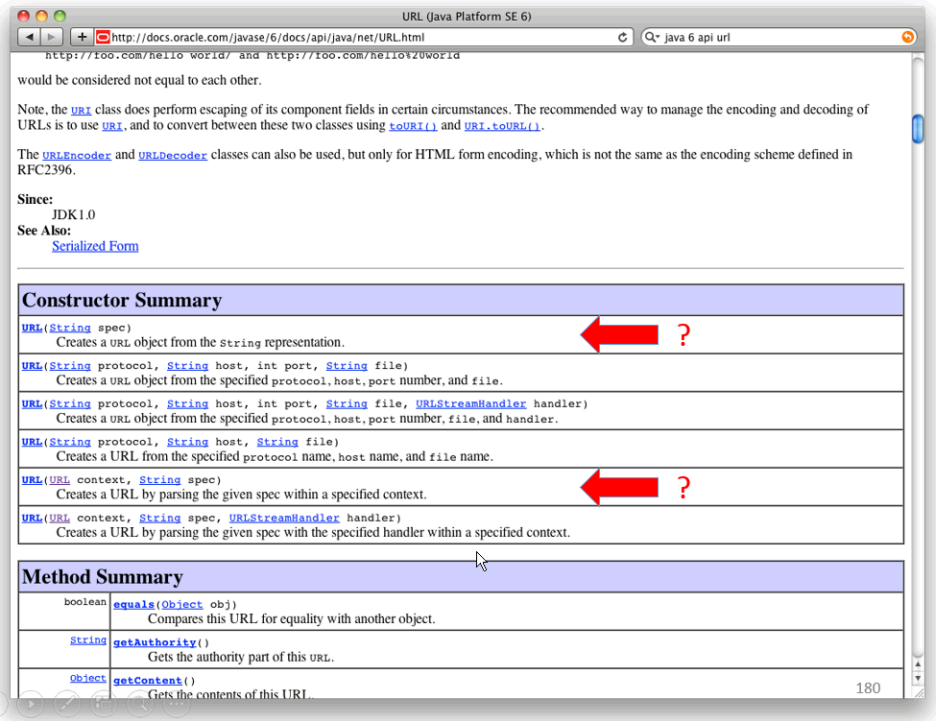
```
Integer someInteger = 723;
```


Lösen eines Beispiel-Problems

- Bsp.-Problem: *Ein Bild aus dem Web laden und in einem Fenster darstellen*

- Wie löse ich das Problem?
  1. Googlen
  2. Java Tutorial zu Rate ziehen
  3. Entsprechende Klassen, Methoden, Attribute in API-Doc nachschlagen
  4. Programm schreiben, testen
  5. Wenns nicht klappt : Gehe zu (1)





URL (Java Platform SE 6)

http://docs.oracle.com/javase/6/docs/api/java/net/URL.html — java 6 api url

http://foo.com/hello world/ and http://foo.com/hello%20world

would be considered not equal to each other.

Note, the URI class does perform escaping of its component fields in certain circumstances. The recommended way to manage the encoding and decoding of URLs is to use URI, and to convert between these two classes using toURI() and URI.toURL().

The URLEncoder and URLDecoder classes can also be used, but only for HTML form encoding, which is not the same as the encoding scheme defined in RFC2396.

Since:
JDK1.0
See Also:
Serialized Form

Constructor Summary

URL(String spec) — Creates a URL object from the String representation.
URL(String protocol, String host, int port, String file) — Creates a URL object from the specified protocol, host, port number, and file.
URL(String protocol, String host, int port, String file, URLStreamHandler handler) — Creates a URL object from the specified protocol, host, port number, file, and handler.
URL(String protocol, String host, String file) — Creates a URL from the specified protocol name, host name, and file name.
URL(URL context, String spec) — Creates a URL by parsing the given spec within a specified context.
URL(URL context, String spec, URLStreamHandler handler) — Creates a URL by parsing the given spec with the specified handler within a specified context.

Method Summary

boolean	equals(Object obj) — Compares this URL for equality with another object.
String	getAuthority() — Gets the authority part of this URL.
Object	getContent() — Gets the contents of this URL.

180

Reading/Loading an Image (The Java™ Tutorials > 2D Graphics > Working with Images)

http://docs.oracle.com/javase/tutorial/2d/images/loadimage.html — java load picture from url and display

The Java™ Tutorials — Download Ebooks · Download JDK · Search Java Tutorials · Hide TOC

Working with Images
- Reading/Loading an Image
- Drawing an Image
- Creating and Drawing to an Image
- Writing/Saving an Image

« Previous • Trail • Next » Home Page > 2D Graphics > Working with Images

Reading/Loading an Image

When you think of digital images, you probably think of sampled image formats such as the JPEG image format used in digital photography, or GIF images commonly used on web pages. All programs that can use these images must first convert them from that external format into an internal format.

Java 2D™ supports loading these external image formats into its `BufferedImage` format using its Image I/O API which is in the `javax.imageio` package. Image I/O has built-in support for GIF, PNG, JPEG, BMP, and WBMP. Image I/O is also extensible so that developers or administrators can "plug-in" support for additional formats. For example, plug-ins for TIFF and JPEG 2000 are separately available.

To load an image from a specific file use the following code:

```
BufferedImage img = null;
try {
    img = ImageIO.read(new File("strawberry.jpg"));
} catch (IOException e) {
}
```

Image I/O recognises the contents of the file as a JPEG format image, and decodes it into a `BufferedImage` which can be directly used by Java 2D.

`LoadImageApp.java` shows how to display this image.

If the code is running in an applet, then its just as easy to obtain the image from the applet codebase :

```
try {
   URL url = new URL(getCodeBase(),
                     "strawberry.jpg");
   img = ImageIO.read(url);
} catch (IOException e) {
}
```

The `getCodeBase` method used in this example returns the URL of the directory containing this applet.

The following example shows how to use the `getCodeBase` method to load the strawberry.jpg file.

182

Reading/Loading an Image (The Java™ Tutorials > 2D Graphics > Working with Images)

http://docs.oracle.com/javase/tutorial/2d/images/loadimage.html — java load picture from url and display

`LoadImageApp.java` shows how to display this image.

If the code is running in an applet, then its just as easy to obtain the image from the applet codebase :

```
try {
   URL url = new URL(getCodeBase(),
                     "strawberry.jpg");
   img = ImageIO.read(url);
} catch (IOException e) {
}
```

The `getCodeBase` method used in this example returns the URL of the directory containing this applet.

The following example shows how to use the `getCodeBase` method to load the strawberry.jpg file.

Note:

If you don't see the applet running, you need to install release 6 (or later) of the Java SE Development Kit (JDK).

`LoadImageApp.java` contains the complete code for this example and this applet requires the `strawberry.jpg` image file.

In addition to reading from files or URLS, Image I/O can read from other sources, such as an `InputStream`. `ImageIO.read()` is the most straightforward convenience API for most applications, but the `javax.imageio.ImageIO` class provides many more static methods for more advanced usages of the Image I/O API. The collection of methods on this class represent just a subset of the rich set of APIs for discovering information about the images and for controlling the image decoding (reading) process.

We will explore some of the other capabilities of Image I/O later in the Writing/Saving an Image section.

183

What's an easy way to display an Image in Java/Scala? - Stack Overflow

http://stackoverflow.com/questions/9027257/whats-an-easy-way-to-display-an-image-in- — easiest way load image java

50.4k • 4 • 20 • 54 — 256 • 1 • 5 — 91% accept rate

feedback

2 Answers — active | oldest | votes

```
JOptionPane.showMessageDialog(parent, new JLabel(new ImageIcon(theImage)));
```

9

Of course, if you have an `URL` for the image (or can form one from a `File` path), it can also be displayed as a tool-tip using HTML.

See Show full Image when hover over thumbnail for source.

link | improve this answer — edited Jan 28 at 2:18 — answered Jan 27 at 0:06 Andrew Thompson 50.4k • 4 • 20 • 54

Was this post useful to you? Yes No

scala × 9773

asked 5 months ago
viewed 222 times
active 5 months ago

Linked
Show full Image when hover over thumbnail as Popup/Overlay

Related

186

JOptionPane (Java Platform SE 6)

http://docs.oracle.com/javase/6/docs/api/javax/swing/JOptionPane.html

java 6 api joptionpane

Overview Package Class Use Tree Deprecated Index Help

Java™ Platform Standard Ed. 6

PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

javax.swing

Class JOptionPane

java.lang.Object
  └ java.awt.Component
    └ java.awt.Container
      └ javax.swing.JComponent
        └ **javax.swing.JOptionPane**

All Implemented Interfaces:
ImageObserver, MenuContainer, Serializable, Accessible

```
public class JOptionPane
extends JComponent
implements Accessible
```

`JOptionPane` makes it easy to pop up a standard dialog box that prompts users for a value or informs them of something. For information about using `JOptionPane`, see How to Make Dialogs, a section in *The Java Tutorial*.

While the `JOptionPane` class may appear complex because of the large number of methods, almost all uses of this class are one-line calls to one of the static `showXxxDialog` methods shown below:

Method Name	Description
showConfirmDialog	Asks a confirming question, like yes/no/cancel.
showInputDialog	Prompt for some input.
showMessageDialog	Tell the user about something that has happened.
showOptionDialog	The Grand Unification of the above three.

Each of these methods also comes in a `showInternalXxx` flavor, which uses an internal frame to hold the dialog box (see `JInternalFrame`). Multiple convenience methods have also been defined -- overloaded versions of the basic methods that use different parameter lists.

All dialogs are modal. Each `showXxxDialog` method blocks the caller until the user's interaction is complete.

The basic appearance of one of these dialog boxes is generally similar to the picture at the right, although the various look-and-feels are

187

JOptionPane (Java Platform SE 6)

http://docs.oracle.com/javase/6/docs/api/javax/swing/JOptionPane.html#showMessageDialog(ja

java 6 api joptionpane

showMessageDialog

```
public static void showMessageDialog(Component parentComponent,
                                      Object message)
                               throws HeadlessException
```

Brings up an information-message dialog titled "Message".

Parameters:
`parentComponent` - determines the `Frame` in which the dialog is displayed; if `null`, or if the `parentComponent` has no `Frame`, a default `Frame` is used
`message` - the `Object` to display
Throws:
`HeadlessException` - if `GraphicsEnvironment.isHeadless` returns `true`
See Also:
`GraphicsEnvironment.isHeadless()`

showMessageDialog

```
public static void showMessageDialog(Component parentComponent,
                                     Object message,
                                     String title,
                                     int messageType)
                               throws HeadlessException
```

Brings up a dialog that displays a message using a default icon determined by the `messageType` parameter.

Parameters:
`parentComponent` - determines the `Frame` in which the dialog is displayed; if `null`, or if the `parentComponent` has no `Frame`, a default `Frame` is used
`message` - the `Object` to display
`title` - the title string for the dialog
`messageType` - the type of message to be displayed: `ERROR_MESSAGE`, `INFORMATION_MESSAGE`, `WARNING_MESSAGE`, `QUESTION_MESSAGE`, or `PLAIN_MESSAGE`
Throws:
`HeadlessException` - if `GraphicsEnvironment.isHeadless` returns `true`
See Also:
`GraphicsEnvironment.isHeadless()`

showMessageDialog

188

JOptionPane (Java Platform SE 6)

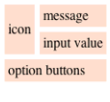
http://docs.oracle.com/javase/6/docs/api/javax/swing/JOptionPane.html#showMessageDialog(ja

Google

convenience methods have also been defined -- overloaded versions of the basic methods that use different parameter lists.

All dialogs are modal. Each `showXxxDialog` method blocks the caller until the user's interaction is complete.

The basic appearance of one of these dialog boxes is generally similar to the picture at the right, although the various look-and-feels are ultimately responsible for the final result. In particular, the look-and-feels will adjust the layout to accommodate the option pane's `ComponentOrientation` property.



Parameters:
The parameters to these methods follow consistent patterns:

parentComponent
Defines the `Component` that is to be the parent of this dialog box. It is used in two ways: the `Frame` that contains it is used as the `Frame` parent for the dialog box, and its screen coordinates are used in the placement of the dialog box. In general, the dialog box is placed just below the component. This parameter may be `null`, in which case a default `Frame` is used as the parent, and the dialog will be centered on the screen (depending on the L&F).
message
A descriptive message to be placed in the dialog box. In the most common usage, message is just a `String` or `String` constant. However, the type of this parameter is actually `Object`. Its interpretation depends on its type:

Object[]
An array of objects is interpreted as a series of messages (one per object) arranged in a vertical stack. The interpretation is recursive -- each object in the array is interpreted according to its type.
Component
The `Component` is displayed in the dialog.
Icon
The `Icon` is wrapped in a `JLabel` and displayed in the dialog.
others
The object is converted to a `String` by calling its `toString` method. The result is wrapped in a `JLabel` and displayed.

messageType
Defines the style of the message. The Look and Feel manager may lay out the dialog differently depending on this value, and will often provide a default icon. The possible values are:

- `ERROR_MESSAGE`
- `INFORMATION_MESSAGE`
- `WARNING_MESSAGE`
- `QUESTION_MESSAGE`
- `PLAIN_MESSAGE`

189

Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace

Debug Java

Package Explorer

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

*ImageDemo.java

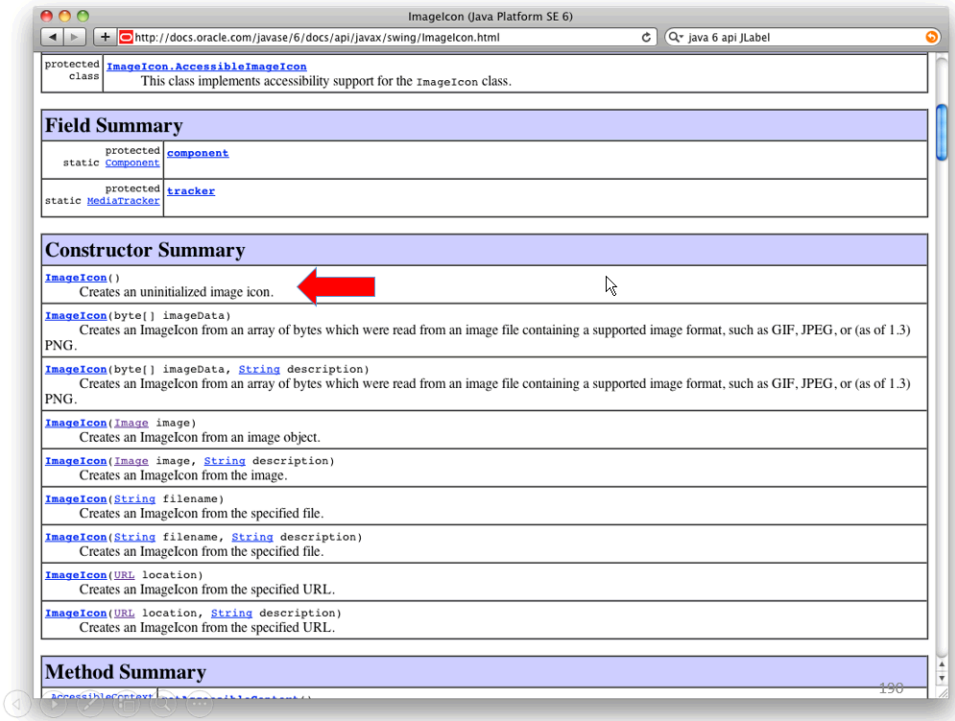
```
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.net.URL;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

public class ImageDemo {

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

            URL url = new URL("http://www.google.com/intl/en_ALL/images/logos/images_logo_lg.gif");
            BufferedImage image = ImageIO.read(url);
            JOptionPane.showMessageDialog(null, new JLabel(new ImageIcon(image)));

        } catch (IOException e) {
            e.printStackTrace();
        }
    }

}
```

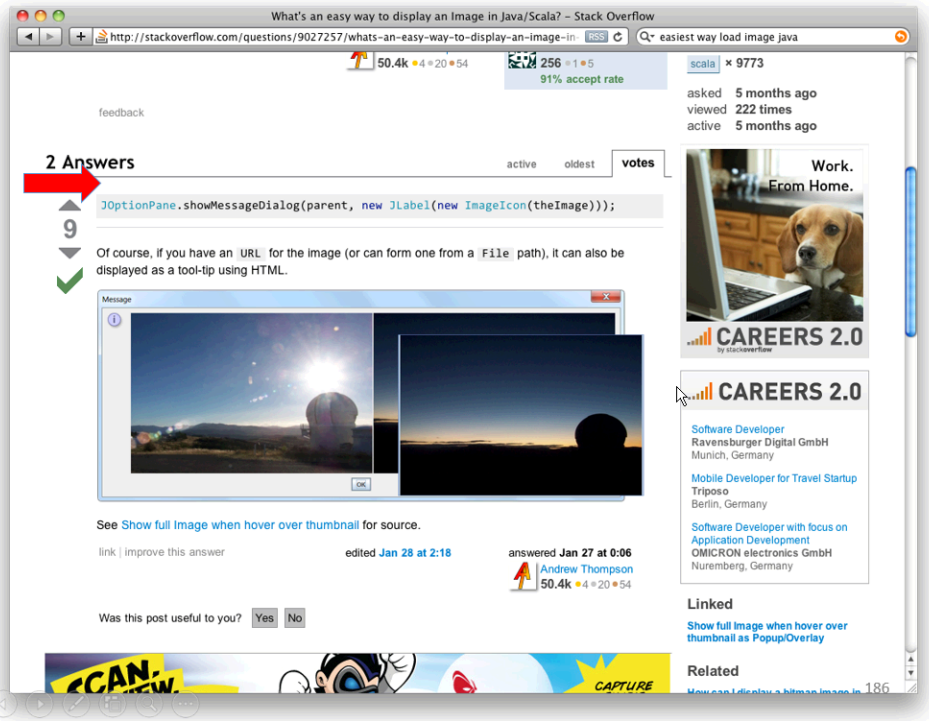
Problems Javadoc Declaration Console

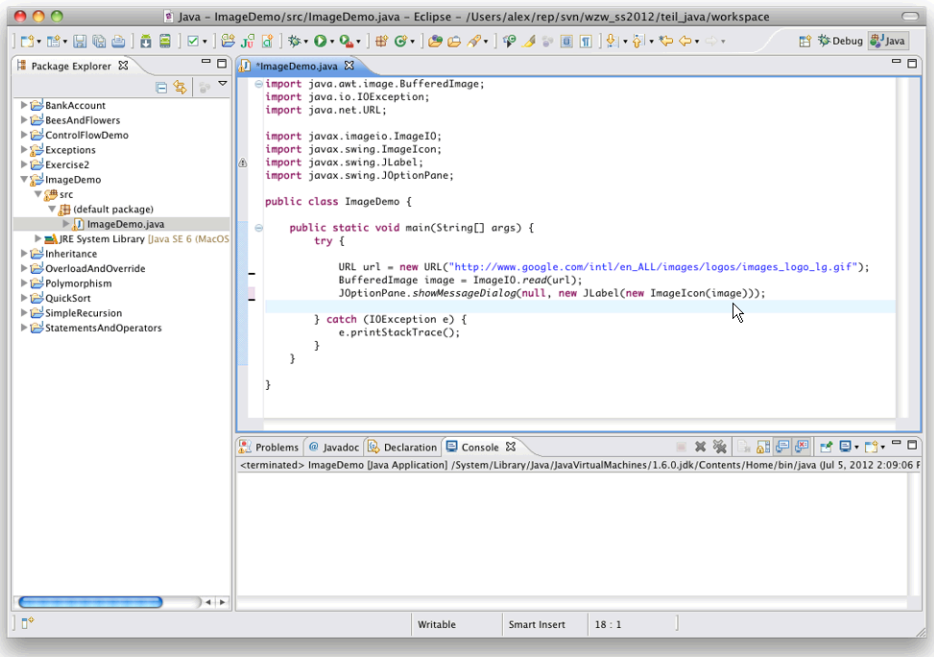
<terminated> ImageDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 5, 2012 2:09:06 F

Writable Smart Insert 18 : 1

192


ImageIcon (Java Platform SE 6)
http://docs.oracle.com/javase/6/docs/api/javax/swing/ImageIcon.html
java 6 api JLabel
protected class ImageIcon.AccessibleImageIcon
This class implements accessibility support for the ImageIcon class.
Field Summary
protected static Component component
protected static MediaTracker tracker
Constructor Summary
ImageIcon()
Creates an uninitialized image icon.
ImageIcon(byte[] imageData)
Creates an ImageIcon from an array of bytes which were read from an image file containing a supported image format, such as GIF, JPEG, or (as of 1.3) PNG.
ImageIcon(byte[] imageData, String description)
Creates an ImageIcon from an array of bytes which were read from an image file containing a supported image format, such as GIF, JPEG, or (as of 1.3) PNG.
ImageIcon(Image image)
Creates an ImageIcon from an image object.
ImageIcon(Image image, String description)
Creates an ImageIcon from the image.
ImageIcon(String filename)
Creates an ImageIcon from the specified file.
ImageIcon(String filename, String description)
Creates an ImageIcon from the specified file.
ImageIcon(URL location)
Creates an ImageIcon from the specified URL.
ImageIcon(URL location, String description)
Creates an ImageIcon from the specified URL.
Method Summary
190


What's an easy way to display an Image in Java/Scala? - Stack Overflow
http://stackoverflow.com/questions/9027257/whats-an-easy-way-to-display-an-image-in-
easiest way load image java
50.4k 4 20 54
256 1 5
91% accept rate
feedback
scala × 9773
asked 5 months ago
viewed 222 times
active 5 months ago
2 Answers
active oldest votes
JOptionPane.showMessageDialog(parent, new JLabel(new ImageIcon(theImage)));
9
Of course, if you have an URL for the image (or can form one from a File path), it can also be displayed as a tool-tip using HTML.
Message
OK
See Show full Image when hover over thumbnail for source.
link | improve this answer
edited Jan 28 at 2:18
answered Jan 27 at 0:06
Andrew Thompson
50.4k 4 20 54
Was this post useful to you? Yes No
Work. From Home.
CAREERS 2.0
by stackoverflow
CAREERS 2.0
Software Developer
Ravensburger Digital GmbH
Munich, Germany
Mobile Developer for Travel Startup
Triposo
Berlin, Germany
Software Developer with focus on Application Development
OMICRON electronics GmbH
Nuremberg, Germany
Linked
Show full Image when hover over thumbnail as Popup/Overlay
Related
186


Java - ImageDemo/src/ImageDemo.java - Eclipse - /Users/alex/rep/svn/wzw_ss2012/teil_java/workspace
Package Explorer
BankAccount
BeesAndFlowers
ControlFlowDemo
Exceptions
Exercise2
ImageDemo
src
(default package)
ImageDemo.java
JRE System Library [Java SE 6 (MacOS
Inheritance
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
*ImageDemo.java
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.net.URL;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
public class ImageDemo {
public static void main(String[] args) {
try {
URL url = new URL("http://www.google.com/intl/en_ALL/images/logos/images_logo_lg.gif");
BufferedImage image = ImageIO.read(url);
JOptionPane.showMessageDialog(null, new JLabel(new ImageIcon(image)));
} catch (IOException e) {
e.printStackTrace();
}
}
}
Problems Javadoc Declaration Console
<terminated> ImageDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Jul 5, 2012 2:09:06 F
Writable
Smart Insert
18 : 1