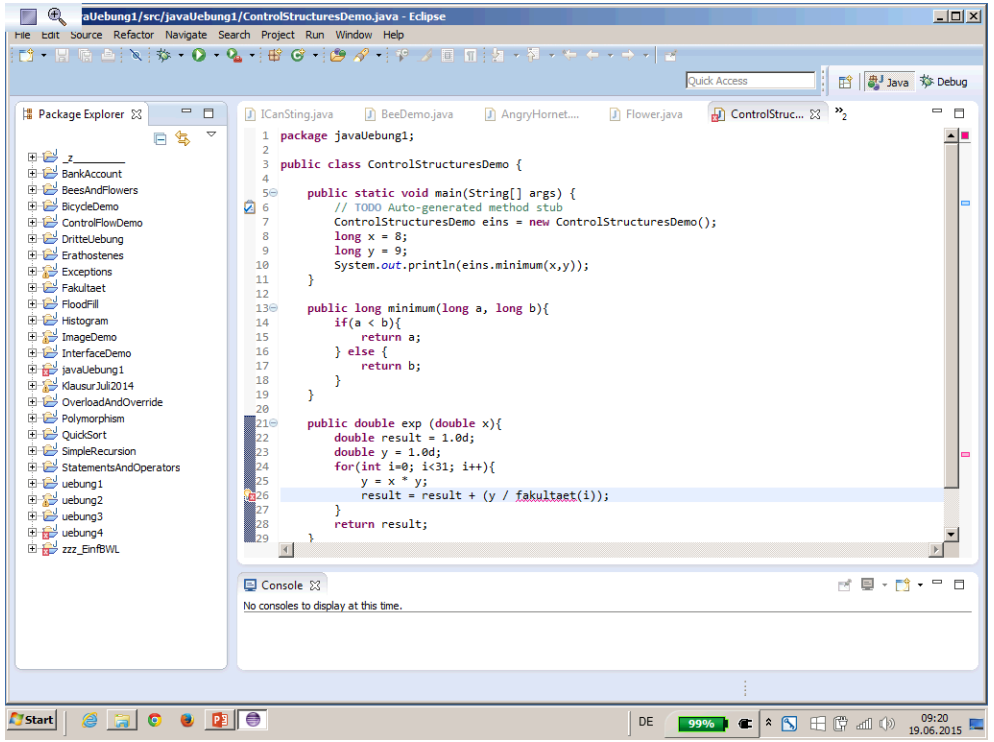
Script generated by TTT

Title: groh: profile1 (19.06.2015)

Date: Fri Jun 19 09:20:31 CEST 2015

Duration: 84:09 min

Pages: 79



Darstellung / Approximation von reellen Zahlen

```
float ggg  = -345545.34534E-12f;
double sss = 3245343455.555E67;
```

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

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

- mit beschränkter Anzahl Bits: Nur Approximation von reellen Zahlen (bspw. π) bzw. rationalen Zahlen mit nicht abbrechender Dezimalbruchentwicklung (bspw. 1/3) möglich. (Standard: IEEE 754).
- Folgen: Numerische Fehler möglich. Bsp:

2^{1000}
2^{5000}

```
double e = Math.pow(2.0d,1000.0d); //1.071508607186267 3E301
double f = Math.pow(2.0d,5000.0d); //Infinity   (Zahlbereichsüberschreitung)
double g = 0.1d + 0.1d + 0.1d; //0.30000000000000004   (Rundungsfehler)
```

siehe auch: http://introcs.cs.princeton.edu/java/91float/

- → Test auf Gleichheit zweier float oder double Werte x, y nicht mit $x == y$ sondern mit $|x - y| < \epsilon$;
  viele weitere Konsequenzen → numerische Mathematik
- Es gibt in Java auch Möglichkeiten mit beliebiger Genauigkeit zu rechnen (bspw. mit Klasse BigDecimal)

Darstellung / Approximation von reellen Zahlen

```
float ggg  = -345545.34534E-12f;
double sss = 3245343455.555E67;
```

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

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

- mit beschränkter Anzahl Bits: Nur Approximation von reellen Zahlen (bspw. π) bzw. rationalen Zahlen mit nicht abbrechender Dezimalbruchentwicklung (bspw. 1/3) möglich. (Standard: IEEE 754).
- Folgen: Numerische Fehler möglich. Bsp:

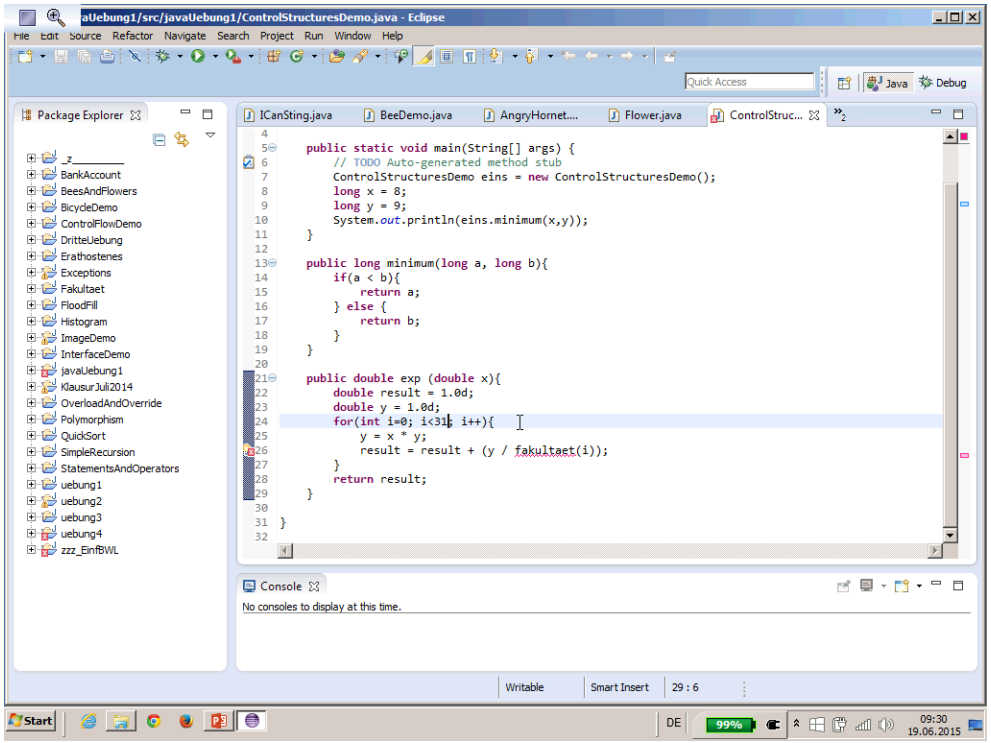
2^{1000}
2^{5000}

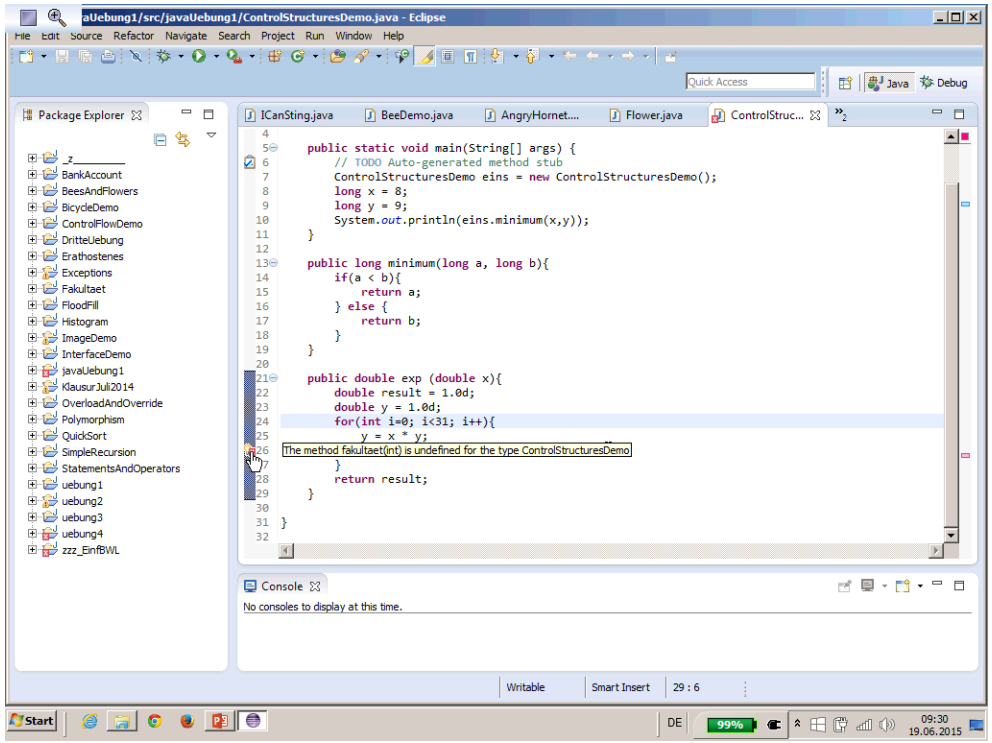
```
double e = Math.pow(2.0d,1000.0d); //1.071508607186267 3E301
double f = Math.pow(2.0d,5000.0d); //Infinity   (Zahlbereichsüberschreitung)
double g = 0.1d + 0.1d + 0.1d; //0.30000000000000004   (Rundungsfehler)
```

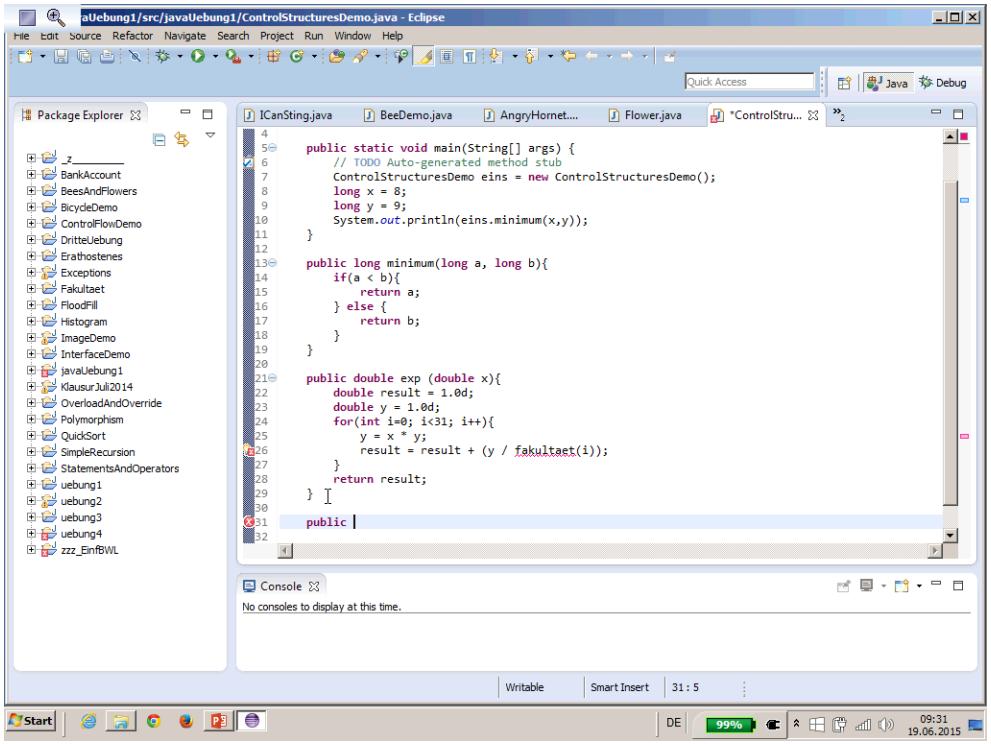
siehe auch: http://introcs.cs.princeton.edu/java/91float/

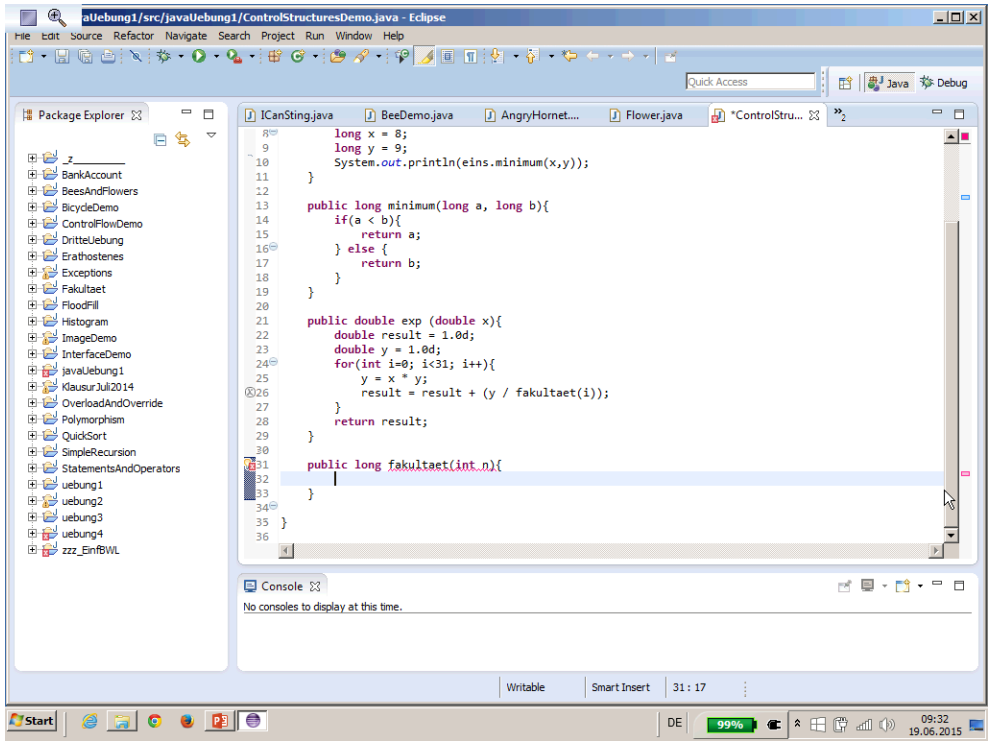
- → Test auf Gleichheit zweier float oder double Werte x, y nicht mit $x == y$ sondern mit $|x - y| < \epsilon$;
  viele weitere Konsequenzen → numerische Mathematik
- Es gibt in Java auch Möglichkeiten mit beliebiger Genauigkeit zu rechnen (bspw. mit Klasse BigDecimal)

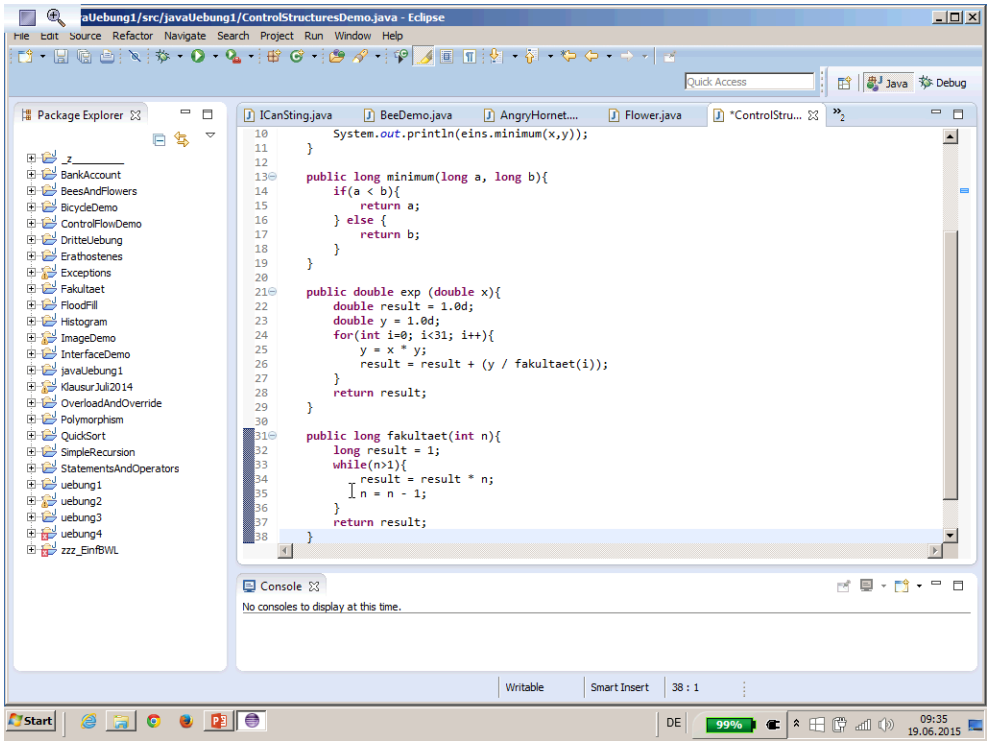
55

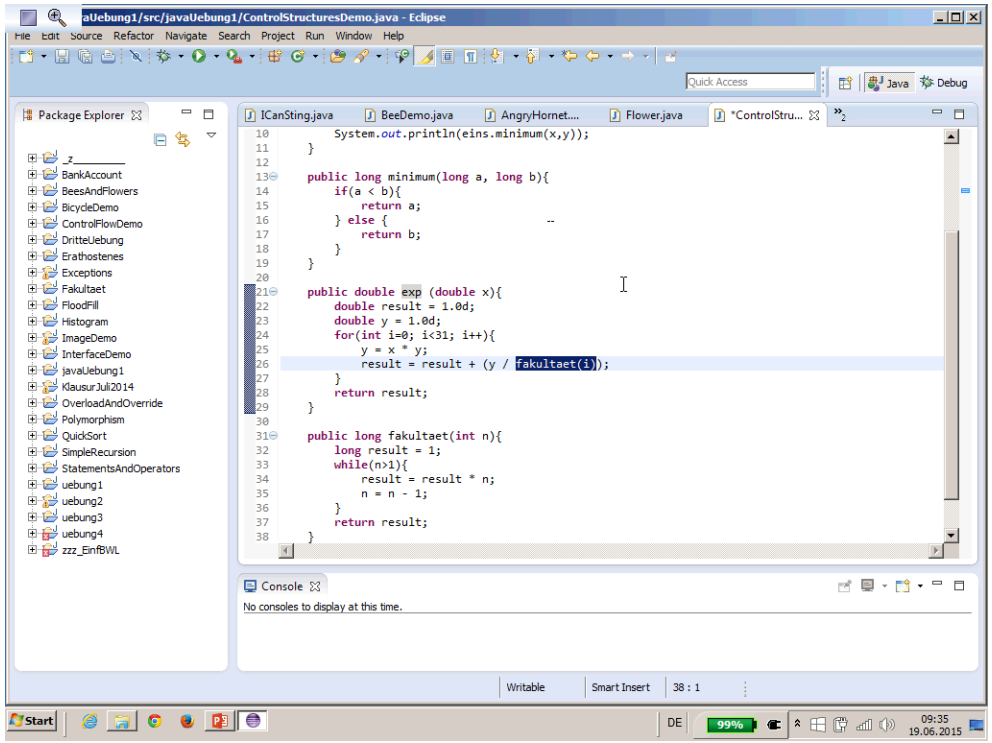

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
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 double y = 1.0d;
24 for(int i=0; i<31; i++){
25 y = x * y;
26 result = result + (y / fakultaet(i));
27 }
28 return result;
29 }
30
31 }
32
Console
No consoles to display at this time.
Writable
Smart Insert
29 : 6
Start
DE
99%
09:30
19.06.2015

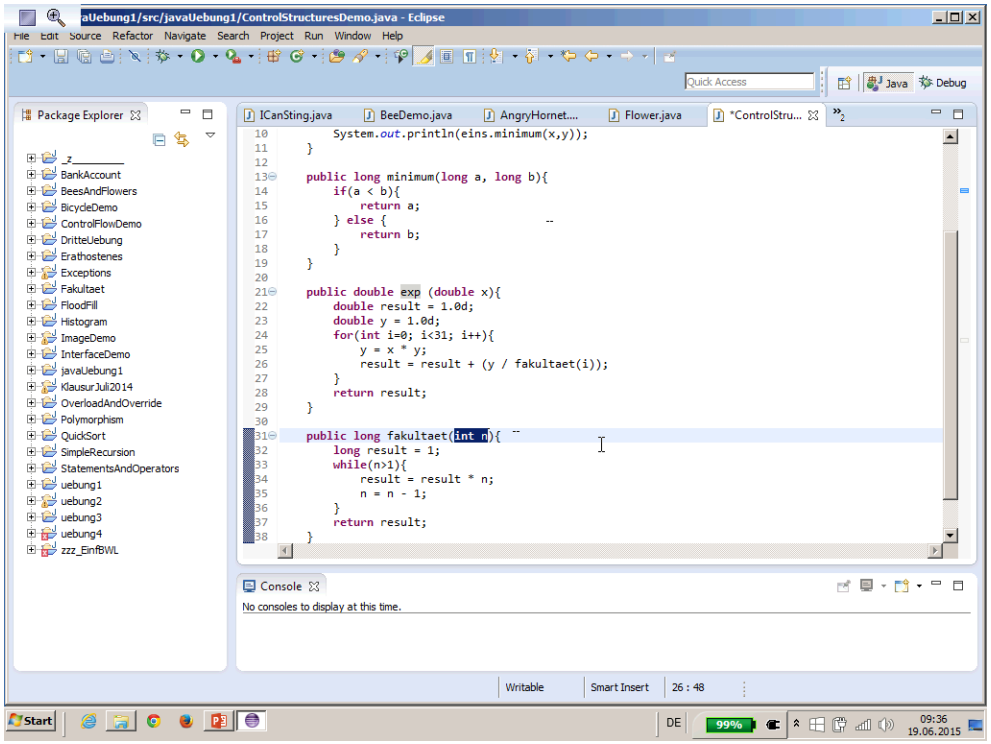

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
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 double y = 1.0d;
24 for(int i=0; i<31; i++){
25 y = x * y;
26 The method fakultaet(int) is undefined for the type ControlStructuresDemo
27 }
28 return result;
29 }
30
31 }
32
Console
No consoles to display at this time.
Writable
Smart Insert
29 : 6
Start
DE
99%
09:30
19.06.2015

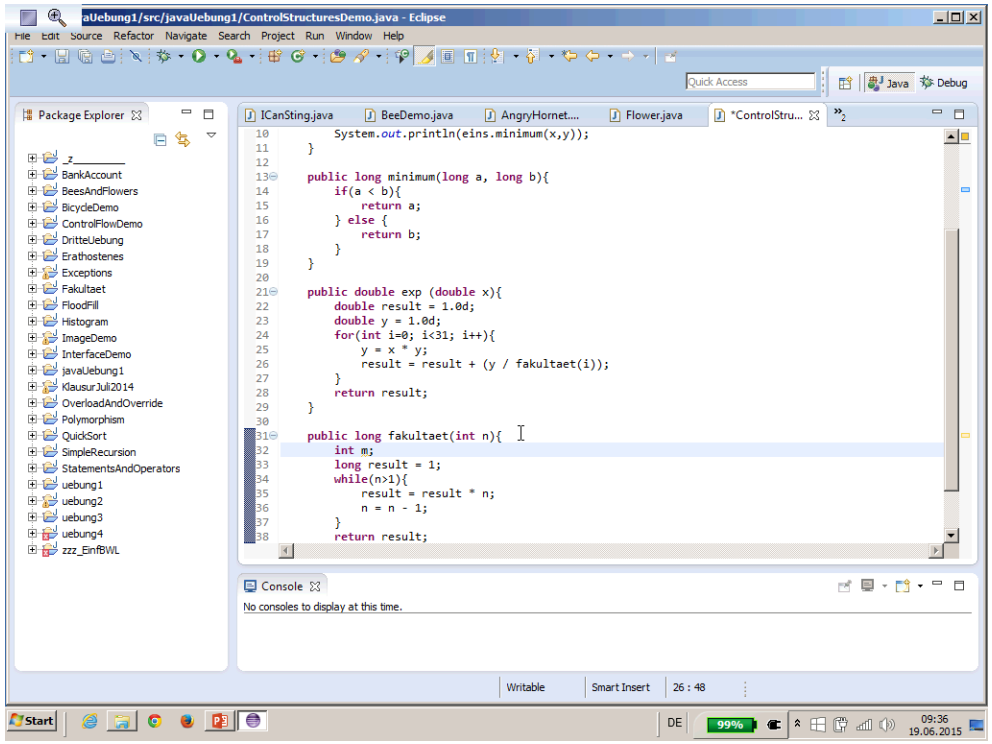

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
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 double y = 1.0d;
24 for(int i=0; i<31; i++){
25 y = x * y;
26 result = result + (y / fakultaet(i));
27 }
28 return result;
29 }
30
31 public
32
Console
No consoles to display at this time.
Writable
Smart Insert
31 : 5
Start
DE
99%
09:31
19.06.2015

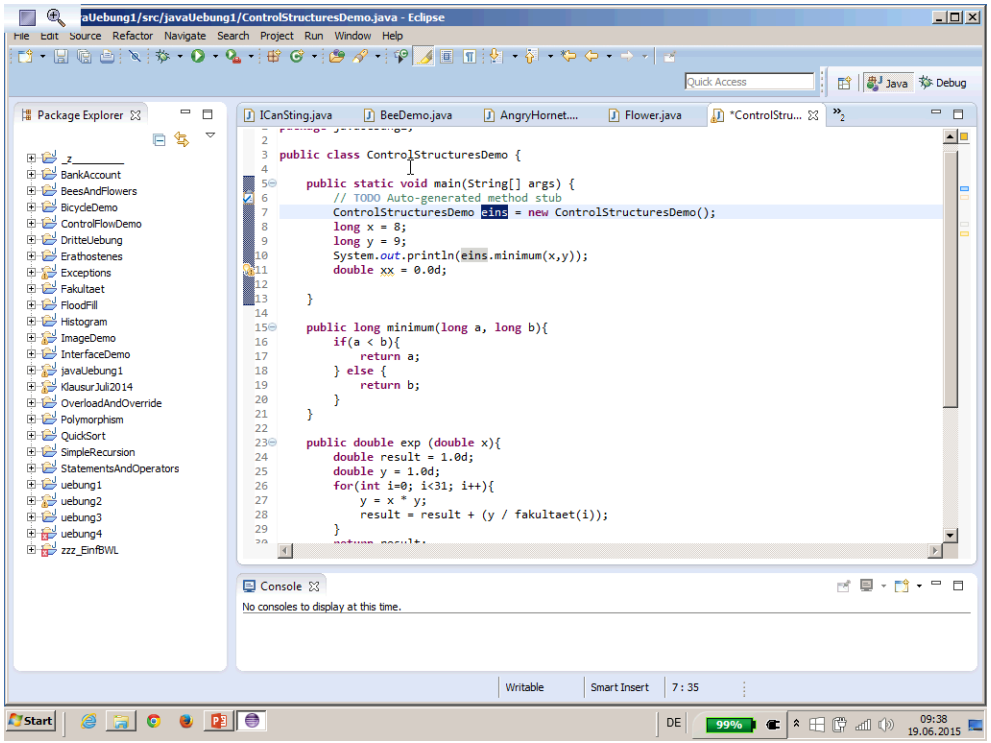

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
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 + (y / fakultaet(i));
27 }
28 return result;
29 }
30
31 public long fakultaet(int n){
32
33 }
34
35 }
36
Console
No consoles to display at this time.
Writable
Smart Insert
31 : 17
Start
DE
99%
09:32
19.06.2015

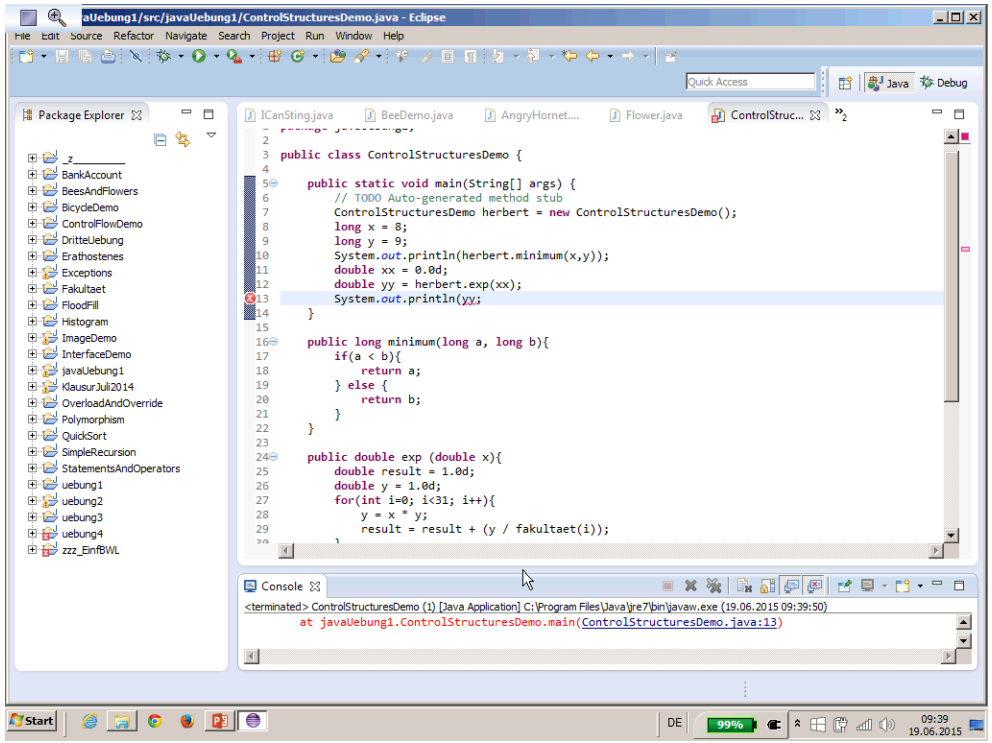

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public 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 + (y / fakultaet(i));
27 }
28 return result;
29 }
30
31 public long fakultaet(int n){
32 long result = 1;
33 while(n>1){
34 result = result * n;
35 n = n - 1;
36 }
37 return result;
38 }
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
Console
No consoles to display at this time.
Writable
Smart Insert
38 : 1
Start
DE
99%
09:35
19.06.2015

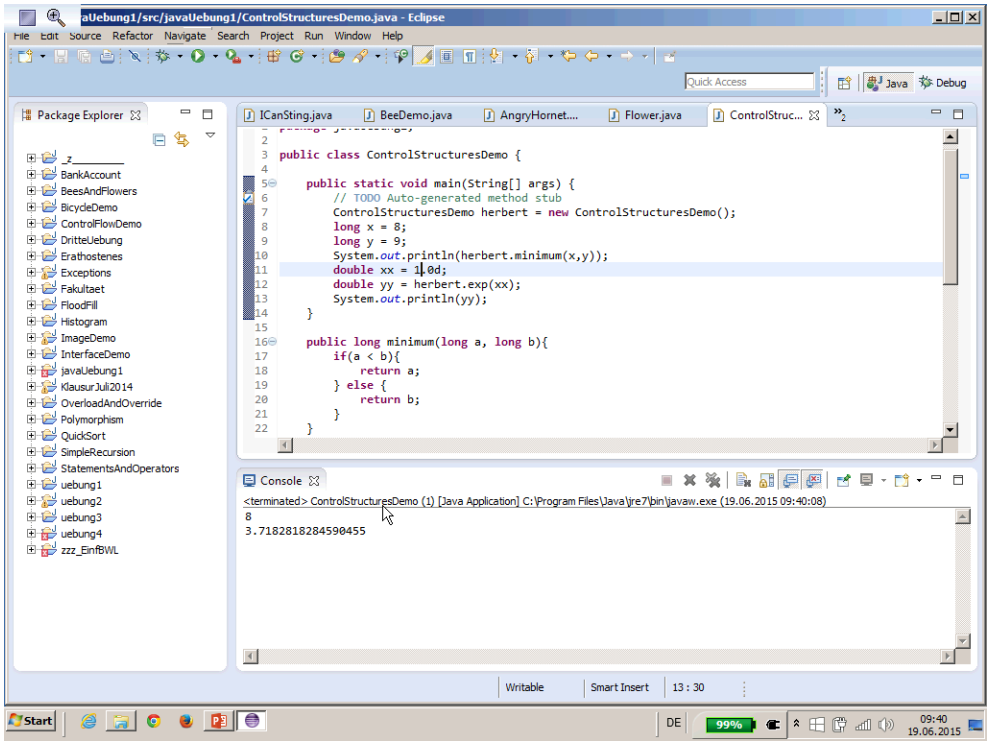

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public 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 + (y / fakultaet(i));
27 }
28 return result;
29 }
30
31 public long fakultaet(int n){
32 long result = 1;
33 while(n>1){
34 result = result * n;
35 n = n - 1;
36 }
37 return result;
38 }
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
Console
No consoles to display at this time.
Writable
Smart Insert
38 : 1
Start
DE
99%
09:35
19.06.2015

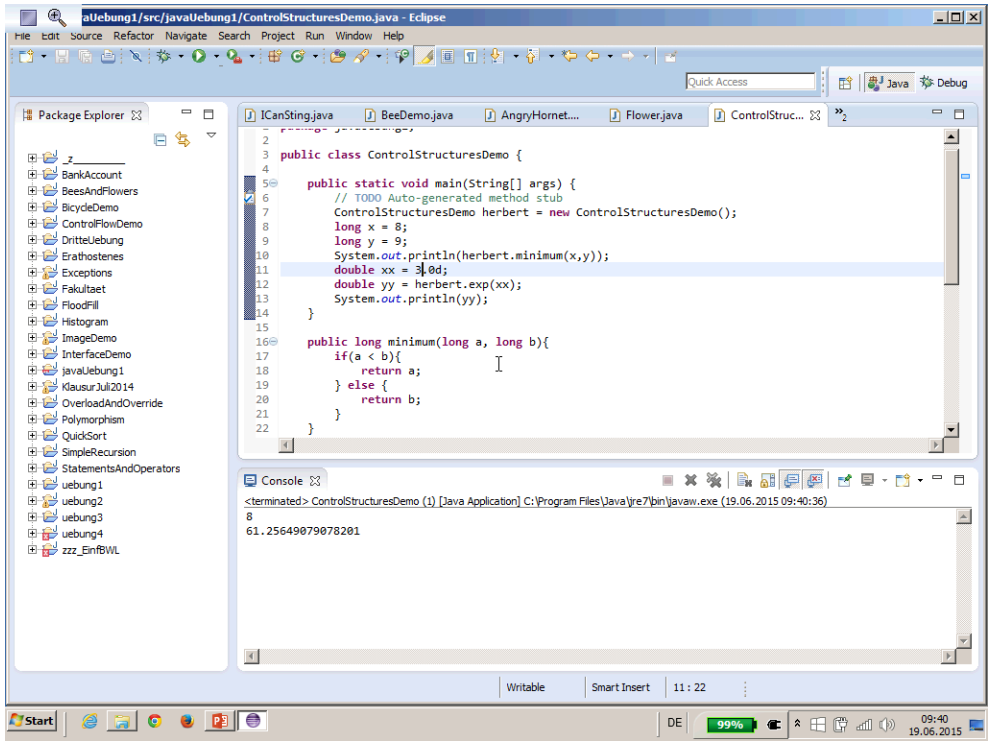

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public 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 + (y / fakultaet(i));
27 }
28 return result;
29 }
30
31 public long fakultaet(int n){
32 long result = 1;
33 while(n>1){
34 result = result * n;
35 n = n - 1;
36 }
37 return result;
38 }
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
Console
No consoles to display at this time.
Writable
Smart Insert
26 : 48
Start
DE
99%
09:36
19.06.2015

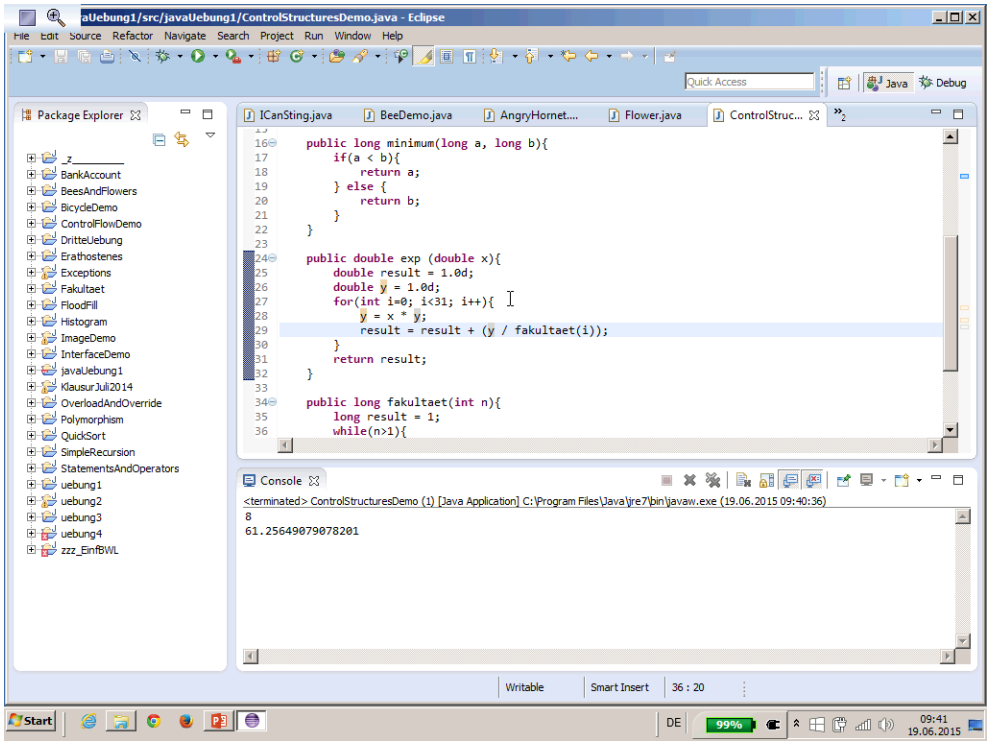

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
10 System.out.println(eins.minimum(x,y));
11 }
12
13 public 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 + (y / fakultaet(i));
27 }
28 return result;
29 }
30
31 public long fakultaet(int n){
32 int m;
33 long result = 1;
34 while(n>1){
35 result = result * n;
36 n = n - 1;
37 }
38 return result;
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
Console
No consoles to display at this time.
Writable
Smart Insert
26 : 48
Start
DE
99%
09:36
19.06.2015

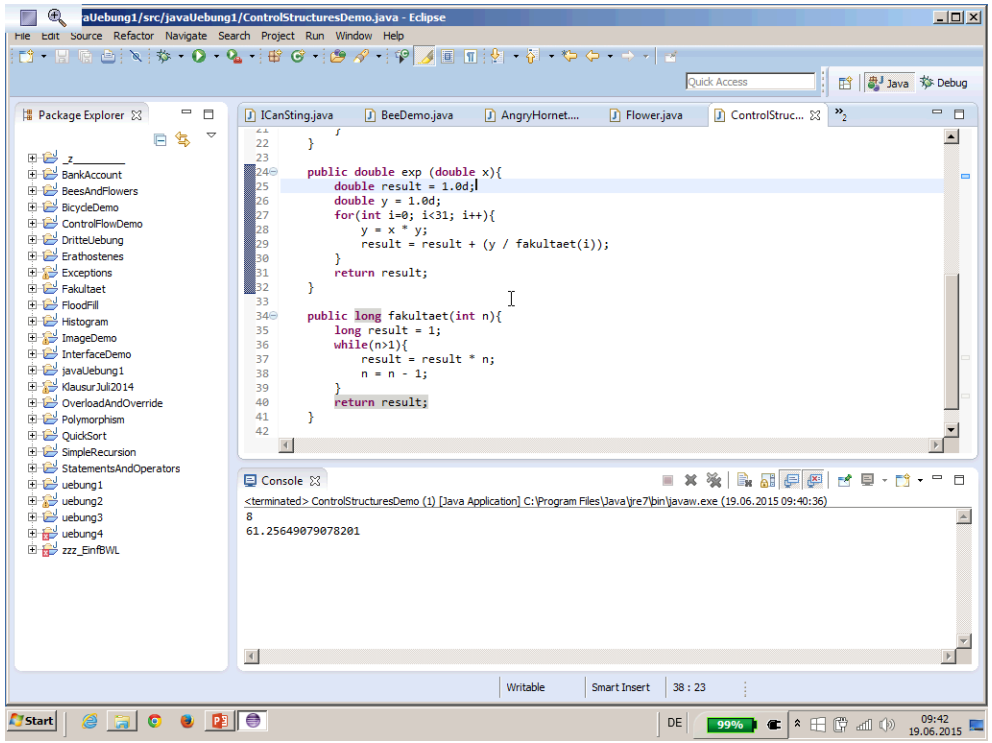

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
public class ControlStructuresDemo {
public static void main(String[] args) {
// TODO Auto-generated method stub
ControlStructuresDemo eins = new ControlStructuresDemo();
long x = 8;
long y = 9;
System.out.println(eins.minimum(x,y));
double xx = 0.0d;
}
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 / fakultaet(i));
}
Console
No consoles to display at this time.
Writable
Smart Insert
7 : 35
09:38
19.06.2015

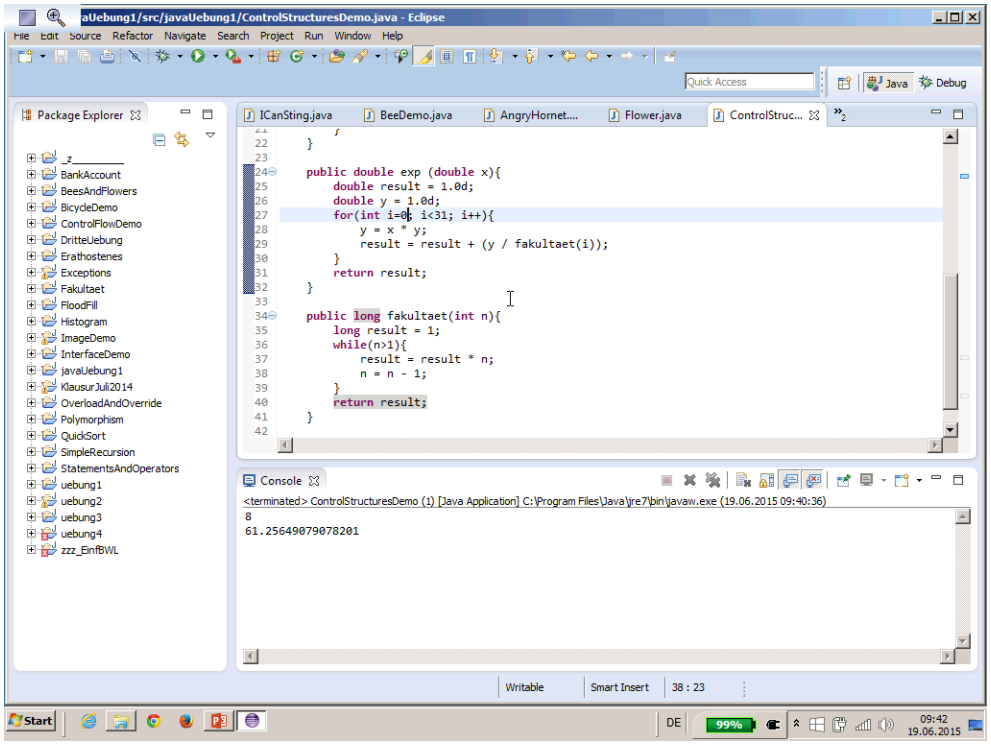

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
public class ControlStructuresDemo {
public static void main(String[] args) {
// TODO Auto-generated method stub
ControlStructuresDemo herbert = new ControlStructuresDemo();
long x = 8;
long y = 9;
System.out.println(herbert.minimum(x,y));
double xx = 0.0d;
double yy = herbert.exp(xx);
System.out.println(yy;
}
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 / fakultaet(i));
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:39:50)
at javaUebung1.ControlStructuresDemo.main(ControlStructuresDemo.java:13)
09:39
19.06.2015

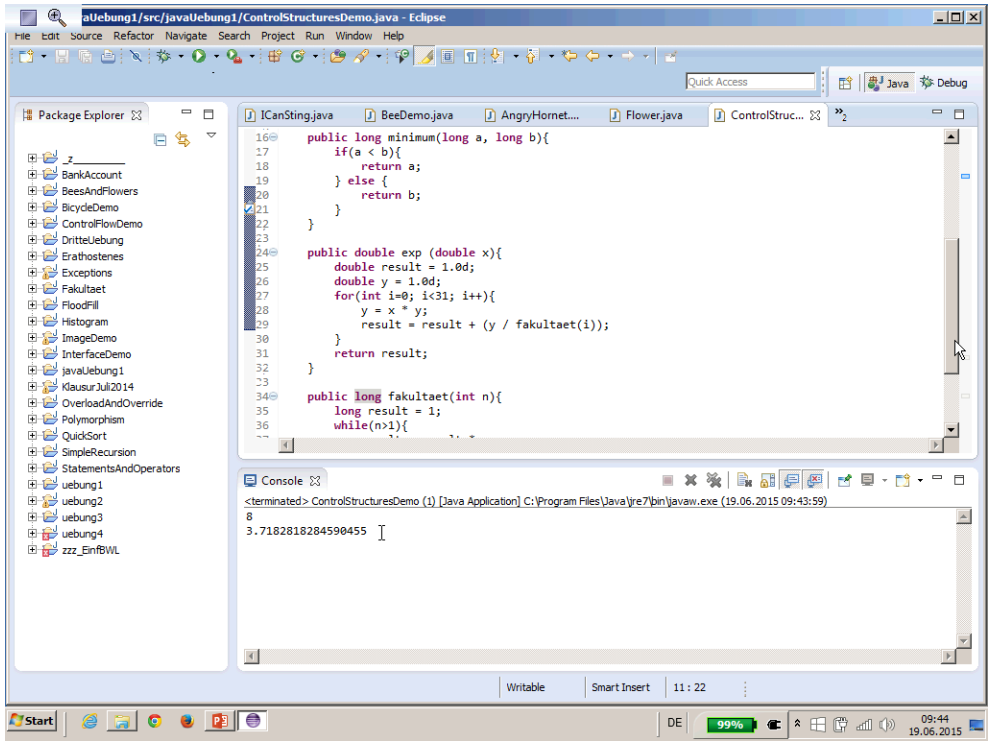

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
public class ControlStructuresDemo {
public static void main(String[] args) {
// TODO Auto-generated method stub
ControlStructuresDemo herbert = new ControlStructuresDemo();
long x = 8;
long y = 9;
System.out.println(herbert.minimum(x,y));
double xx = 1.0d;
double yy = herbert.exp(xx);
System.out.println(yy);
}
public long minimum(long a, long b){
if(a < b){
return a;
} else {
return b;
}
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:40:08)
8
3.7182818284590455
Writable
Smart Insert
13 : 30
09:40
19.06.2015

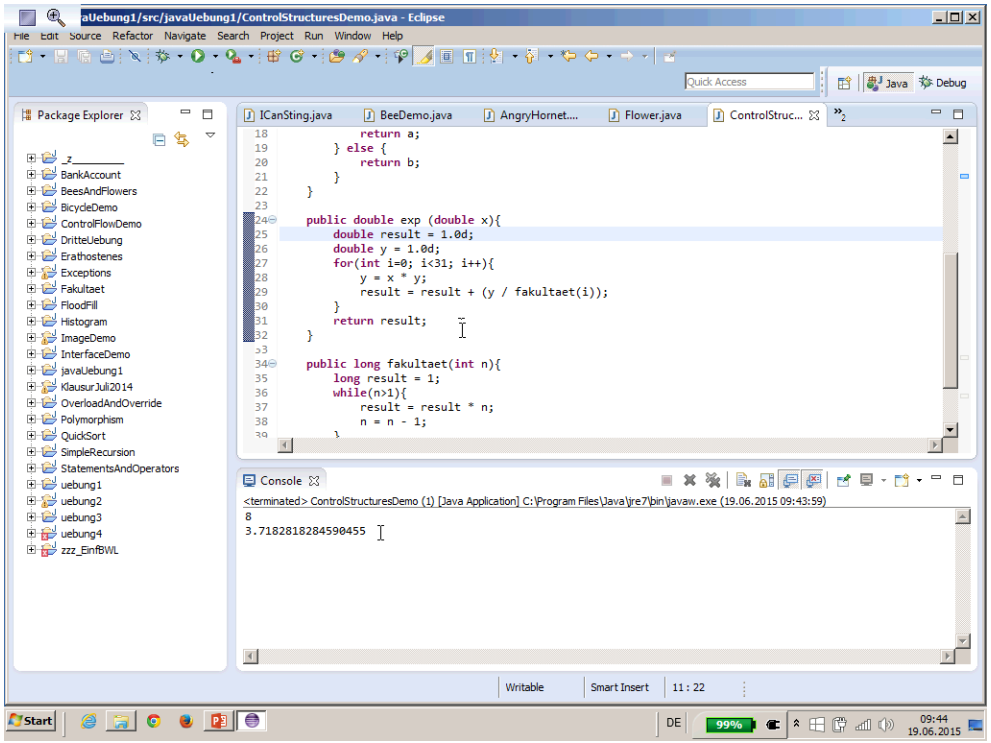

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
public class ControlStructuresDemo {
public static void main(String[] args) {
// TODO Auto-generated method stub
ControlStructuresDemo herbert = new ControlStructuresDemo();
long x = 8;
long y = 9;
System.out.println(herbert.minimum(x,y));
double xx = 3.0d;
double yy = herbert.exp(xx);
System.out.println(yy);
}
public long minimum(long a, long b){
if(a < b){
return a;
} else {
return b;
}
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:40:36)
8
61.25649079078201
Writable
Smart Insert
11 : 22
09:40
19.06.2015

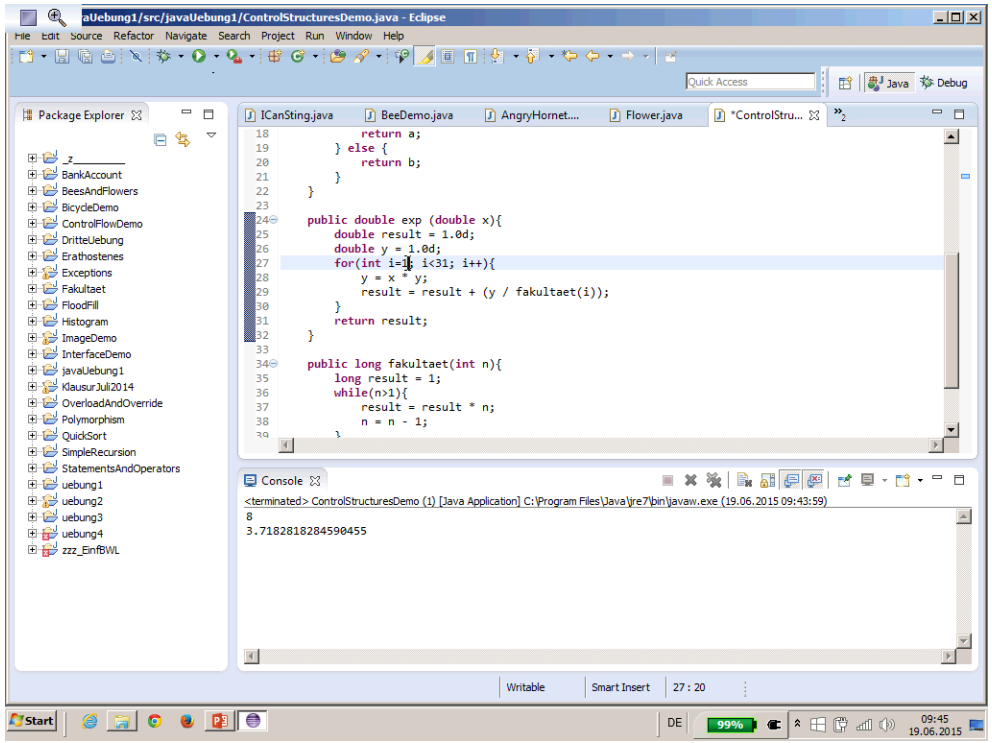

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
16 public long minimum(long a, long b){
17 if(a < b){
18 return a;
19 } else {
20 return b;
21 }
22 }
23
24 public double exp (double x){
25 double result = 1.0d;
26 double y = 1.0d;
27 for(int i=0; i<31; i++){
28 y = x * y;
29 result = result + (y / fakultaet(i));
30 }
31 return result;
32 }
33
34 public long fakultaet(int n){
35 long result = 1;
36 while(n>1){
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:40:36)
8
61.25649079078201
Writable
Smart Insert
36 : 20
09:41
19.06.2015

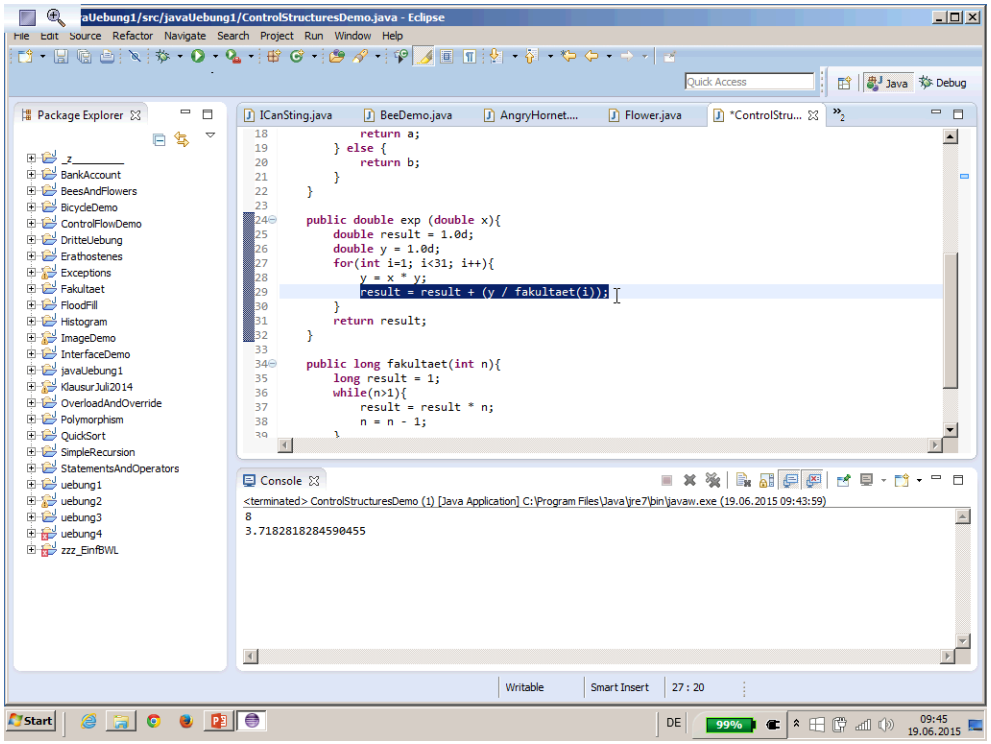

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
22 }
23
24 public double exp (double x){
25 double result = 1.0d;
26 double y = 1.0d;
27 for(int i=0; i<31; i++){
28 y = x * y;
29 result = result + (y / fakultaet(i));
30 }
31 return result;
32 }
33
34 public long fakultaet(int n){
35 long result = 1;
36 while(n>1){
37 result = result * n;
38 n = n - 1;
39 }
40 return result;
41 }
42
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:40:36)
8
61.25649079078201
Writable
Smart Insert
38 : 23
09:42
19.06.2015

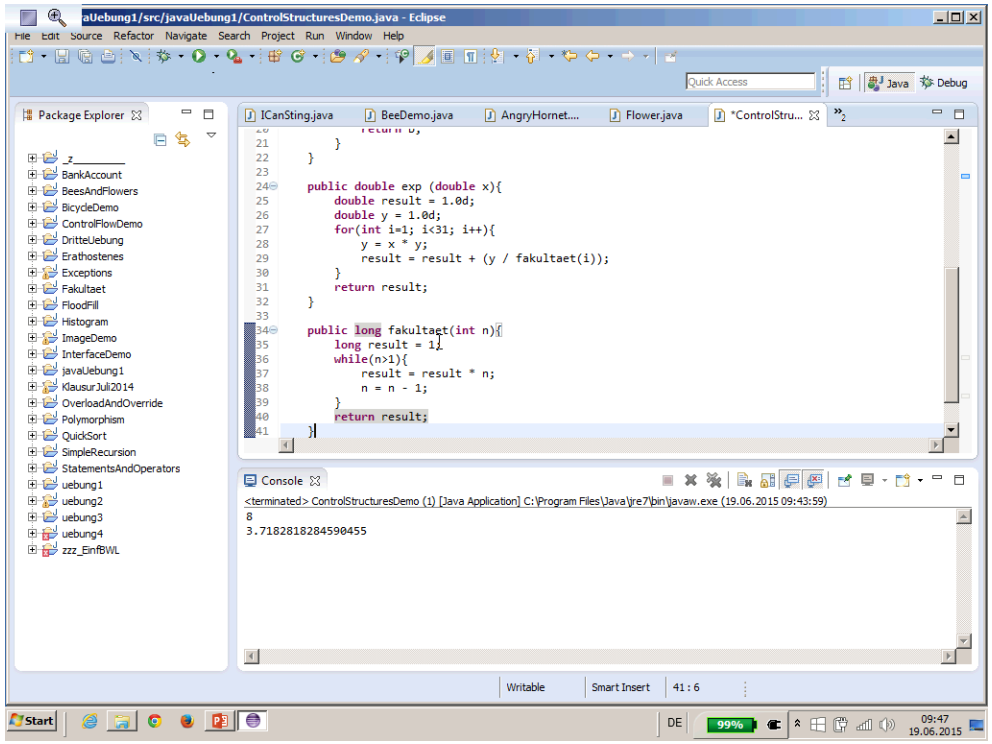

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
22 }
23
24 public double exp (double x){
25 double result = 1.0d;
26 double y = 1.0d;
27 for(int i=0; i<31; i++){
28 y = x * y;
29 result = result + (y / fakultaet(i));
30 }
31 return result;
32 }
33
34 public long fakultaet(int n){
35 long result = 1;
36 while(n>1){
37 result = result * n;
38 n = n - 1;
39 }
40 return result;
41 }
42
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:40:36)
8
61.25649079078201
Writable
Smart Insert
38 : 23
09:42
19.06.2015

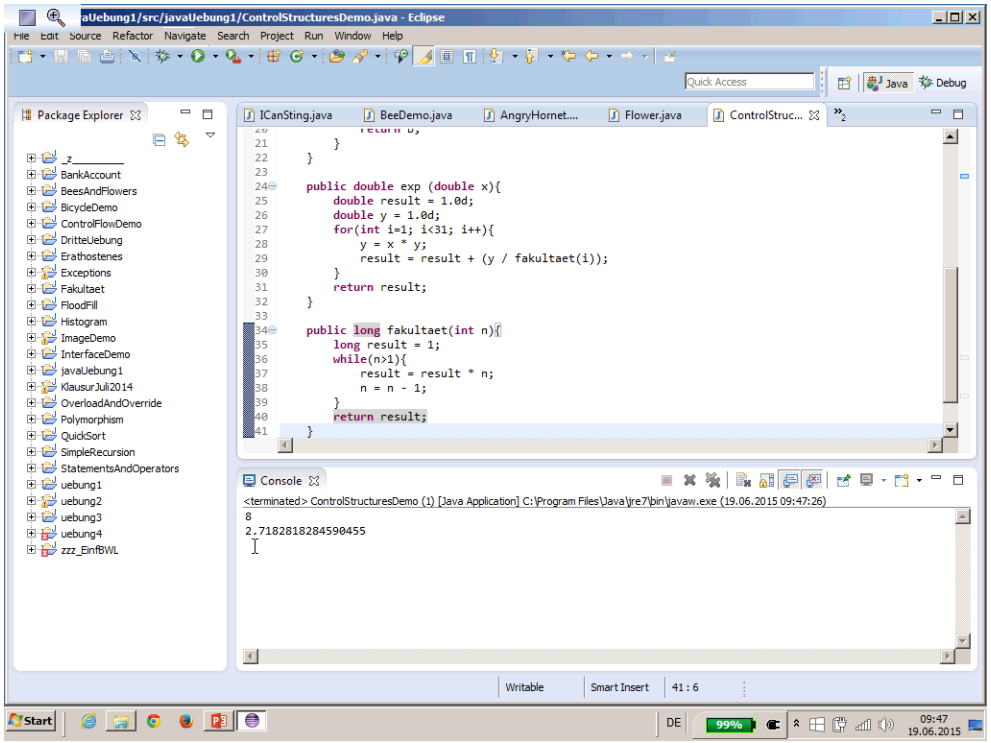

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
16 public long minimum(long a, long b){
17 if(a < b){
18 return a;
19 } else {
20 return b;
21 }
22 }
23
24 public double exp (double x){
25 double result = 1.0d;
26 double y = 1.0d;
27 for(int i=0; i<31; i++){
28 y = x * y;
29 result = result + (y / fakultaet(i));
30 }
31 return result;
32 }
33
34 public long fakultaet(int n){
35 long result = 1;
36 while(n>1){
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:43:59)
8
3.7182818284590455
Writable
Smart Insert
11 : 22
09:44
19.06.2015

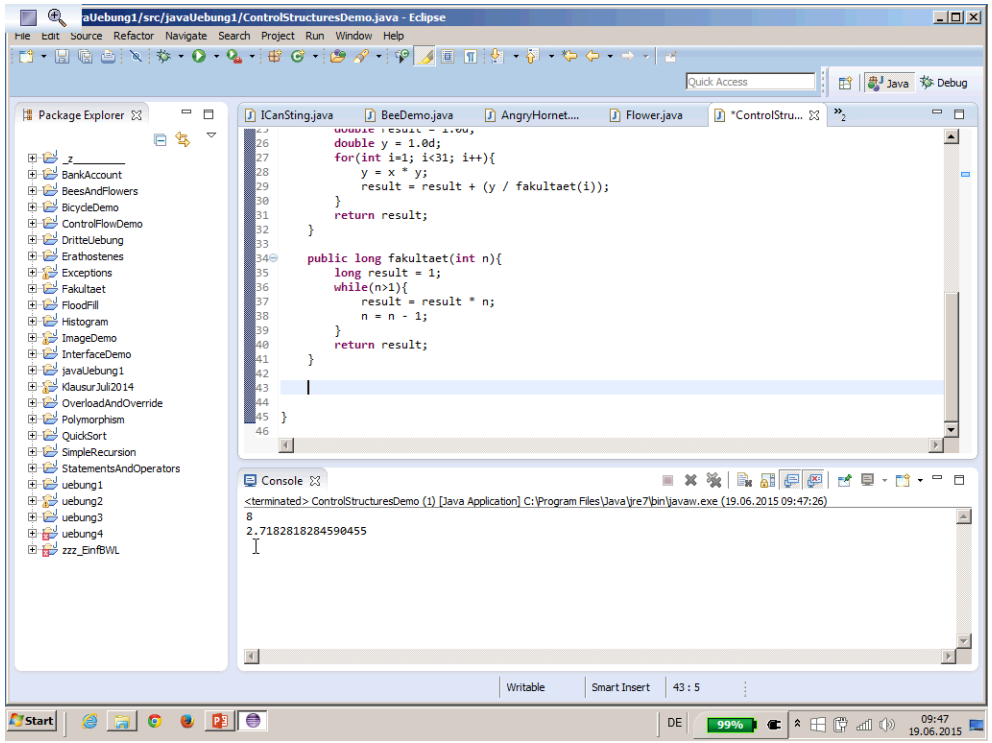

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
18 return a;
19 } else {
20 return b;
21 }
22 }
23
24 public double exp (double x){
25 double result = 1.0d;
26 double y = 1.0d;
27 for(int i=0; i<31; i++){
28 y = x * y;
29 result = result + (y / fakultaet(i));
30 }
31 return result;
32 }
33
34 public long fakultaet(int n){
35 long result = 1;
36 while(n>1){
37 result = result * n;
38 n = n - 1;
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:43:59)
8
3.7182818284590455
Writable
Smart Insert
11 : 22
Start
DE
99%
09:44
19.06.2015

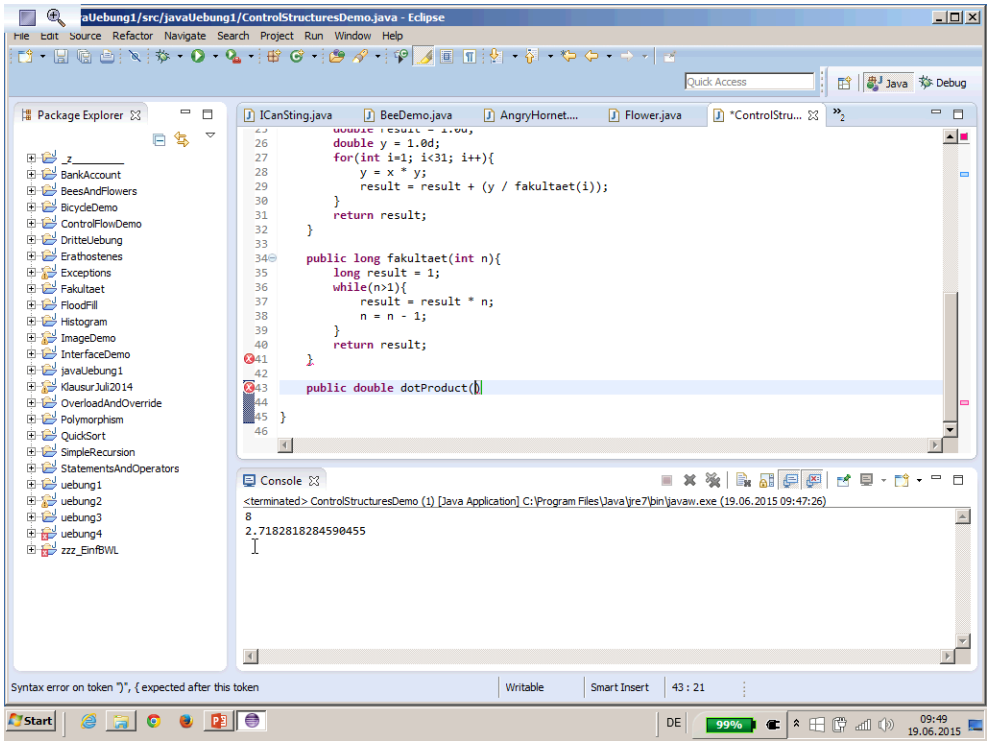

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
18 return a;
19 } else {
20 return b;
21 }
22 }
23
24 public double exp (double x){
25 double result = 1.0d;
26 double y = 1.0d;
27 for(int i=1; i<31; i++){
28 y = x * y;
29 result = result + (y / fakultaet(i));
30 }
31 return result;
32 }
33
34 public long fakultaet(int n){
35 long result = 1;
36 while(n>1){
37 result = result * n;
38 n = n - 1;
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:43:59)
8
3.7182818284590455
Writable
Smart Insert
27 : 20
Start
DE
99%
09:45
19.06.2015

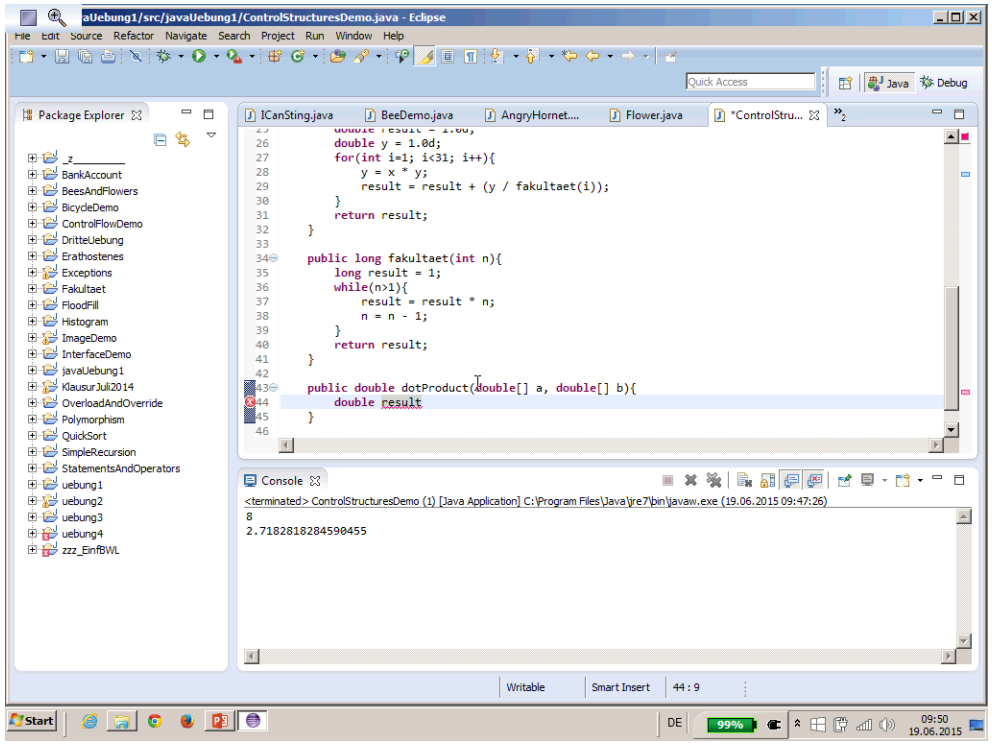

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
18 return a;
19 } else {
20 return b;
21 }
22 }
23
24 public double exp (double x){
25 double result = 1.0d;
26 double y = 1.0d;
27 for(int i=1; i<31; i++){
28 y = x * y;
29 result = result + (y / fakultaet(i));
30 }
31 return result;
32 }
33
34 public long fakultaet(int n){
35 long result = 1;
36 while(n>1){
37 result = result * n;
38 n = n - 1;
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:43:59)
8
3.7182818284590455
Writable
Smart Insert
27 : 20
Start
DE
99%
09:45
19.06.2015

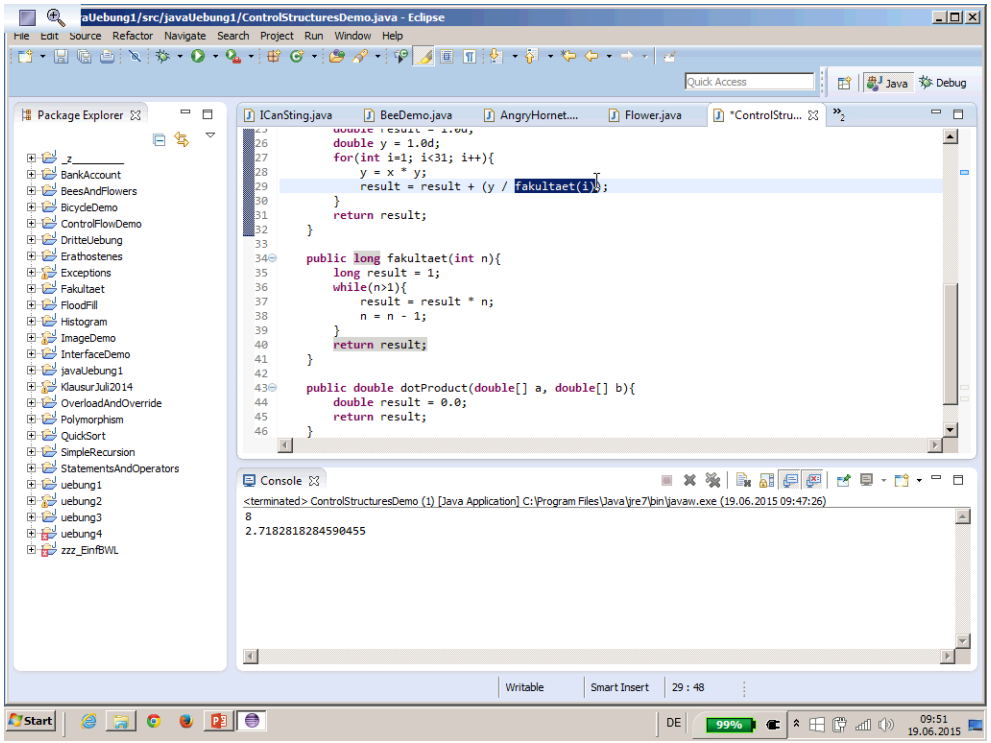

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
21 }
22 }
23
24 public double exp (double x){
25 double result = 1.0d;
26 double y = 1.0d;
27 for(int i=1; i<31; i++){
28 y = x * y;
29 result = result + (y / fakultaet(i));
30 }
31 return result;
32 }
33
34 public long fakultaet(int n){
35 long result = 1;
36 while(n>1){
37 result = result * n;
38 n = n - 1;
39 }
40 return result;
41 }
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:43:59)
8
3.7182818284590455
Writable
Smart Insert
41 : 6
Start
DE
99%
09:47
19.06.2015

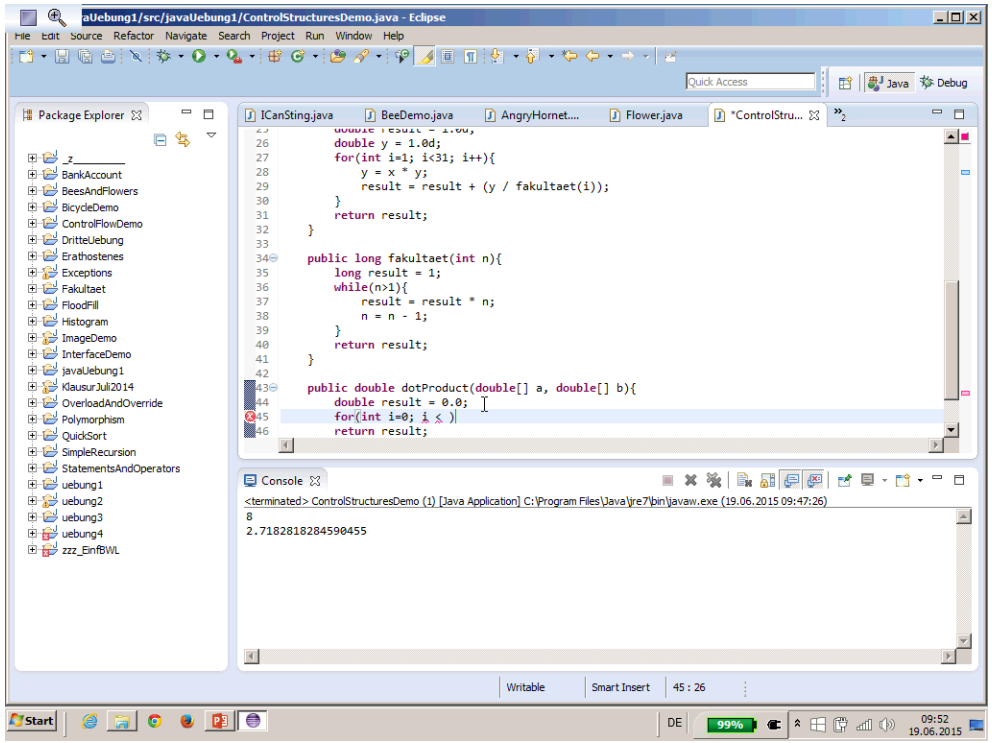

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
}
}
public double exp (double x){
double result = 1.0d;
double y = 1.0d;
for(int i=1; i<31; i++){
y = x * y;
result = result + (y / fakultaet(i));
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
41 : 6
Start
DE
99%
09:47
19.06.2015

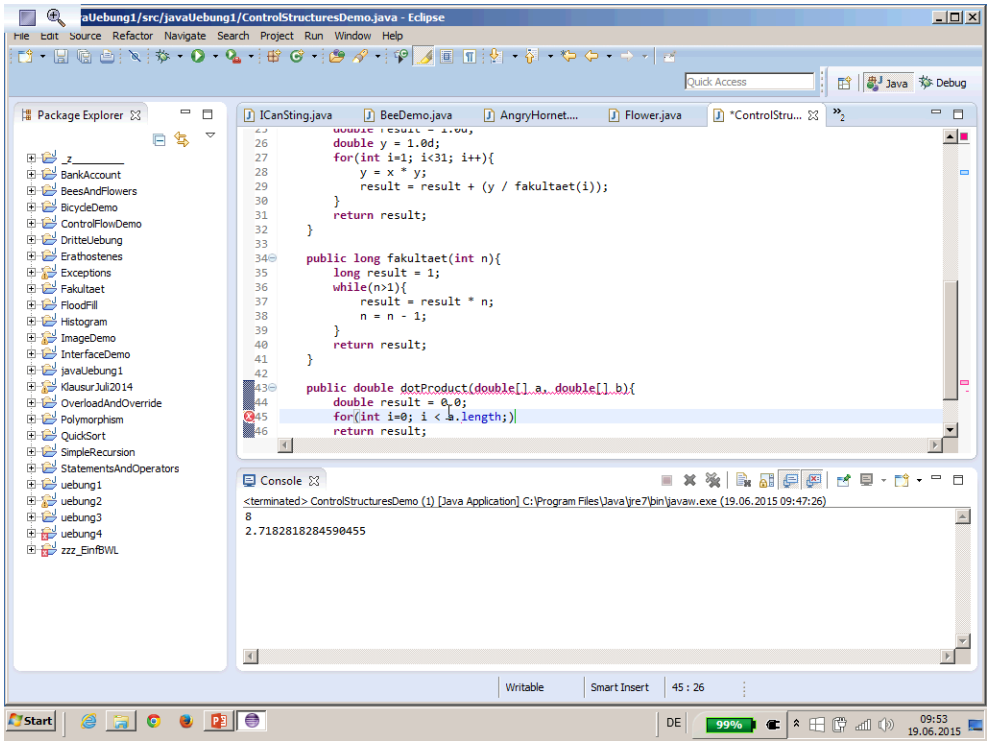

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
double y = 1.0d;
for(int i=1; i<31; i++){
y = x * y;
result = result + (y / fakultaet(i));
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
43 : 5
Start
DE
99%
09:47
19.06.2015

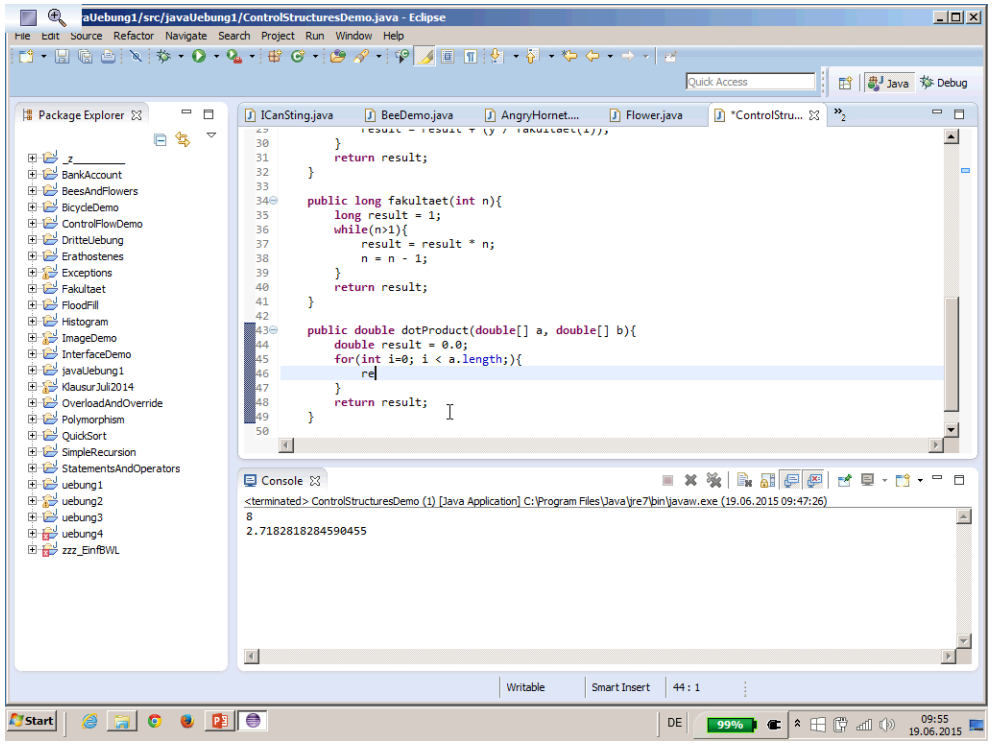

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
double y = 1.0d;
for(int i=1; i<31; i++){
y = x * y;
result = result + (y / fakultaet(i));
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct()
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Syntax error on token ")", { expected after this token
Writable
Smart Insert
43 : 21
Start
DE
99%
09:49
19.06.2015

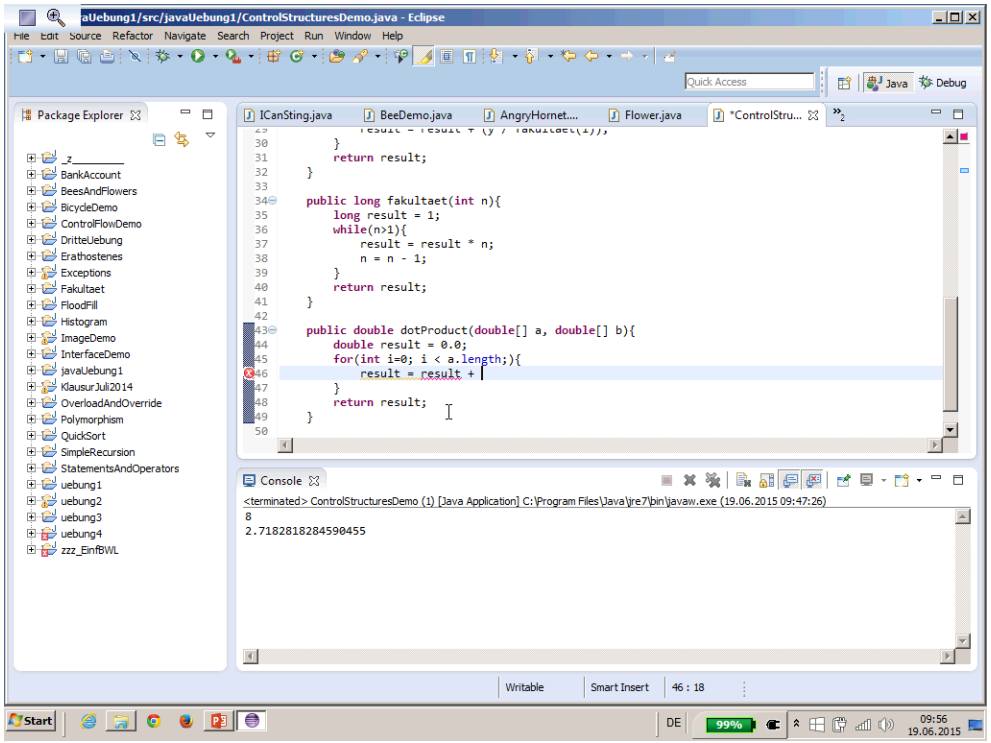

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
double y = 1.0d;
for(int i=1; i<31; i++){
y = x * y;
result = result + (y / fakultaet(i));
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
44 : 9
Start
DE
99%
09:50
19.06.2015

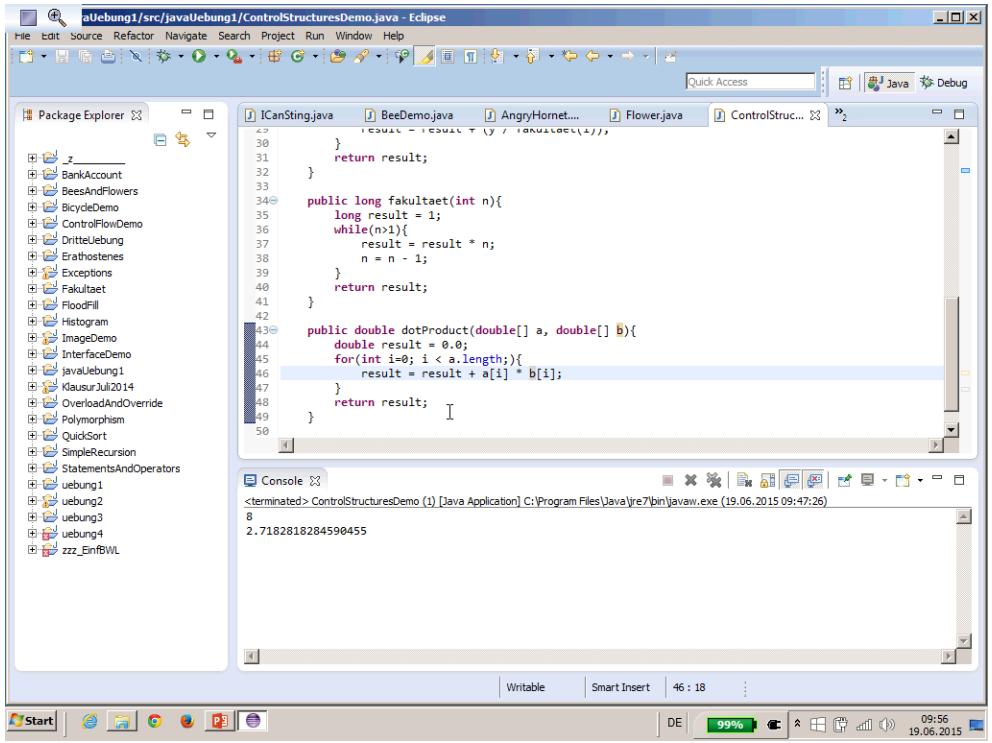

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
double y = 1.0d;
for(int i=1; i<31; i++){
y = x * y;
result = result + (y / fakultaet(i));
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
return result;
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
29 : 48
Start
DE
99%
09:51
19.06.2015

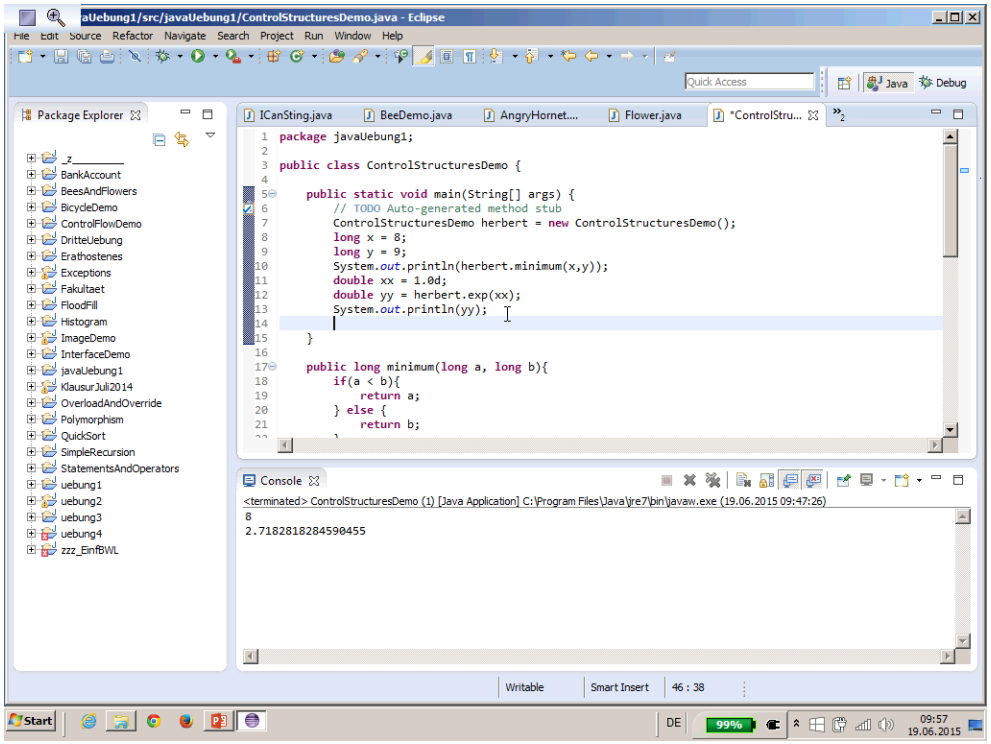

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
double y = 1.0d;
for(int i=1; i<31; i++){
y = x * y;
result = result + (y / fakultaet(i));
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
for(int i=0; i <);
return result;
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
45 : 26
Start
DE
99%
09:52
19.06.2015

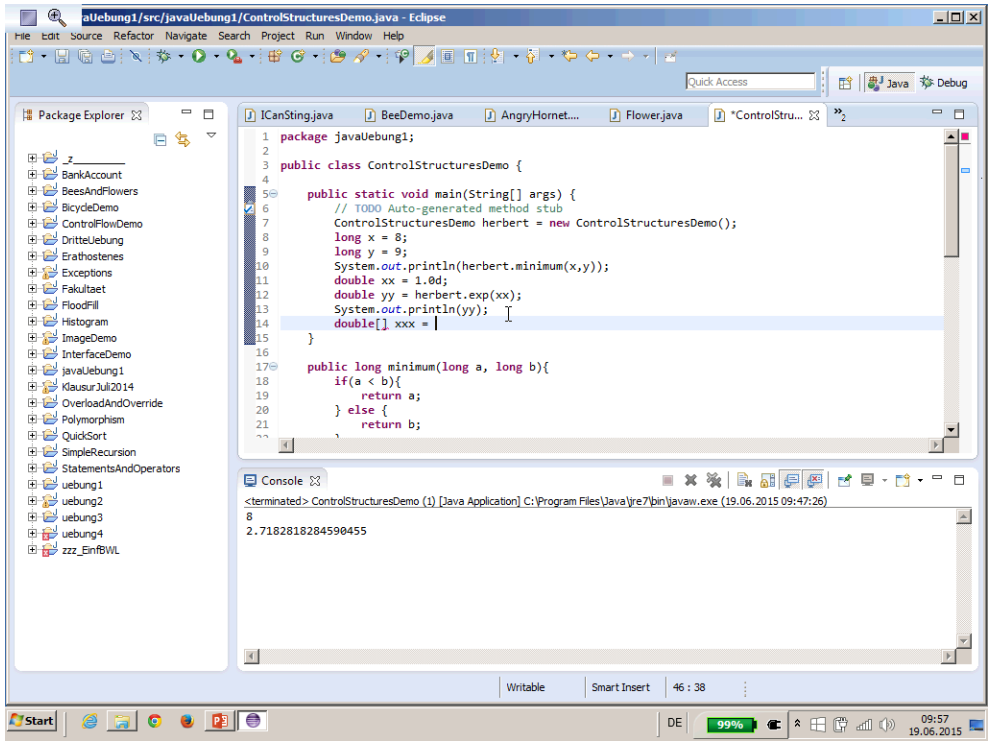

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
double y = 1.0d;
for(int i=1; i<31; i++){
y = x * y;
result = result + (y / fakultaet(i));
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
for(int i=0; i < a.length;)
return result;
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
45 : 26
Start
DE
99%
09:53
19.06.2015

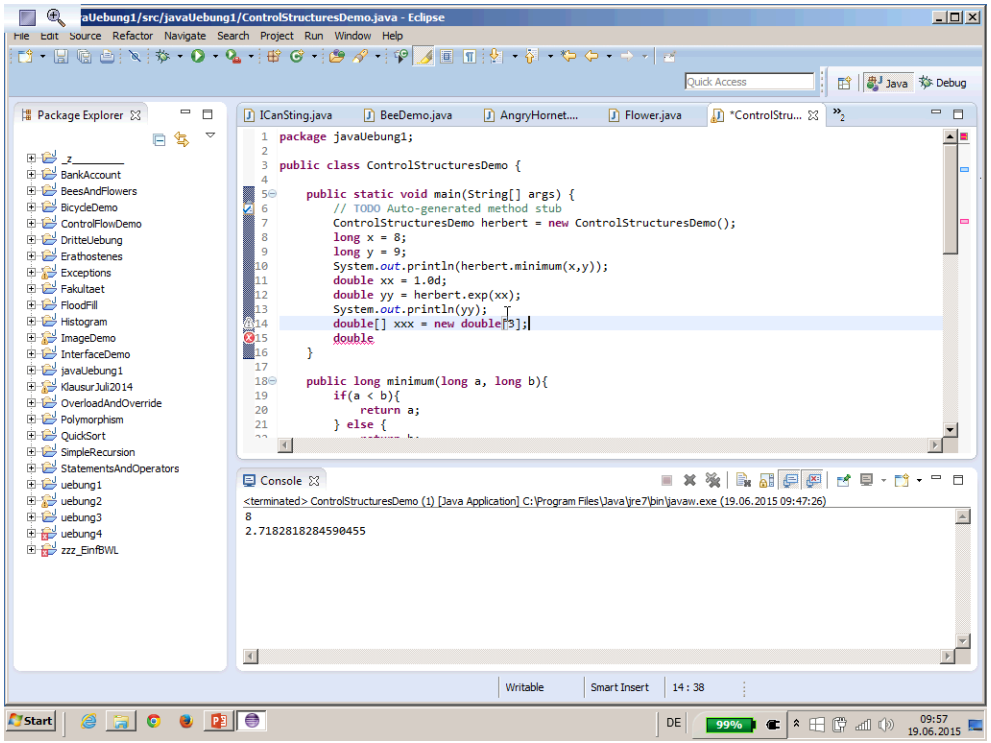

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
for(int i=0; i < a.length;){
re
}
return result;
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
44 : 1
Start
DE
99%
09:55
19.06.2015

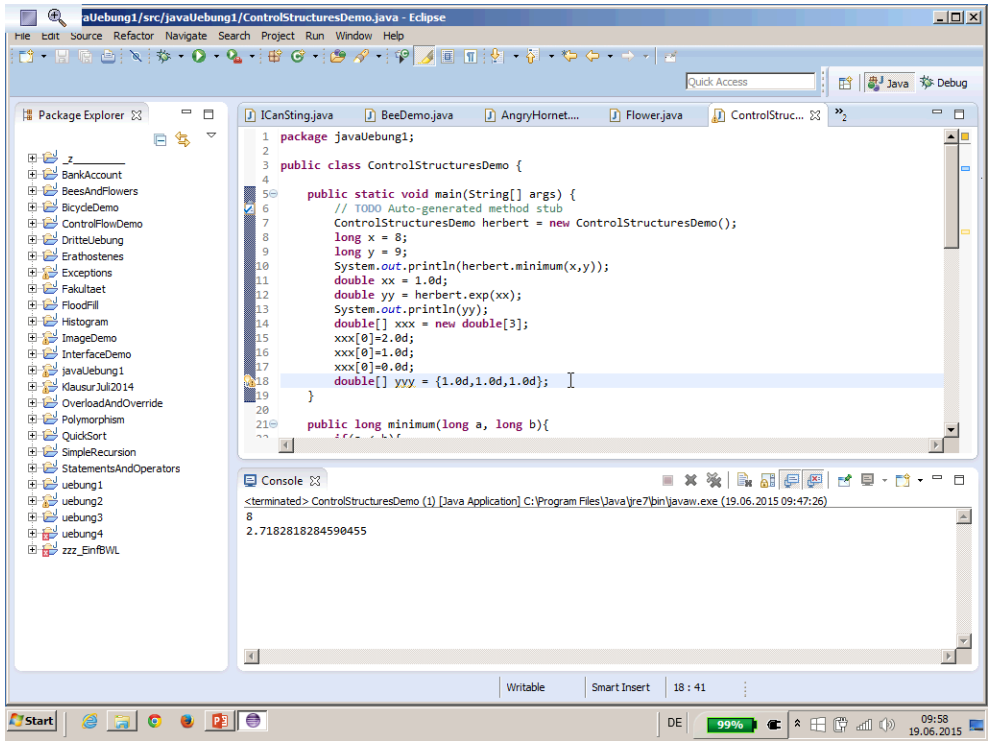

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
30 }
31 return result;
32 }
33
34 public long fakultaet(int n){
35 long result = 1;
36 while(n>1){
37 result = result * n;
38 n = n - 1;
39 }
40 return result;
41 }
42
43 public double dotProduct(double[] a, double[] b){
44 double result = 0.0;
45 for(int i=0; i < a.length;){
46 result = result +
47 }
48 return result;
49 }
50
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
46 : 18
Start
DE
99%
09:56
19.06.2015

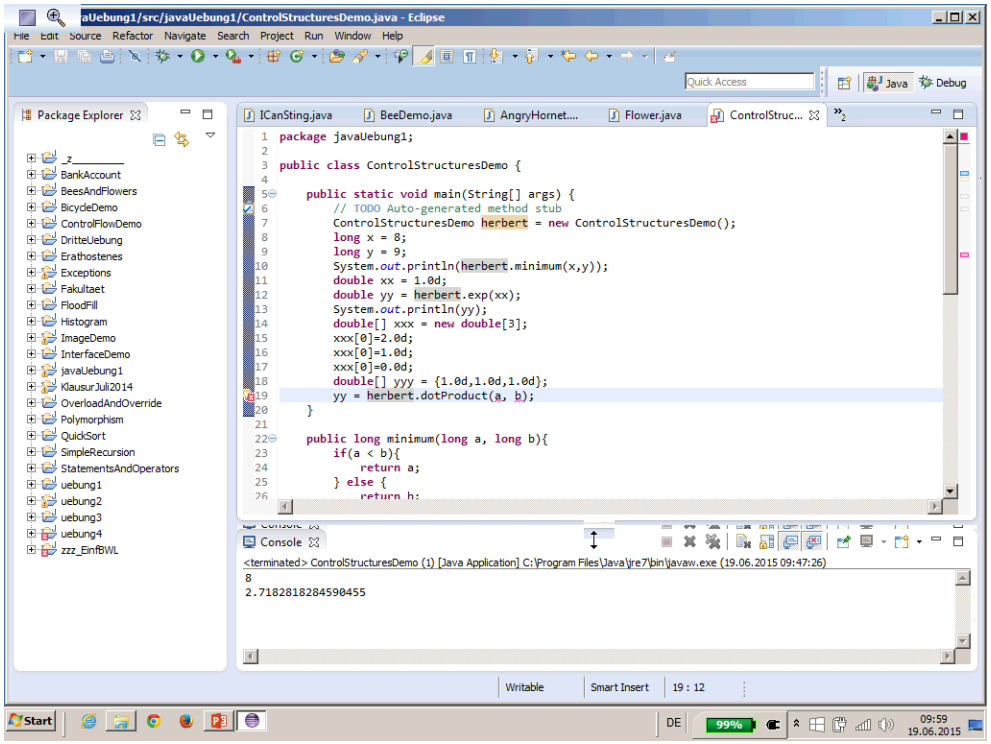

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
30 }
31 return result;
32 }
33
34 public long fakultaet(int n){
35 long result = 1;
36 while(n>1){
37 result = result * n;
38 n = n - 1;
39 }
40 return result;
41 }
42
43 public double dotProduct(double[] a, double[] b){
44 double result = 0.0;
45 for(int i=0; i < a.length;){
46 result = result + a[i] * b[i];
47 }
48 return result;
49 }
50
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
46 : 18
Start
DE
99%
09:56
19.06.2015

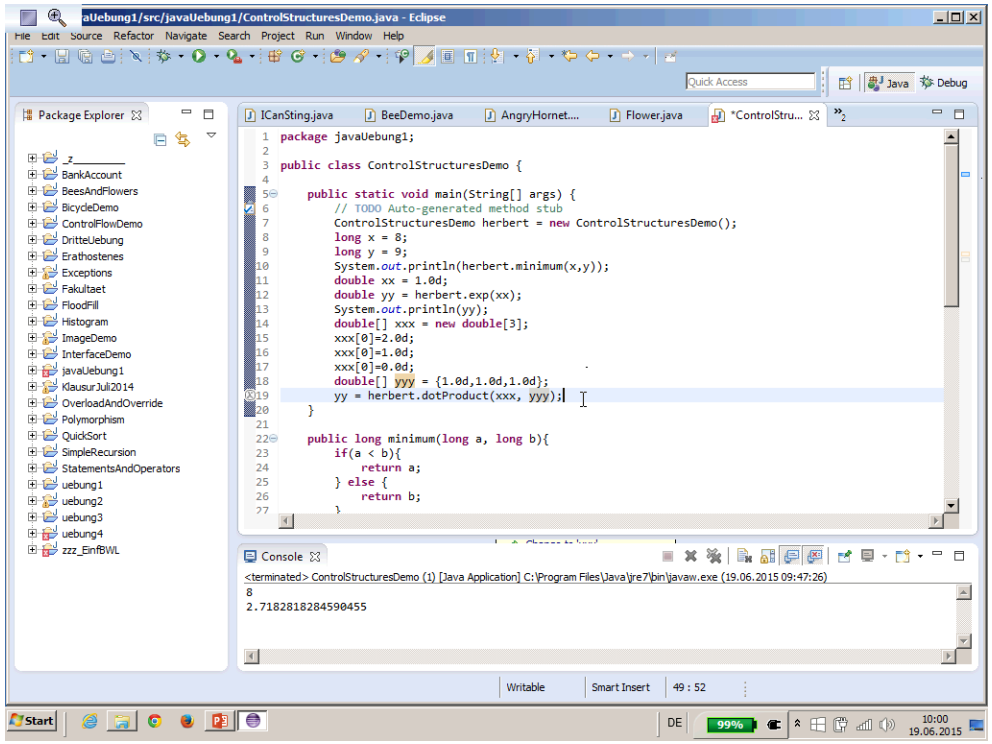

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo herbert = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(herbert.minimum(x,y));
11 double xx = 1.0d;
12 double yy = herbert.exp(xx);
13 System.out.println(yy);
14
15 }
16
17 public long minimum(long a, long b){
18 if(a < b){
19 return a;
20 } else {
21 return b;
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
46 : 38
Start
DE
99%
09:57
19.06.2015

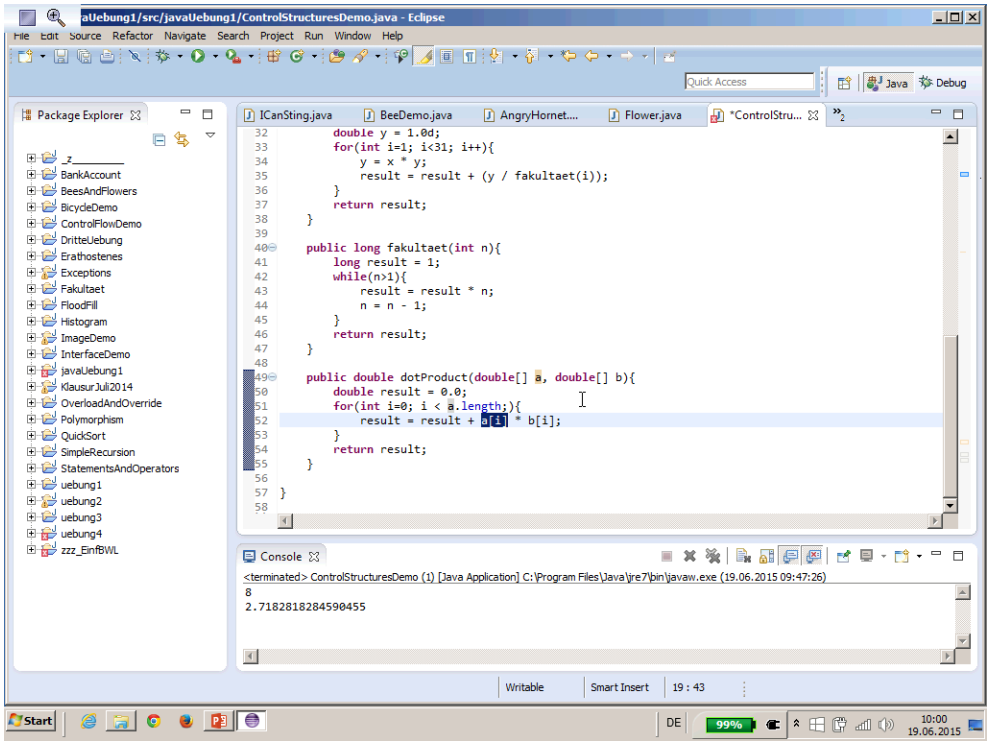

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo herbert = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(herbert.minimum(x,y));
11 double xx = 1.0d;
12 double yy = herbert.exp(xx);
13 System.out.println(yy);
14 double[] xxx =
15 }
16
17 public long minimum(long a, long b){
18 if(a < b){
19 return a;
20 } else {
21 return b;
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
46 : 38
Start
DE
99%
09:57
19.06.2015

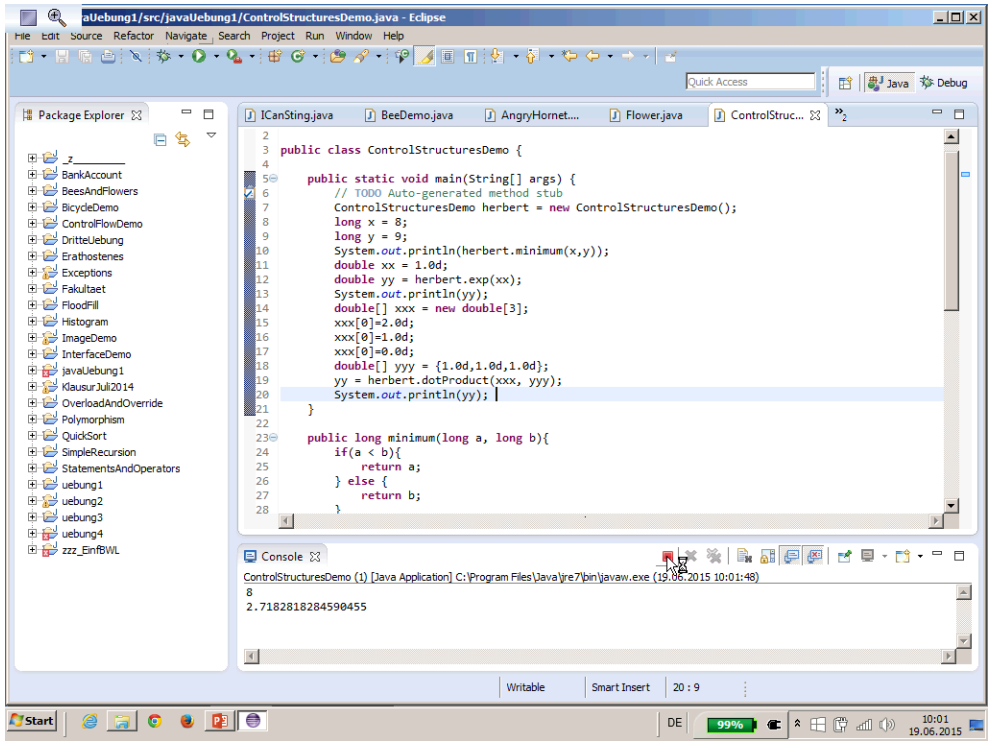

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo herbert = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(herbert.minimum(x,y));
11 double xx = 1.0d;
12 double yy = herbert.exp(xx);
13 System.out.println(yy);
14 double[] xxx = new double[3];
15 double
16 }
17
18 public long minimum(long a, long b){
19 if(a < b){
20 return a;
21 } else {
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
14 : 38
Start
DE
99%
09:57
19.06.2015

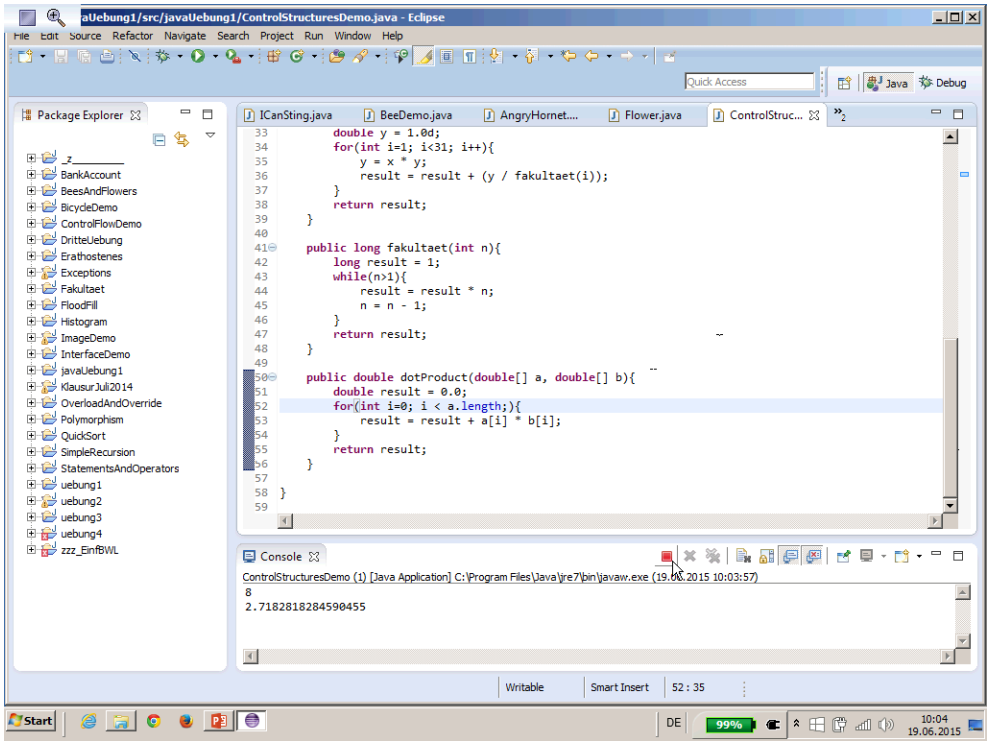

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo herbert = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(herbert.minimum(x,y));
11 double xx = 1.0d;
12 double yy = herbert.exp(xx);
13 System.out.println(yy);
14 double[] xxx = new double[3];
15 xxx[0]=2.0d;
16 xxx[0]=1.0d;
17 xxx[0]=0.0d;
18 double[] yyy = {1.0d,1.0d,1.0d};
19 }
20
21 public long minimum(long a, long b){
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
18 : 41
Start
DE
99%
09:58
19.06.2015

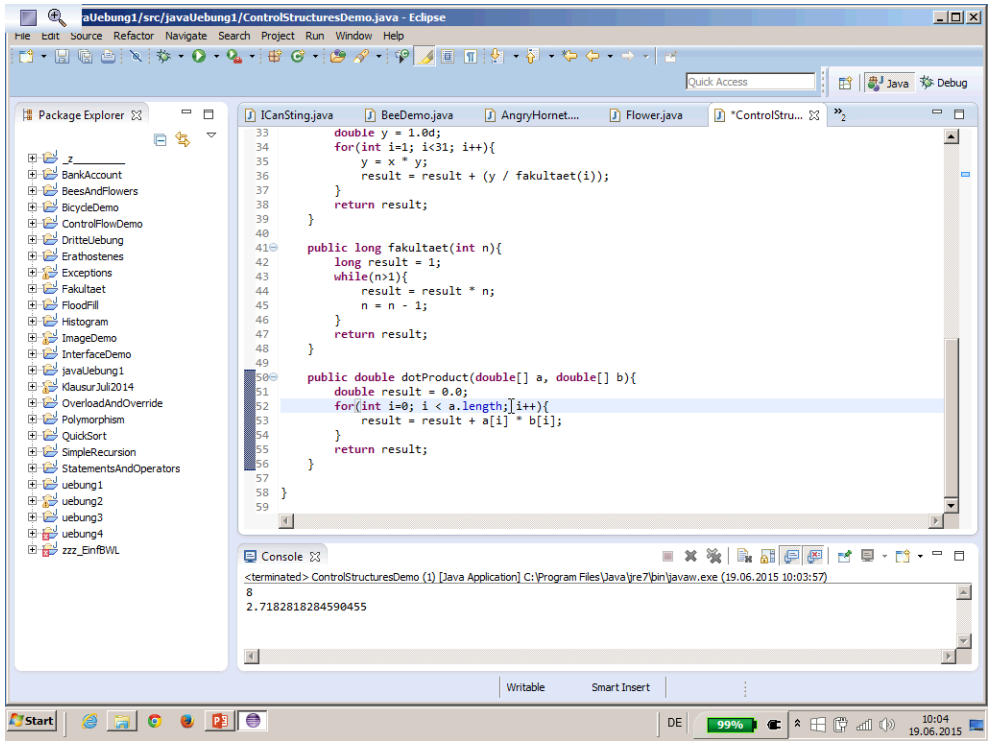

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo herbert = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(herbert.minimum(x,y));
11 double xx = 1.0d;
12 double yy = herbert.exp(xx);
13 System.out.println(yy);
14 double[] xxx = new double[3];
15 xxx[0]=2.0d;
16 xxx[0]=1.0d;
17 xxx[0]=0.0d;
18 double[] yyy = {1.0d,1.0d,1.0d};
19 yy = herbert.dotProduct(a, b);
20 }
21
22 public long minimum(long a, long b){
23 if(a < b){
24 return a;
25 } else {
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
19 : 12
Start
DE
99%
09:59
19.06.2015

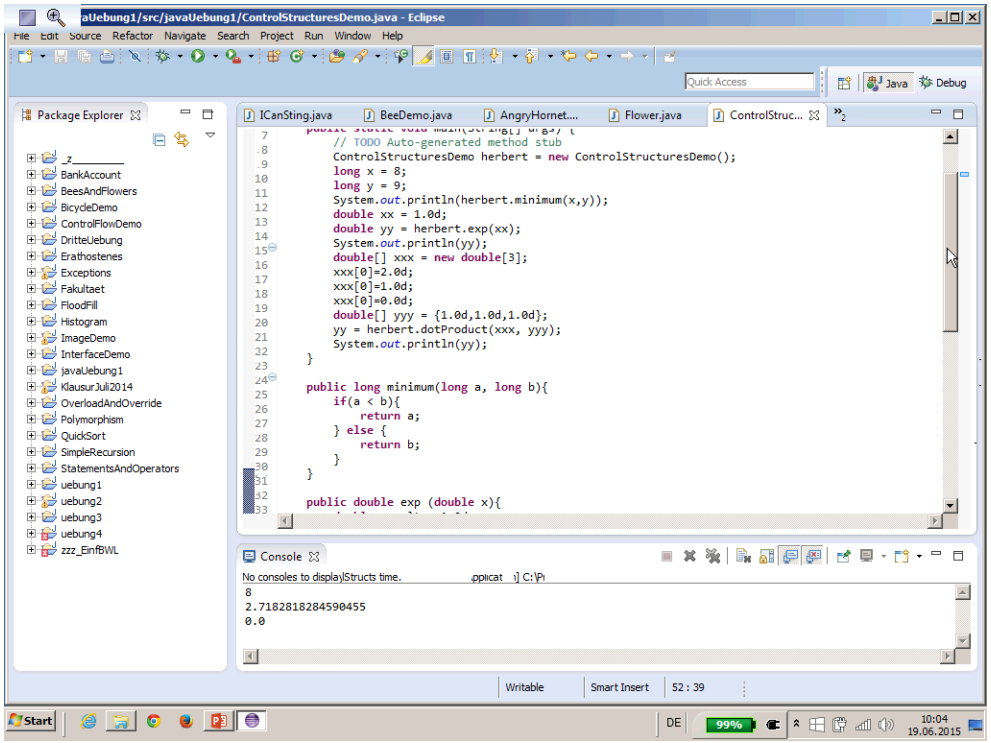

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
1 package javaUebung1;
2
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo herbert = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(herbert.minimum(x,y));
11 double xx = 1.0d;
12 double yy = herbert.exp(xx);
13 System.out.println(yy);
14 double[] xxx = new double[3];
15 xxx[0]=2.0d;
16 xxx[0]=1.0d;
17 xxx[0]=0.0d;
18 double[] yyy = {1.0d,1.0d,1.0d};
19 yy = herbert.dotProduct(xxx, yyy);
20 }
21
22 public long minimum(long a, long b){
23 if(a < b){
24 return a;
25 } else {
26 return b;
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
49 : 52
Start
DE
99%
10:00
19.06.2015

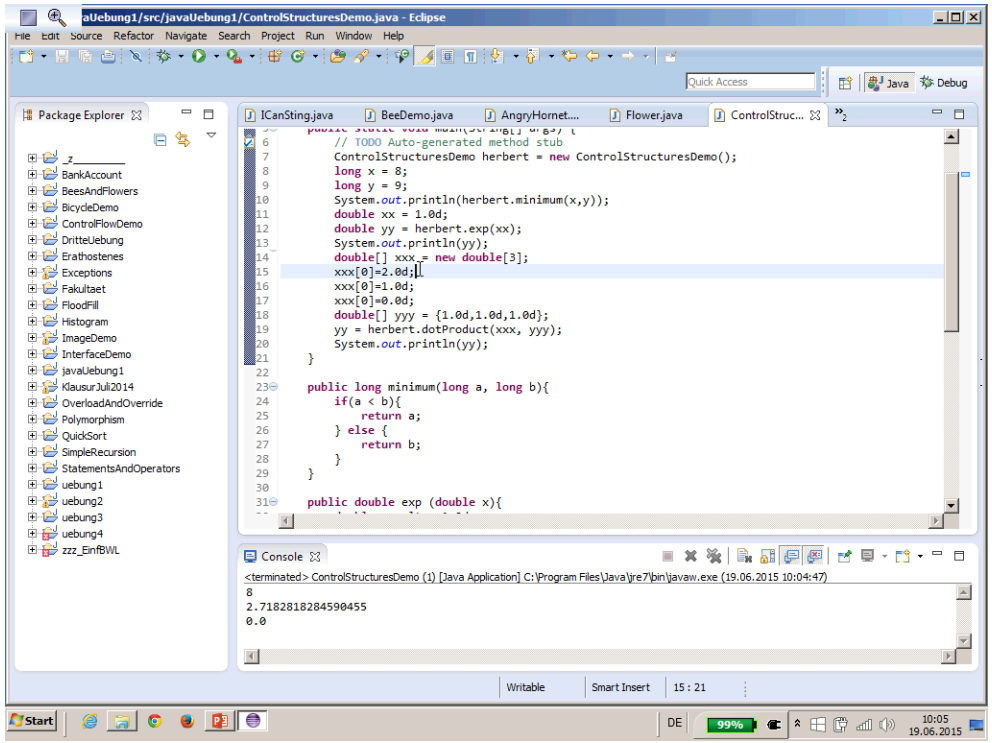

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
32 double y = 1.0d;
33 for(int i=1; i<31; i++){
34 y = x * y;
35 result = result + (y / fakultaet(i));
36 }
37 return result;
38 }
39
40 public long fakultaet(int n){
41 long result = 1;
42 while(n>1){
43 result = result * n;
44 n = n - 1;
45 }
46 return result;
47 }
48
49 public double dotProduct(double[] a, double[] b){
50 double result = 0.0;
51 for(int i=0; i < a.length;){
52 result = result + a[i] * b[i];
53 }
54 return result;
55 }
56
57 }
58
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 09:47:26)
8
2.7182818284590455
Writable
Smart Insert
19 : 43
Start
DE
99%
10:00
19.06.2015

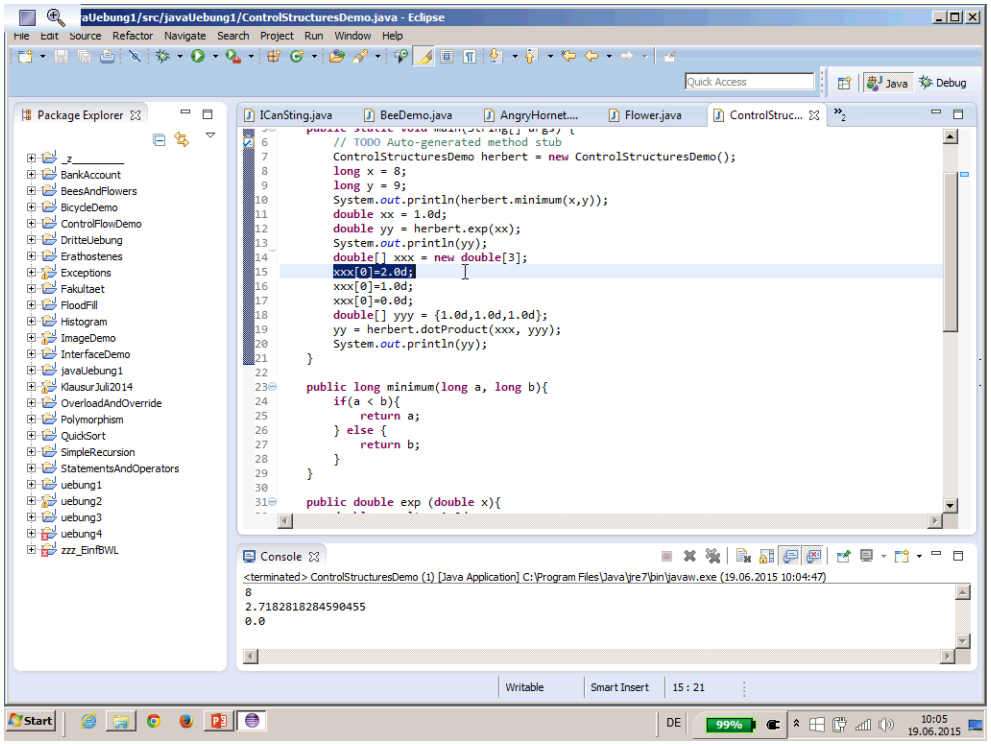

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
3 public class ControlStructuresDemo {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ControlStructuresDemo herbert = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(herbert.minimum(x,y));
11 double xx = 1.0d;
12 double yy = herbert.exp(xx);
13 System.out.println(yy);
14 double[] xxx = new double[3];
15 xxx[0]=2.0d;
16 xxx[0]=1.0d;
17 xxx[0]=0.0d;
18 double[] yyy = {1.0d,1.0d,1.0d};
19 yy = herbert.dotProduct(xxx, yyy);
20 System.out.println(yy);
21 }
22
23 public long minimum(long a, long b){
24 if(a < b){
25 return a;
26 } else {
27 return b;
28 }
Console
ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:01:48)
8
2.7182818284590455
Writable
Smart Insert
20 : 9
Start
DE
99%
10:01
19.06.2015

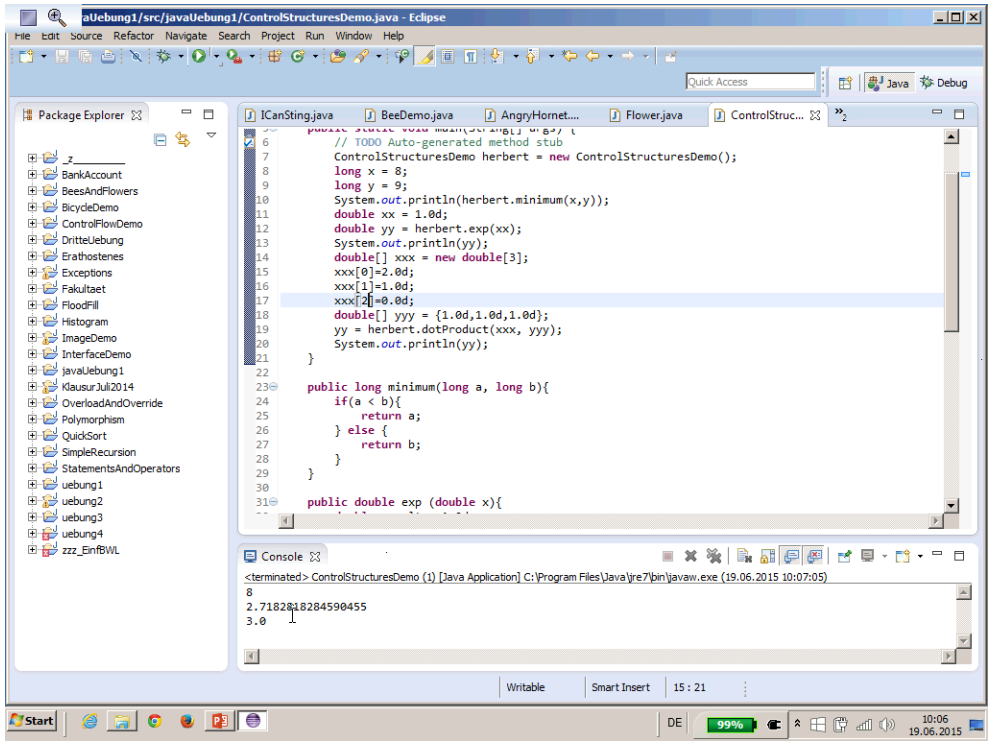

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
33 double y = 1.0d;
34 for(int i=1; i<31; i++){
35 y = x * y;
36 result = result + (y / fakultaet(i));
37 }
38 return result;
39 }
40
41 public long fakultaet(int n){
42 long result = 1;
43 while(n>1){
44 result = result * n;
45 n = n - 1;
46 }
47 return result;
48 }
49
50 public double dotProduct(double[] a, double[] b){
51 double result = 0.0;
52 for(int i=0; i < a.length;){
53 result = result + a[i] * b[i];
54 }
55 return result;
56 }
57
58 }
59
Console
ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:03:57)
8
2.7182818284590455
Writable
Smart Insert
52 : 35
Start
DE
99%
10:04
19.06.2015

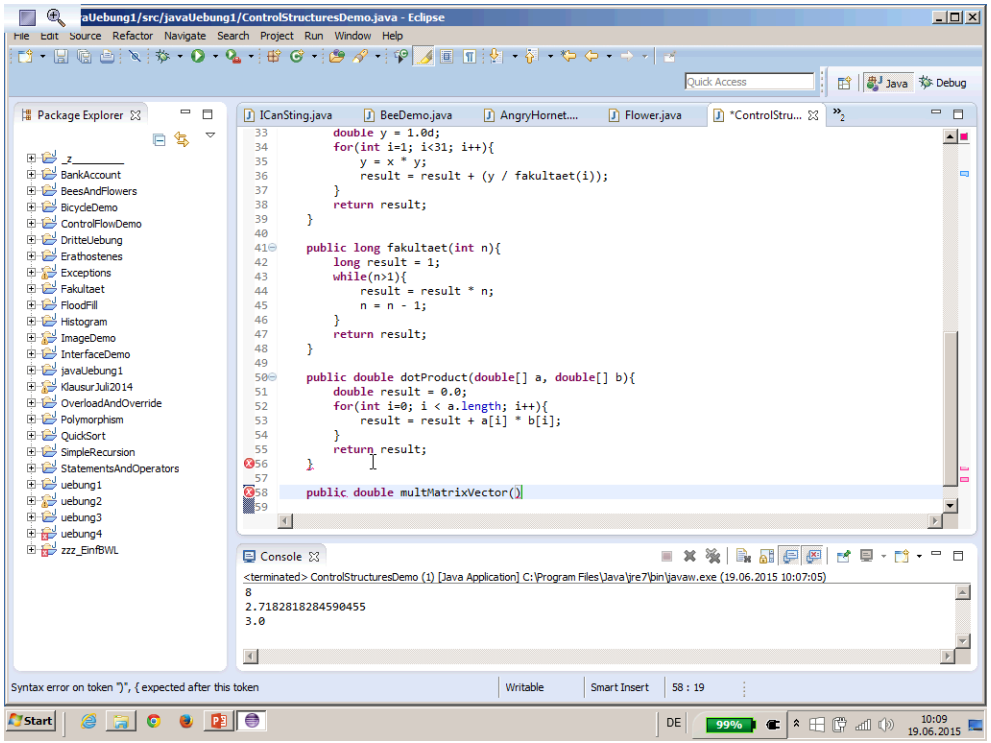

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
33 double y = 1.0d;
34 for(int i=1; i<31; i++){
35 y = x * y;
36 result = result + (y / fakultaet(i));
37 }
38 return result;
39 }
40
41 public long fakultaet(int n){
42 long result = 1;
43 while(n>1){
44 result = result * n;
45 n = n - 1;
46 }
47 return result;
48 }
49
50 public double dotProduct(double[] a, double[] b){
51 double result = 0.0;
52 for(int i=0; i < a.length;i++){
53 result = result + a[i] * b[i];
54 }
55 return result;
56 }
57
58 }
59
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:03:57)
8
2.7182818284590455
Writable
Smart Insert
Start
DE
99%
10:04
19.06.2015

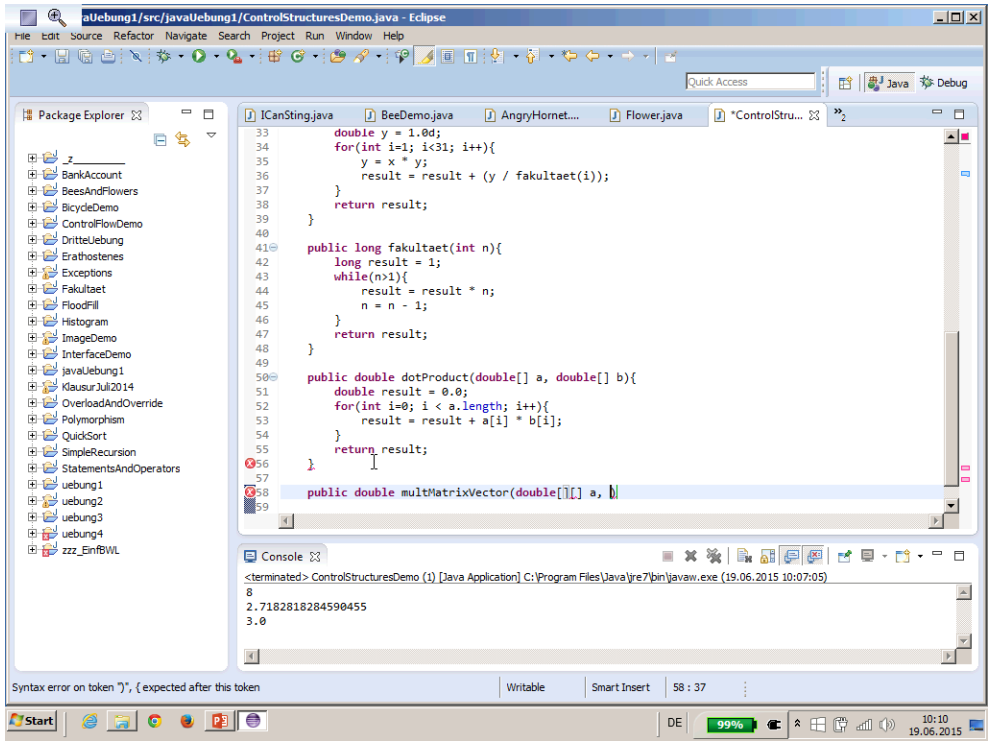

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
7 public static void main(String[] args) {
8 // TODO Auto-generated method stub
9 ControlStructuresDemo herbert = new ControlStructuresDemo();
10 long x = 8;
11 long y = 9;
12 System.out.println(herbert.minimum(x,y));
13 double xx = 1.0d;
14 double yy = herbert.exp(xx);
15 System.out.println(yy);
16 double[] xxx = new double[3];
17 xxx[0]=2.0d;
18 xxx[0]=1.0d;
19 xxx[0]=0.0d;
20 double[] yyy = {1.0d,1.0d,1.0d};
21 yy = herbert.dotProduct(xxx, yyy);
22 System.out.println(yy);
23 }
24
25 public long minimum(long a, long b){
26 if(a < b){
27 return a;
28 } else {
29 return b;
30 }
31 }
32
33 public double exp (double x){
Console
No consoles to display|Structs time. pplicat i] C:\P\
8
2.7182818284590455
0.0
Writable
Smart Insert
52 : 39
Start
DE
99%
10:04
19.06.2015

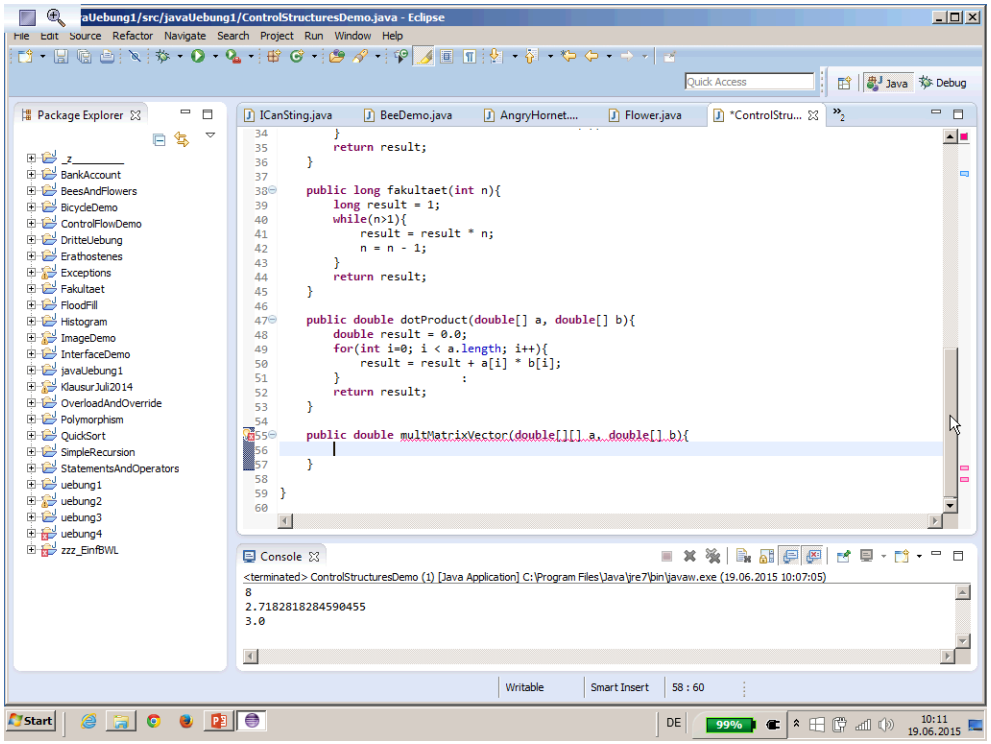

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
6 // TODO Auto-generated method stub
7 ControlStructuresDemo herbert = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(herbert.minimum(x,y));
11 double xx = 1.0d;
12 double yy = herbert.exp(xx);
13 System.out.println(yy);
14 double[] xxx = new double[3];
15 xxx[0]=2.0d;
16 xxx[0]=1.0d;
17 xxx[0]=0.0d;
18 double[] yyy = {1.0d,1.0d,1.0d};
19 yy = herbert.dotProduct(xxx, yyy);
20 System.out.println(yy);
21 }
22
23 public long minimum(long a, long b){
24 if(a < b){
25 return a;
26 } else {
27 return b;
28 }
29 }
30
31 public double exp (double x){
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:04:47)
8
2.7182818284590455
0.0
Writable
Smart Insert
15 : 21
Start
DE
99%
10:05
19.06.2015

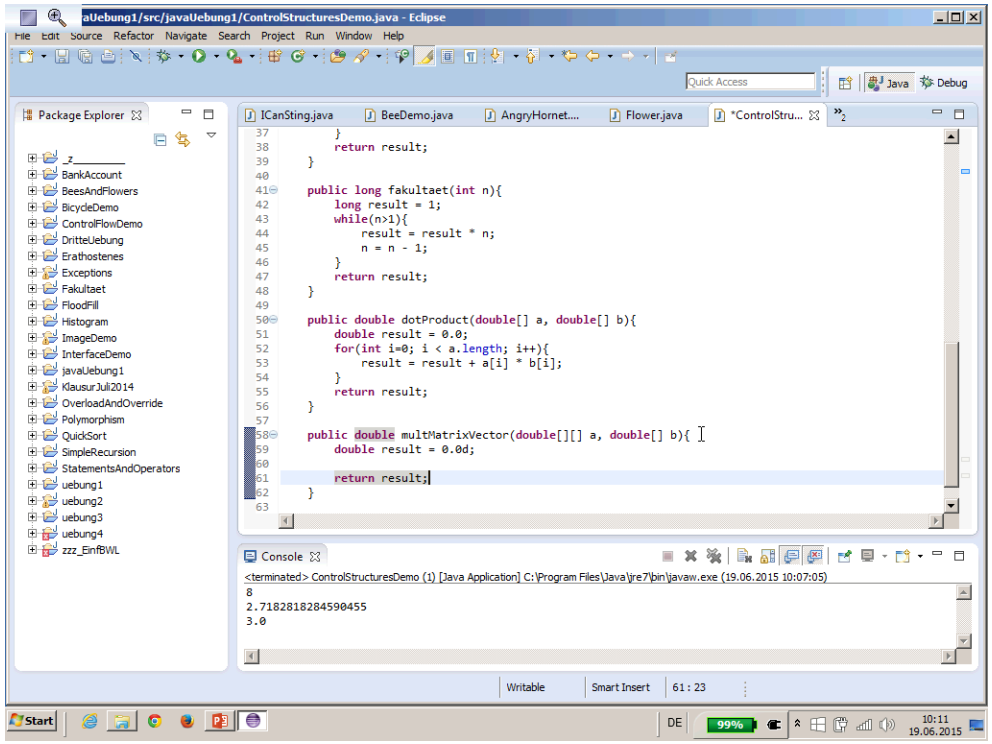

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
6 // TODO Auto-generated method stub
7 ControlStructuresDemo herbert = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(herbert.minimum(x,y));
11 double xx = 1.0d;
12 double yy = herbert.exp(xx);
13 System.out.println(yy);
14 double[] xxx = new double[3];
15 xxx[0]=2.0d;
16 xxx[0]=1.0d;
17 xxx[0]=0.0d;
18 double[] yyy = {1.0d,1.0d,1.0d};
19 yy = herbert.dotProduct(xxx, yyy);
20 System.out.println(yy);
21 }
22
23 public long minimum(long a, long b){
24 if(a < b){
25 return a;
26 } else {
27 return b;
28 }
29 }
30
31 public double exp (double x){
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:04:47)
8
2.7182818284590455
0.0
Writable
Smart Insert
15 : 21
Start
DE
99%
10:05
19.06.2015

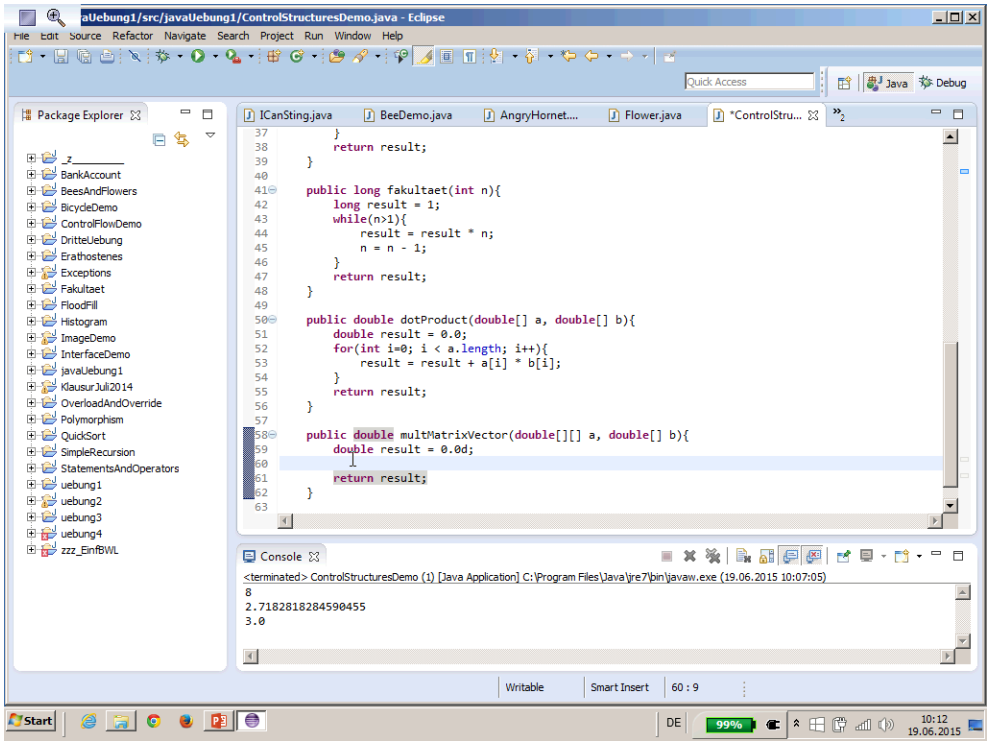

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
ControlStruc...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
6 // TODO Auto-generated method stub
7 ControlStructuresDemo herbert = new ControlStructuresDemo();
8 long x = 8;
9 long y = 9;
10 System.out.println(herbert.minimum(x,y));
11 double xx = 1.0d;
12 double yy = herbert.exp(xx);
13 System.out.println(yy);
14 double[] xxx = new double[3];
15 xxx[0]=2.0d;
16 xxx[1]=1.0d;
17 xxx[2]=0.0d;
18 double[] yyy = {1.0d,1.0d,1.0d};
19 yy = herbert.dotProduct(xxx, yyy);
20 System.out.println(yy);
21 }
22
23 public long minimum(long a, long b){
24 if(a < b){
25 return a;
26 } else {
27 return b;
28 }
29 }
30
31 public double exp (double x){
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
15 : 21
Start
DE
99%
10:06
19.06.2015

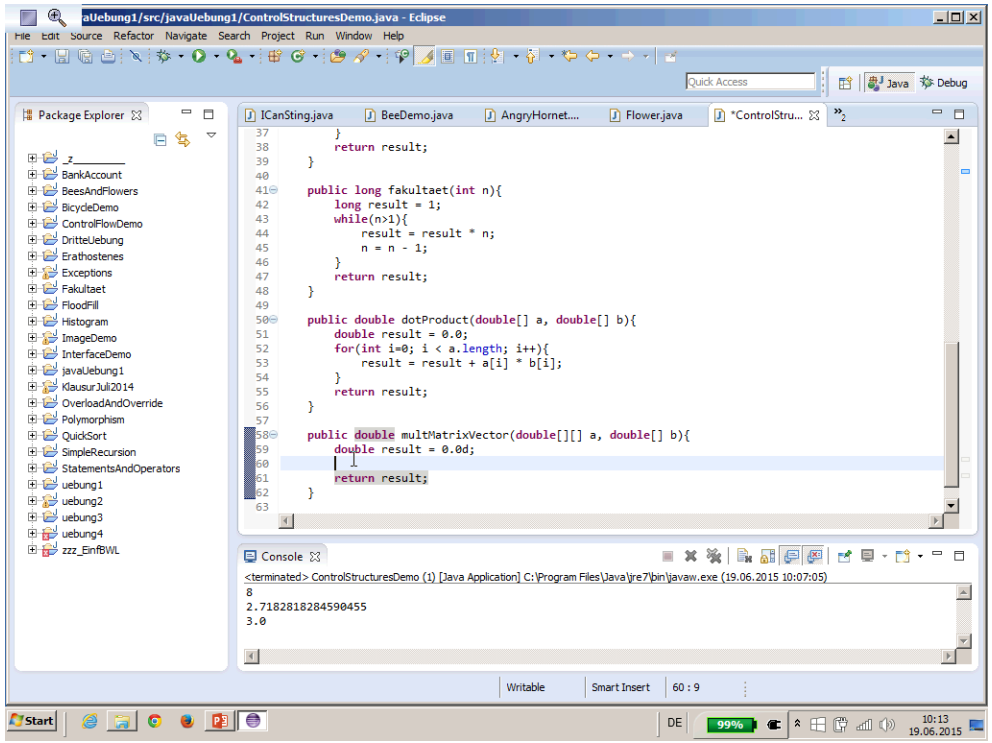

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
33 double y = 1.0d;
34 for(int i=1; i<31; i++){
35 y = x * y;
36 result = result + (y / fakultaet(i));
37 }
38 return result;
39 }
40
41 public long fakultaet(int n){
42 long result = 1;
43 while(n>1){
44 result = result * n;
45 n = n - 1;
46 }
47 return result;
48 }
49
50 public double dotProduct(double[] a, double[] b){
51 double result = 0.0;
52 for(int i=0; i < a.length; i++){
53 result = result + a[i] * b[i];
54 }
55 return result;
56 }
57
58 public double multMatrixVector()
59
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Syntax error on token ")", { expected after this token
Writable
Smart Insert
58 : 19
Start
DE
99%
10:09
19.06.2015

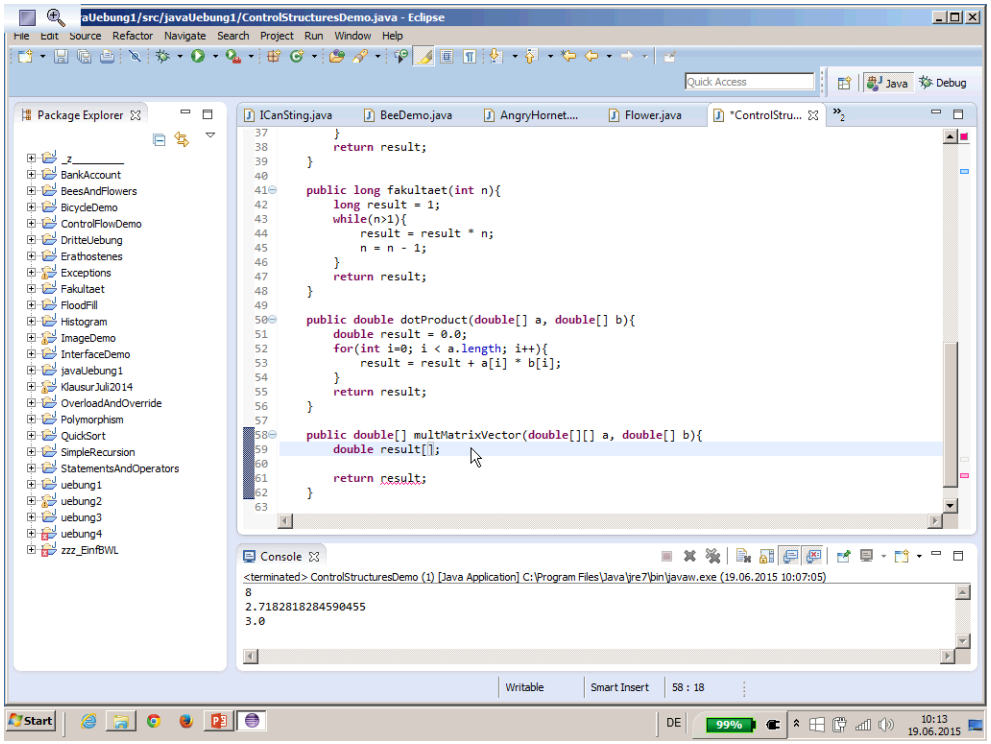

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
33 double y = 1.0d;
34 for(int i=1; i<31; i++){
35 y = x * y;
36 result = result + (y / fakultaet(i));
37 }
38 return result;
39 }
40
41 public long fakultaet(int n){
42 long result = 1;
43 while(n>1){
44 result = result * n;
45 n = n - 1;
46 }
47 return result;
48 }
49
50 public double dotProduct(double[] a, double[] b){
51 double result = 0.0;
52 for(int i=0; i < a.length; i++){
53 result = result + a[i] * b[i];
54 }
55 return result;
56 }
57
58 public double multMatrixVector(double[][] a, b
59
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Syntax error on token ")", { expected after this token
Writable
Smart Insert
58 : 37
Start
DE
99%
10:10
19.06.2015

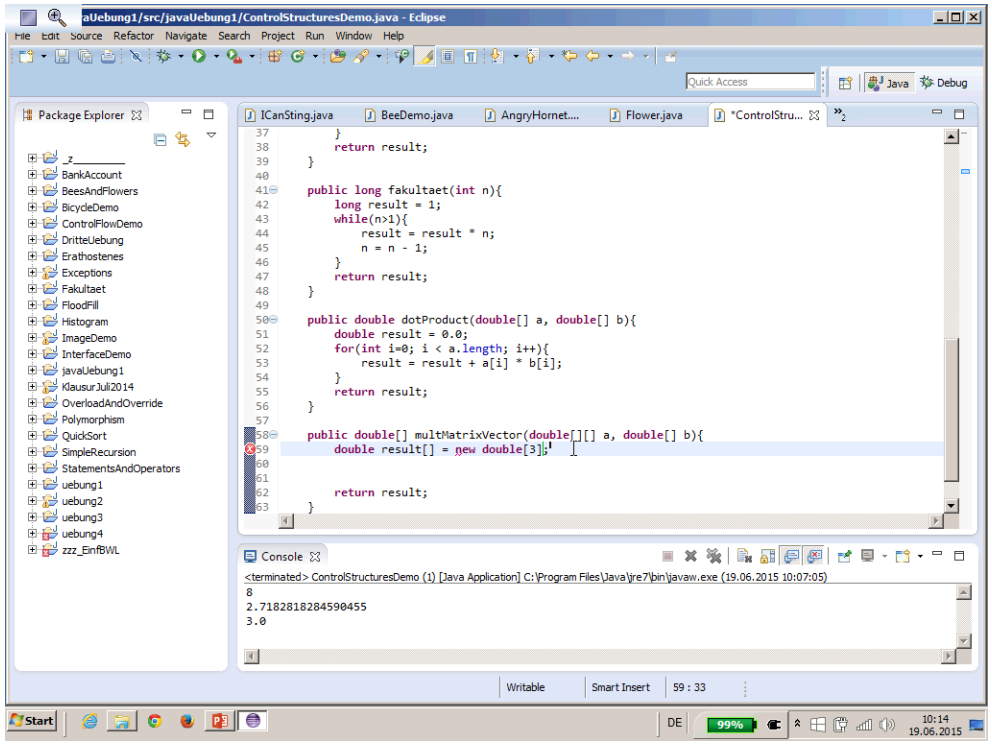

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
34 }
35 return result;
36 }
37
38 public long fakultaet(int n){
39 long result = 1;
40 while(n>1){
41 result = result * n;
42 n = n - 1;
43 }
44 return result;
45 }
46
47 public double dotProduct(double[] a, double[] b){
48 double result = 0.0;
49 for(int i=0; i < a.length; i++){
50 result = result + a[i] * b[i];
51 }
52 return result;
53 }
54
55 public double multMatrixVector(double[][] a, double[] b){
56
57 }
58
59 }
60
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
58 : 60
Start
DE
99%
10:11
19.06.2015

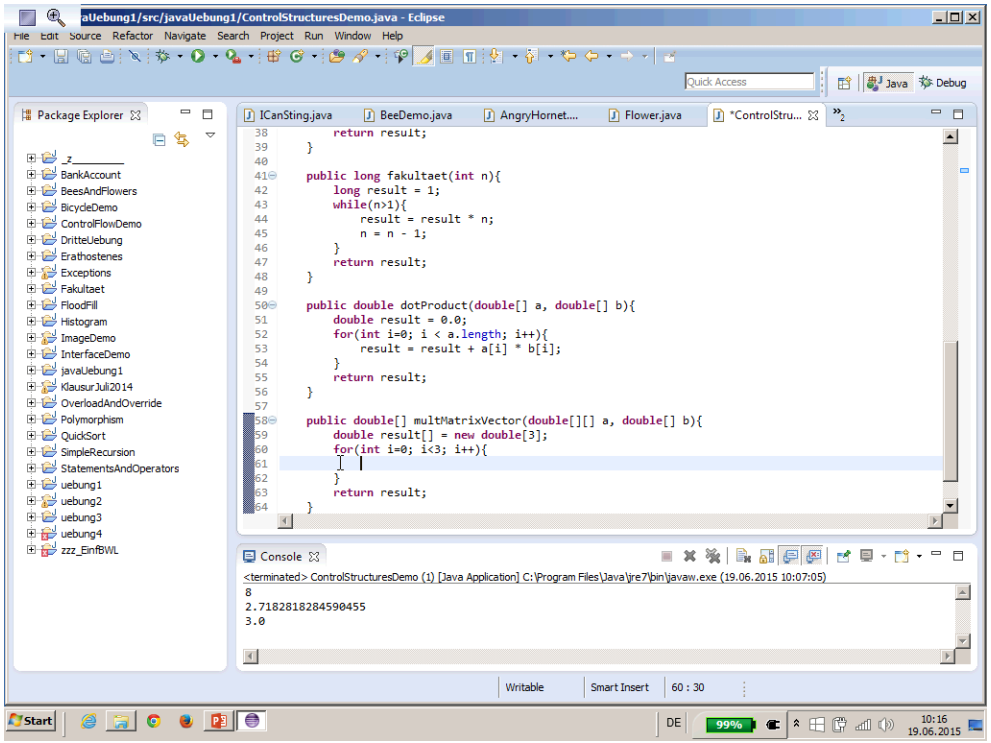

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
37 }
38 return result;
39 }
40
41 public long fakultaet(int n){
42 long result = 1;
43 while(n>1){
44 result = result * n;
45 n = n - 1;
46 }
47 return result;
48 }
49
50 public double dotProduct(double[] a, double[] b){
51 double result = 0.0;
52 for(int i=0; i < a.length; i++){
53 result = result + a[i] * b[i];
54 }
55 return result;
56 }
57
58 public double multMatrixVector(double[][] a, double[] b){
59 double result = 0.0d;
60
61 return result;
62 }
63
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
61 : 23
Start
DE
99%
10:11
19.06.2015

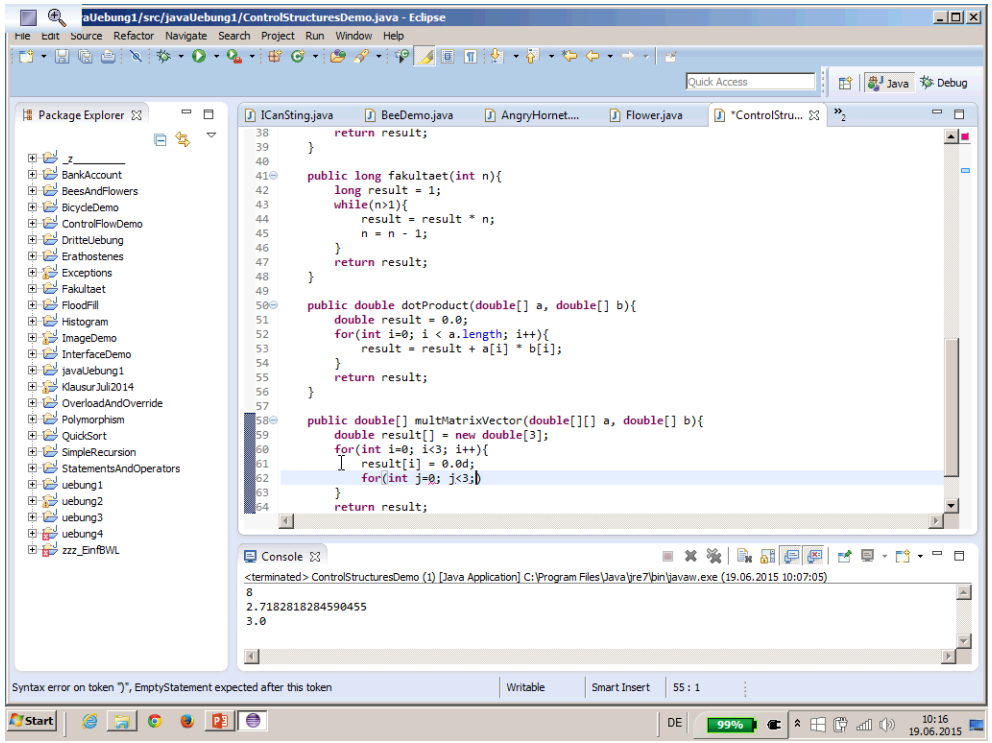

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
for(int i=0; i < a.length; i++){
result = result + a[i] * b[i];
}
return result;
}
public double multMatrixVector(double[][] a, double[] b){
double result = 0.0d;
return result;
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
60 : 9
Start
DE
99%
10:12
19.06.2015

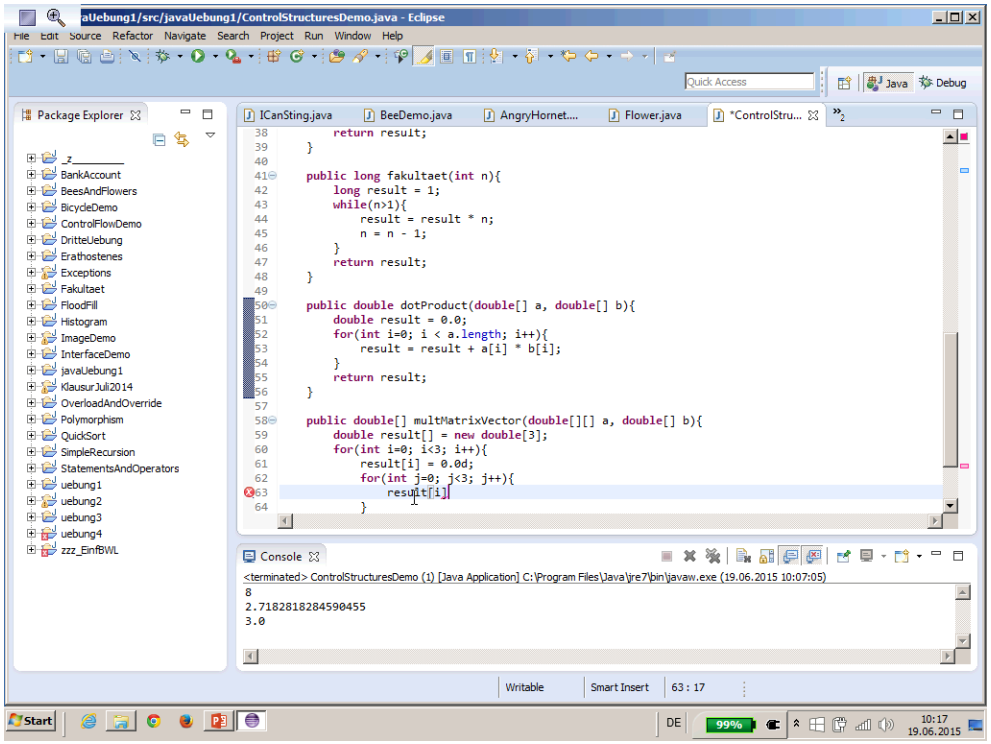

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
for(int i=0; i < a.length; i++){
result = result + a[i] * b[i];
}
return result;
}
public double multMatrixVector(double[][] a, double[] b){
double result = 0.0d;
return result;
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
60 : 9
Start
DE
99%
10:13
19.06.2015

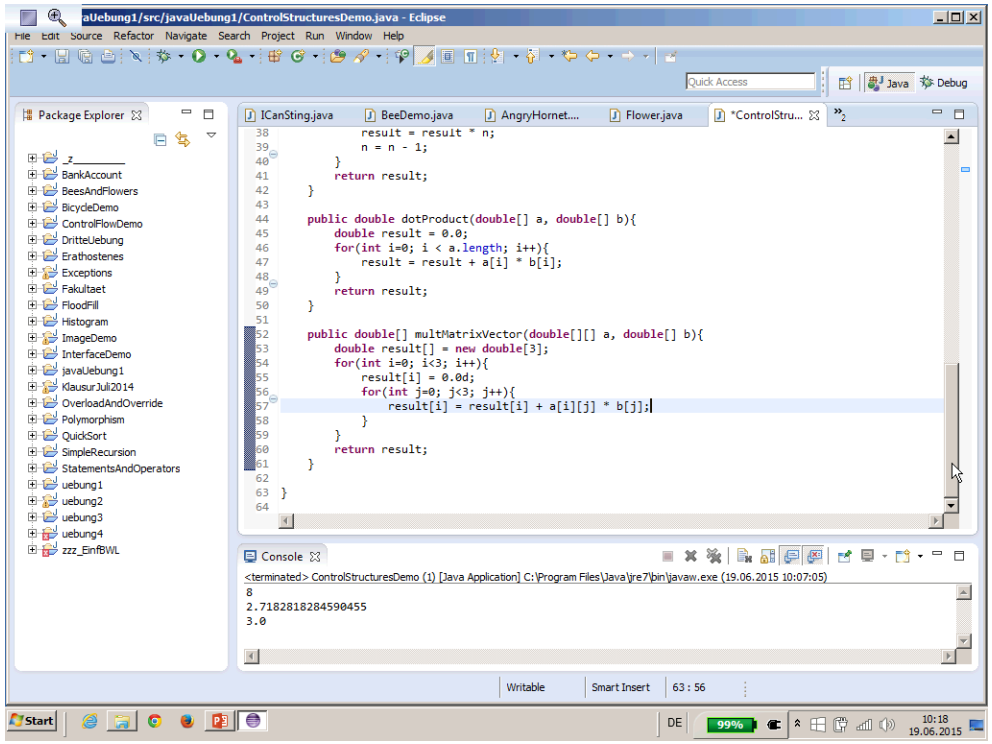

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
for(int i=0; i < a.length; i++){
result = result + a[i] * b[i];
}
return result;
}
public double[] multMatrixVector(double[][] a, double[] b){
double result[];
return result;
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
58 : 18
Start
DE
99%
10:13
19.06.2015

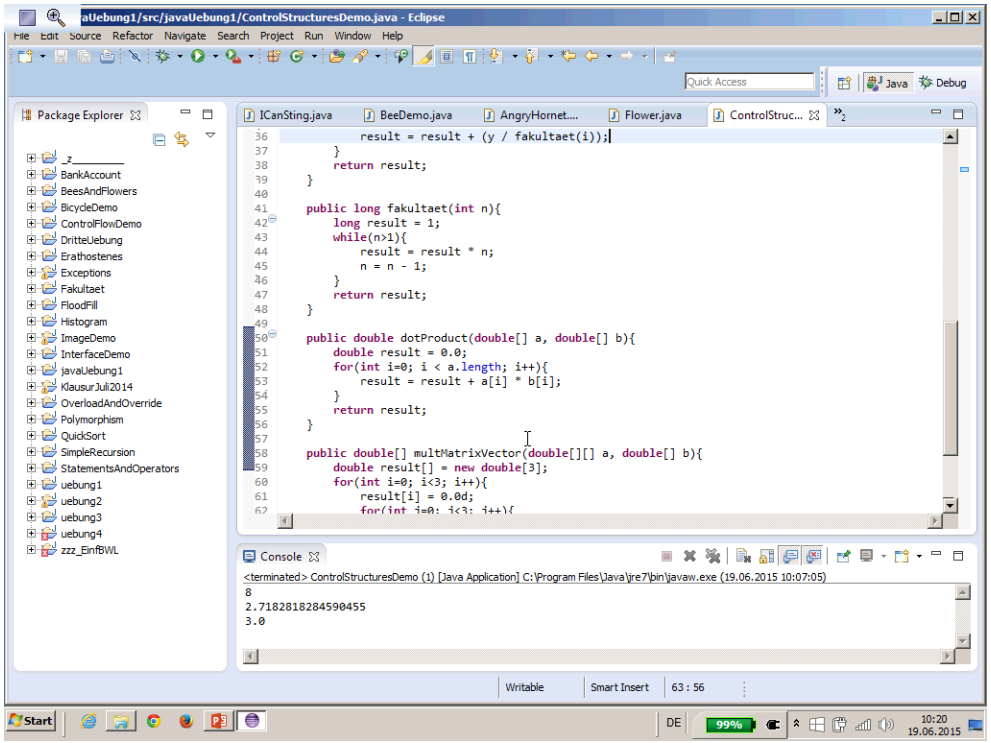

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
_z_____
BankAccount
BeesAndFlowers
BicycleDemo
ControlFlowDemo
DritteUebung
Erathostenes
Exceptions
Fakultaet
FloodFill
Histogram
ImageDemo
InterfaceDemo
javaUebung1
KlausurJuli2014
OverloadAndOverride
Polymorphism
QuickSort
SimpleRecursion
StatementsAndOperators
uebung1
uebung2
uebung3
uebung4
zzz_EinfBWL
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
for(int i=0; i < a.length; i++){
result = result + a[i] * b[i];
}
return result;
}
public double[] multMatrixVector(double[][] a, double[] b){
double result[] = new double[3];
return result;
}
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
59 : 33
Start
DE
99%
10:14
19.06.2015

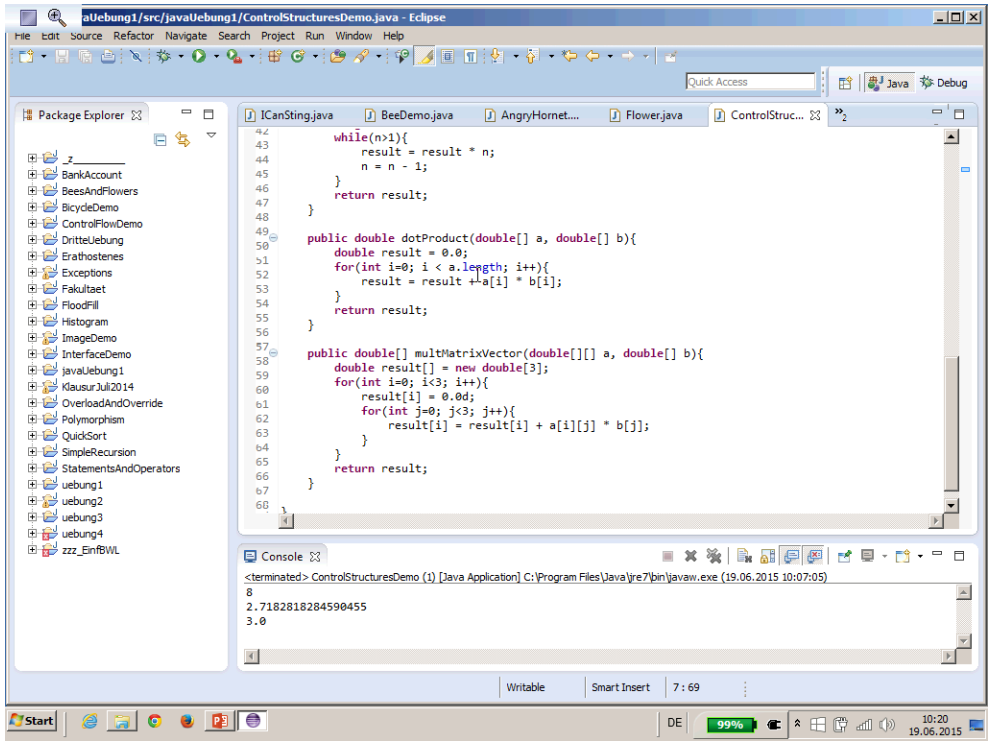

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
38 return result;
39 }
40
41 public long fakultaet(int n){
42 long result = 1;
43 while(n>1){
44 result = result * n;
45 n = n - 1;
46 }
47 return result;
48 }
49
50 public double dotProduct(double[] a, double[] b){
51 double result = 0.0;
52 for(int i=0; i < a.length; i++){
53 result = result + a[i] * b[i];
54 }
55 return result;
56 }
57
58 public double[] multMatrixVector(double[][] a, double[] b){
59 double result[] = new double[3];
60 for(int i=0; i<3; i++){
61
62 }
63 return result;
64 }
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
60 : 30
Start
DE
99%
10:16
19.06.2015

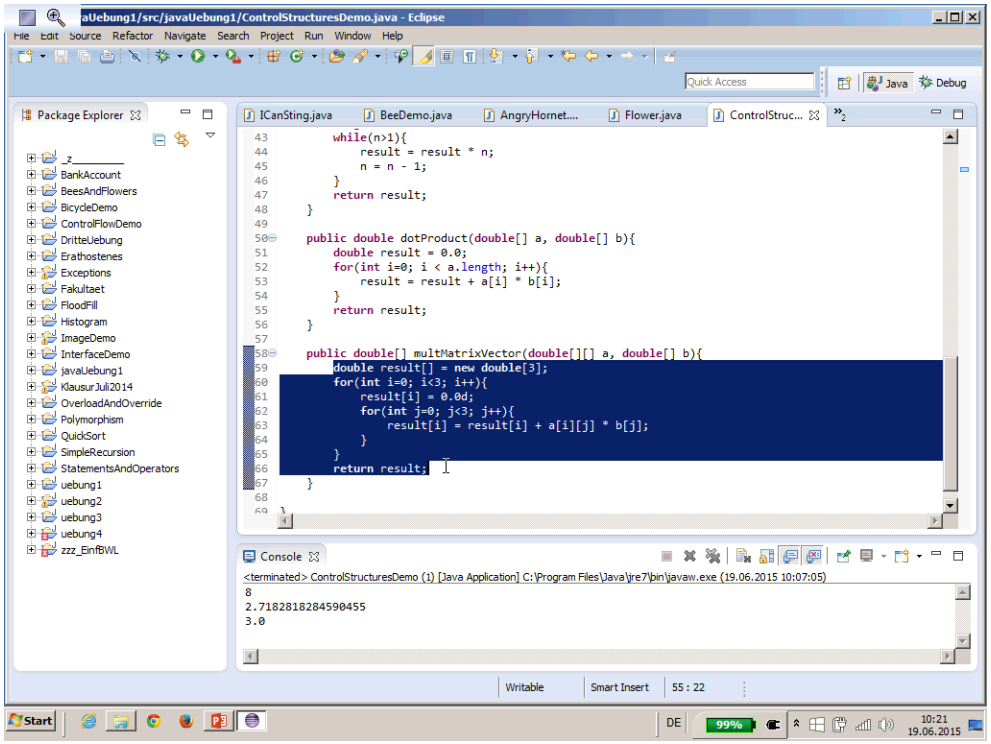

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
38 return result;
39 }
40
41 public long fakultaet(int n){
42 long result = 1;
43 while(n>1){
44 result = result * n;
45 n = n - 1;
46 }
47 return result;
48 }
49
50 public double dotProduct(double[] a, double[] b){
51 double result = 0.0;
52 for(int i=0; i < a.length; i++){
53 result = result + a[i] * b[i];
54 }
55 return result;
56 }
57
58 public double[] multMatrixVector(double[][] a, double[] b){
59 double result[] = new double[3];
60 for(int i=0; i<3; i++){
61 result[i] = 0.0d;
62 for(int j=0; j<3;)
63 }
64 return result;
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Syntax error on token ")", EmptyStatement expected after this token
Writable
Smart Insert
55 : 1
Start
DE
99%
10:16
19.06.2015

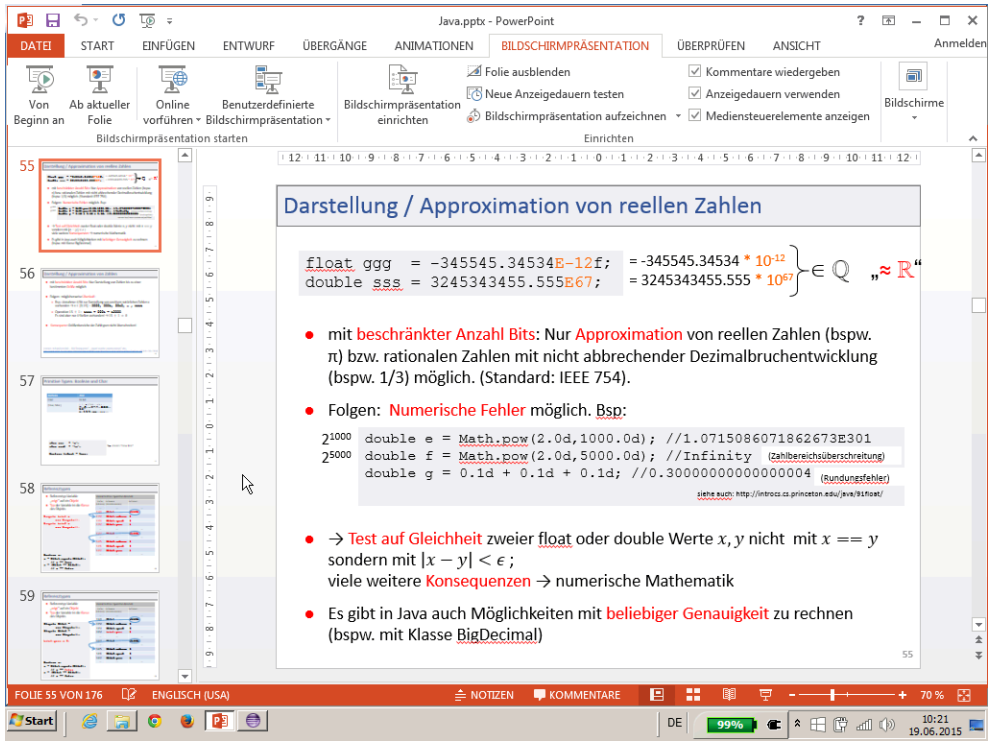

aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
38 return result;
39 }
40
41 public long fakultaet(int n){
42 long result = 1;
43 while(n>1){
44 result = result * n;
45 n = n - 1;
46 }
47 return result;
48 }
49
50 public double dotProduct(double[] a, double[] b){
51 double result = 0.0;
52 for(int i=0; i < a.length; i++){
53 result = result + a[i] * b[i];
54 }
55 return result;
56 }
57
58 public double[] multMatrixVector(double[][] a, double[] b){
59 double result[] = new double[3];
60 for(int i=0; i<3; i++){
61 result[i] = 0.0d;
62 for(int j=0; j<3; j++){
63 result[i]
64 }
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
63 : 17
Start
DE
99%
10:17
19.06.2015


aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
ICanSting.java
BeeDemo.java
AngryHornet....
Flower.java
*ControlStru...
38 result = result * n;
39 n = n - 1;
40 }
41 return result;
42 }
43
44 public double dotProduct(double[] a, double[] b){
45 double result = 0.0;
46 for(int i=0; i < a.length; i++){
47 result = result + a[i] * b[i];
48 }
49 return result;
50 }
51
52 public double[] multMatrixVector(double[][] a, double[] b){
53 double result[] = new double[3];
54 for(int i=0; i<3; i++){
55 result[i] = 0.0d;
56 for(int j=0; j<3; j++){
57 result[i] = result[i] + a[i][j] * b[j];
58 }
59 }
60 return result;
61 }
62
63 }
64
Console
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
63 : 56
Start
DE
99%
10:18
19.06.2015


aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
result = result + (y / fakultaet(i));
}
return result;
}
public long fakultaet(int n){
long result = 1;
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
for(int i=0; i < a.length; i++){
result = result + a[i] * b[i];
}
return result;
}
public double[] multMatrixVector(double[][] a, double[] b){
double result[] = new double[3];
for(int i=0; i<3; i++){
result[i] = 0.0d;
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
63 : 56


aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
for(int i=0; i < a.length; i++){
result = result + a[i] * b[i];
}
return result;
}
public double[] multMatrixVector(double[][] a, double[] b){
double result[] = new double[3];
for(int i=0; i<3; i++){
result[i] = 0.0d;
for(int j=0; j<3; j++){
result[i] = result[i] + a[i][j] * b[j];
}
}
return result;
}
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
7 : 69


aUebung1/src/javaUebung1/ControlStructuresDemo.java - Eclipse
while(n>1){
result = result * n;
n = n - 1;
}
return result;
}
public double dotProduct(double[] a, double[] b){
double result = 0.0;
for(int i=0; i < a.length; i++){
result = result + a[i] * b[i];
}
return result;
}
public double[] multMatrixVector(double[][] a, double[] b){
double result[] = new double[3];
for(int i=0; i<3; i++){
result[i] = 0.0d;
for(int j=0; j<3; j++){
result[i] = result[i] + a[i][j] * b[j];
}
return result;
<terminated> ControlStructuresDemo (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (19.06.2015 10:07:05)
8
2.7182818284590455
3.0
Writable
Smart Insert
55 : 22


Java.pptx - PowerPoint
Darstellung / Approximation von reellen Zahlen
float ggg = -345545.34534E-12f;
double sss = 3245343455.555E67;
= -345545.34534 * 10-12
= 3245343455.555 * 1067
∈ ℚ „≈ ℝ“
mit beschränkter Anzahl Bits: Nur Approximation von reellen Zahlen (bspw. π) bzw. rationalen Zahlen mit nicht abbrechender Dezimalbruchentwicklung (bspw. 1/3) möglich. (Standard: IEEE 754).
Folgen: Numerische Fehler möglich. Bsp:
21000
25000
double e = Math.pow(2.0d,1000.0d); //1.0715086071862673E301
double f = Math.pow(2.0d,5000.0d); //Infinity (Zahlbereichsüberschreitung)
double g = 0.1d + 0.1d + 0.1d; //0.3000000000000004 (Rundungsfehler)
siehe auch: http://introcs.cs.princeton.edu/java/91float/
→ Test auf Gleichheit zweier float oder double Werte x, y nicht mit x == y sondern mit |x − y| < ϵ ;
viele weitere Konsequenzen → numerische Mathematik
Es gibt in Java auch Möglichkeiten mit beliebiger Genauigkeit zu rechnen (bspw. mit Klasse BigDecimal)
FOLIE 55 VON 176
ENGLISCH (USA)
NOTIZEN
KOMMENTARE
70 %

Klassen-Definitionen – Generelle Form

```
class Bicycle {
    public int
    public int
    public int

    public Bi
        gear
        cade
        spee
    }

    public vo
        cade
    }

    public void chang
        gear = newVa
    }

    public void speed
        speed = spee
    }

    public void apply
        speed = spee
    }
}
```

Klassen-Definitionen – Generelle Form:

```
modifier class MyClass extends MySuperClass
                implements YourInterface1, ...,
                           YourInterfaceN
{
    // Attribute, Konstruktoren, Methoden
}
```

- wobei (Access) *modifier* : Bestimmte Kombinationen von { `public, final, abstract` } und {protected, private, static} (aber nur für innere Klassen)
- wobei Namen der Klasse, Oberklasse und der Interfaces frei wählbar

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

    public MountainBike(int startHeight, int startCadence,
                        int startSpeed, int startGear)
    {
        super(startCadence, startSpeed, startGear);
        seatHeight = startHeight;
    }

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


Klassen-Definitionen – Generelle Form

class Bicycle { ... }

Klassen-Definitionen – Generelle Form:
modifier class MyClass extends MySuperClass
implements YourInterface1, ..., YourInterfaceN
{ // Attribute, Konstruktoren, Methoden }

- wobei (Access) *modifier* : Bestimmte Kombinationen von { `public, final, abstract` } und {protected, private, static} (aber nur für innere Klassen)
- wobei Namen der Klasse, Oberklasse und der Interfaces frei wählbar

public class MountainBike extends Bicycle { ... }

Klassen-Definitionen – Generelle Form

class Bicycle { ... }

Klassen-Definitionen – Generelle Form:
modifier class MyClass extends MySuperClass
implements YourInterface1, ..., YourInterfaceN
{ // Attribute, Konstruktoren, Methoden }

- wobei (Access) *modifier* : Bestimmte Kombinationen von { `public, final, abstract` } und {protected, private, static} (aber nur für innere Klassen)
- wobei Namen der Klasse, Oberklasse und der Interfaces frei wählbar

public class MountainBike extends Bicycle { ... }

Attribut-Deklarationen – Generelle Form

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

    publi

    }

    publi

    }

    public void chang
        gear = newVa
    }

    public void speed
        speed = spee
    }

    public void apply
        speed = spee
    }
}
```

Attribut-Deklarationen – Generelle Form

```
modifier type fieldName;
```

- wobei (Access) *modifier* : Bestimmte Kombinationen von { `public, protected, private, static, final` }
- wobei *type*: Jeder primitive or Referenz-Typ

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

    public MountainBike(int startHeight, int startCadence,
                        int startSpeed, int startGear)
    {
        super(startCadence, startSpeed, startGear);
        seatHeight = startHeight;
    }

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

Methoden-Definitionen – Generelle Form

Methoden-Deklarationen – Generelle Form:

modifier typeOfReturnValue methodName (parameters)
throwsClauses {
statements
}

- wobei (Access) *modifier* : Bestimmte Kombinationen von { `public`, `protected`, `private`, `static`, `final` }
- wobei *typeOfReturnValue* : Jeder primitive or Referenz-Typ des Rückgabewerts oder `void`
- *parameters*: *(kommt gleich)*
- *throwsClauses*: *(kommt bei Exceptions)*

```
class Bicycle {
        public i
        public i
        public i

        public B
              gea
              cad
              spe
        }

        public v
              cad
        }

        public v
              gear = newVa
        }

        public void speed
              speed = spee
        }

        public void apply
              speed = spee
        }
}
```

```
    public MountainBike(int startHeight, int startCadence,
                        int startSpeed, int startGear)
    {
        super(startCadence, startSpeed, startGear);
        seatHeight = startHeight;
    }

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

101

Methoden-Definitionen – Generelle Form

Methoden-Deklarationen – Generelle Form:

modifier typeOfReturnValue methodName (parameters)
throwsClauses {
statements
}

- wobei (Access) *modifier* : Bestimmte Kombinationen von { `public`, `protected`, `private`, `static`, `final` }
- wobei *typeOfReturnValue* : Jeder primitive or Referenz-Typ des Rückgabewerts oder `void`
- *parameters*: *(kommt gleich)*
- *throwsClauses*: *(kommt bei Exceptions)*

Methoden-Definitionen – Generelle Form

Methoden-Deklarationen – Generelle Form:

modifier typeOfReturnValue methodName (parameters)
throwsClauses {
statements
}

- wobei (Access) *modifier* : Bestimmte Kombinationen von { `public`, `protected`, `private`, `static`, `final` }
- wobei *typeOfReturnValue* : Jeder primitive or Referenz-Typ des Rückgabewerts oder `void`
- *parameters*: *(kommt gleich)*
- *throwsClauses*: *(kommt bei Exceptions)*

Konstruktoren (Constructors)

- können vollständige und konsistente Initialisierung der Objekte sicherstellen
- (auch generell für Methoden): Mehrere Varianten anbieten --> verschiedene Grade an Details für verschiedene Nutzer der Klasse (<---> Abstraktion (API), Information Hiding)
- Unterklassen-Konstruktoren: Zugriff auf Oberklassen-Konstruktor mit **`super`** und Erweiterung nach Bedarf

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

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

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

```
class Tandem extends Bicycle {
  int numberOfDrivers;

  Tandem(int c, int s, int g, int n) {
    super(c, s, g);
    numberOfDrivers = n;
  }
}
```

103

Konstruktoren (Constructors)

- können vollständige und konsistente Initialisierung der Objekte sicherstellen
- (auch generell für Methoden): Mehrere Varianten anbieten --> verschiedene Grade an Details für verschiedene Nutzer der Klasse (<---> Abstraktion (API), Information Hiding)
- Unterklassen-Konstruktoren: Zugriff auf Oberklassen-Konstruktor mit **super** und Erweiterung nach Bedarf

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

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

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

```
class Tandem extends Bicycle {
  int numberOfDrivers;

  Tandem(int c, int s, int g, int n) {
    super(c, s, g);
    numberOfDrivers = n;
  }
}
```


Konstruktoren (Constructors)

- können vollständige und konsistente Initialisierung der Objekte sicherstellen
- (auch generell für Methoden): Mehrere Varianten anbieten --> verschiedene Grade an Details für verschiedene Nutzer der Klasse (<---> Abstraktion (API), Information Hiding)
- Unterklassen-Konstruktoren: Zugriff auf Oberklassen-Konstruktor mit **super** und Erweiterung nach Bedarf

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

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

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

```
class Tandem extends Bicycle {
  int numberOfDrivers;

  Tandem(int c, int s, int g, int n) {
    super(c, s, g);
    numberOfDrivers = n;
  }
}
```

103

Konstruktoren - Beispiel

```
class Person {                              // no constructor provided
  String firstName;
  String lastName;
  long   taxIdent;                          // Requirement: taxIdent must be unique!
}
```

```
// Manual initialization, easy to make a mistake (e.g. what about `taxIdent`?)
Person p1     = new Person();
p1.firstName  = "Max";
p1.lastName   = "Mustermann";
p1.taxIdent   = 12345;

Person p2     = new Person();
p2.firstName  = "Fabienne";
p2.lastName   = "Fabelhaft";
p2.taxIdent   = 12345;        // oops!
```


Konstruktoren - Beispiel

```
class Person {                              // no constructor provided
  String firstName;
  String lastName;
  long   taxIdent;                          // Requirement: taxIdent must be unique!
}
```

```
// Manual initialization, easy to make a mistake (e.g. what about `taxIdent`?)
Person p1     = new Person();
p1.firstName  = "Max";
p1.lastName   = "Mustermann";
p1.taxIdent   = 12345;

Person p2     = new Person();
p2.firstName  = "Fabienne";
p2.lastName   = "Fabelhaft";
p2.taxIdent   = 12345;        // oops!
```

104

Konstruktoren - Beispiel

```
class Person {
  String firstName;
  String lastName;
  long   taxIdent;                              // Requirement: taxIdent must be unique!

  Person(String fName, String lName, long tIdent) { // now: constructor provided
    firstName = fName;
    lastName = lName;

    // Use the given tax identifier `tIdent` only if we can make sure it is unique:
    if (isUniqueTaxIdentifier(tIdent)) {
      taxIdent = tIdent;
    } else {
      System.err.println("Not unique!");
    }
  }
}
```

```
// Complete and consistent.
Person p1 = new Person("Max", "Mustermann", 12345);
```


Konstruktoren - Beispiel

```
class Person {
  String firstName;
  String lastName;
  long   taxIdent;                              // Requirement: taxIdent must be unique!

  Person(String fName, String lName, long tIdent) { // now: constructor provided
    firstName = fName;
    lastName = lName;

    // Use the given tax identifier `tIdent` only if we can make sure it is unique:
    if (isUniqueTaxIdentifier(tIdent)) {
      taxIdent = tIdent;
    } else {
      System.err.println("Not unique!");
    }
  }
}
```

```
// Complete and consistent.
Person p1 = new Person("Max", "Mustermann", 12345);
```

105

Konstruktoren - Beispiel

```
class Person {
  String firstName;
  String lastName;
  long   taxIdent;                              // Requirement: taxIdent must be unique!

  Person(String fName, String lName, long tIdent) { // now: constructor provided
    firstName = fName;
    lastName = lName;

    // Use the given tax identifier `tIdent` only if we can make sure it is unique:
    if (isUniqueTaxIdentifier(tIdent)) {
      taxIdent = tIdent;
    } else {
      System.err.println("Not unique!");
    }
  }
}
```

```
// Complete and consistent.
Person p1 = new Person("Max", "Mustermann", 12345);
```

105

Methodenaufruf und Parameter-Übergabe

Übergabe einer Liste von Parametern an Methoden / Konstruktoren

```
int doSomething(int primitiveParameter1,
                SomeClass referenceParameter)
{
       int someInt = 17 + 9;
       primitiveParameter1 = 0;
       referenceParameter = null;
       return someInt;
}
```