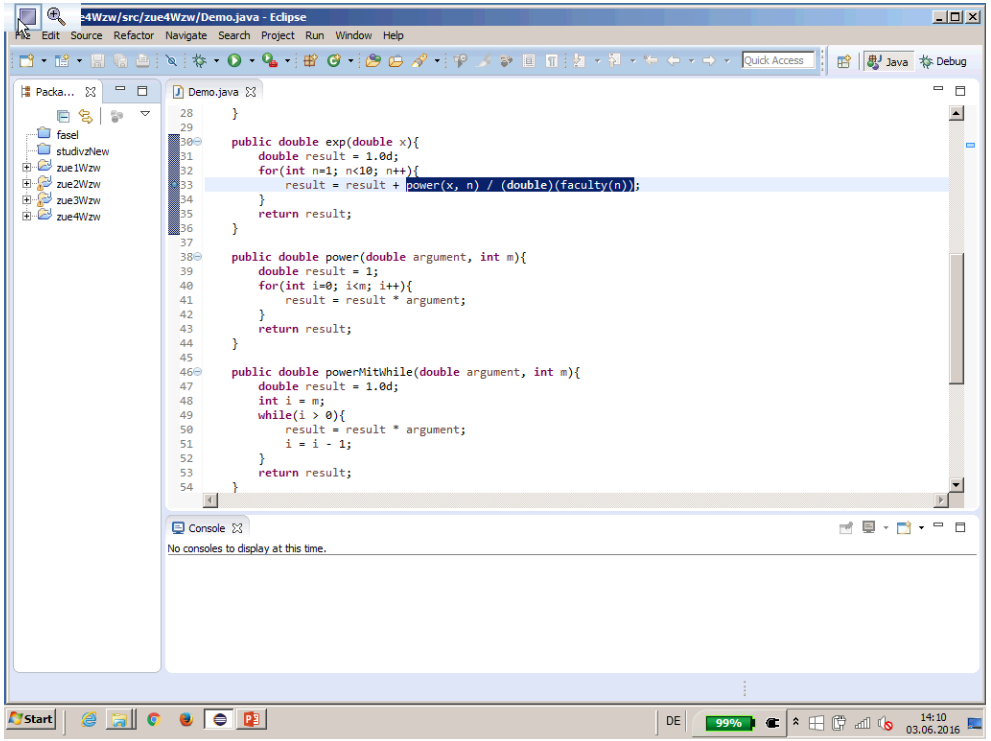
Script generated by TTT

Title: groh: profile1 (03.06.2016)

Date: Fri Jun 03 14:10:36 CEST 2016

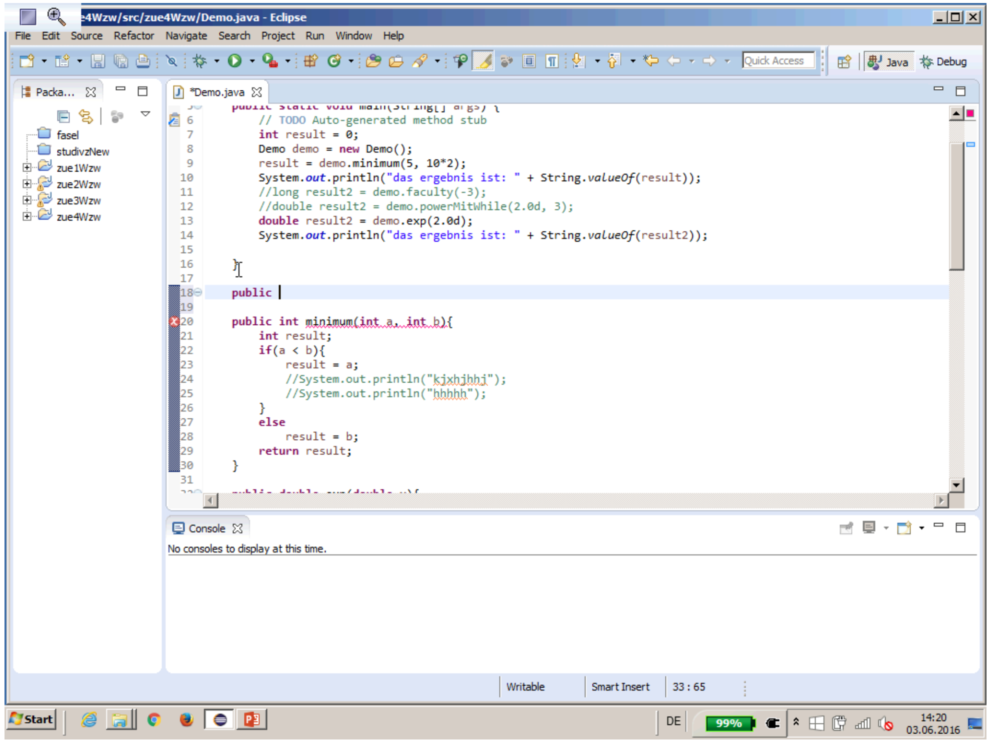
Duration: 88:27 min

Pages: 42



```java
    }

    public double exp(double x){
        double result = 1.0d;
        for(int n=1; n<10; n++){
            result = result + power(x, n) / (double)(faculty(n));
        }
        return result;
    }

    public double power(double argument, int m){
        double result = 1;
        for(int i=0; i<m; i++){
            result = result * argument;
        }
        return result;
    }

    public double powerMitWhile(double argument, int m){
        double result = 1.0d;
        int i = m;
        while(i > 0){
            result = result * argument;
            i = i - 1;
        }
        return result;
    }
```

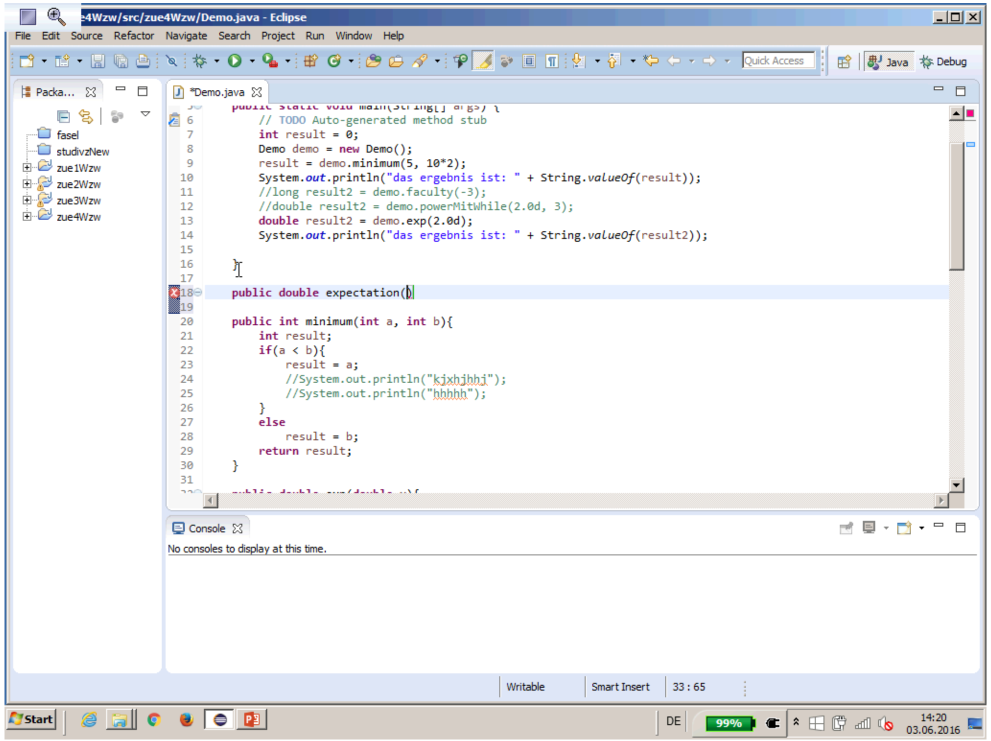
No consoles to display at this time.



```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));

    }

    public 

    public int minimum(int a, int b){
        int result;
        if(a < b){
            result = a;
            //System.out.println("kjxhjhhj");
            //System.out.println("hhhhh");
        }
        else
            result = b;
        return result;
    }

    public double exp(double x){
```

No consoles to display at this time.

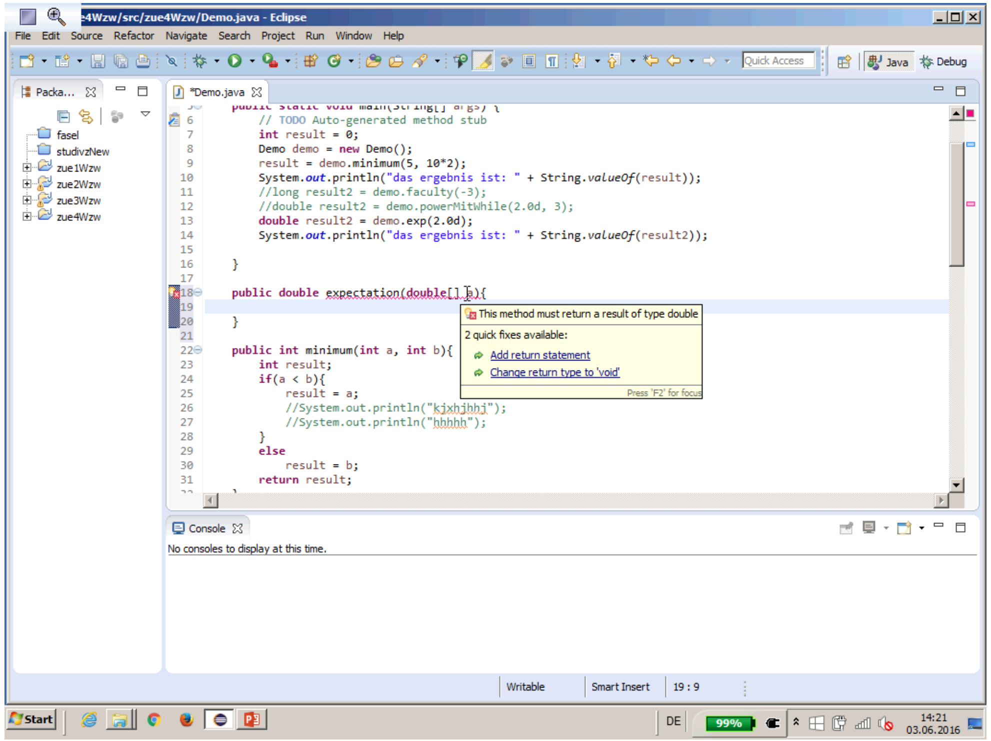
Writable   Smart Insert   33 : 65



```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));

    }

    public double expectation()

    public int minimum(int a, int b){
        int result;
        if(a < b){
            result = a;
            //System.out.println("kjxhjhhj");
            //System.out.println("hhhhh");
        }
        else
            result = b;
        return result;
    }

    public double exp(double x){
```

No consoles to display at this time.

Writable   Smart Insert   33 : 65



```
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));

    }

    public double expectation(double[] a){

    }

    public int minimum(int a, int b){
        int result;
        if(a < b){
            result = a;
            //System.out.println("kjxhjhhj");
            //System.out.println("hhhhh");
        }
        else
            result = b;
        return result;
```

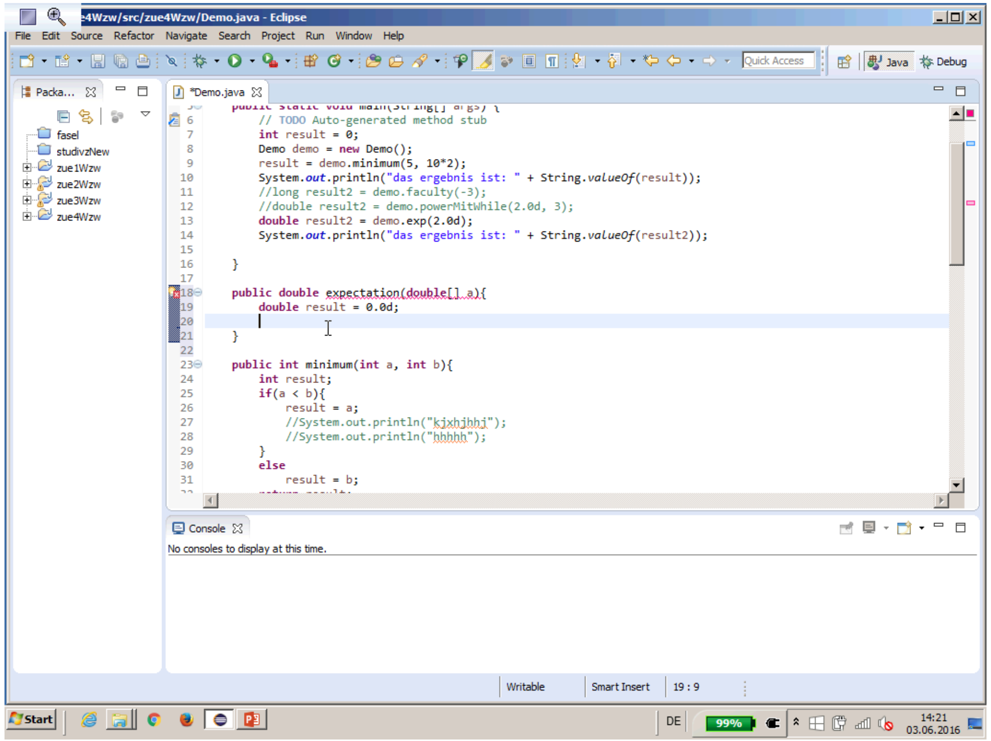
This method must return a result of type double

2 quick fixes available:

- Add return statement
- Change return type to 'void'

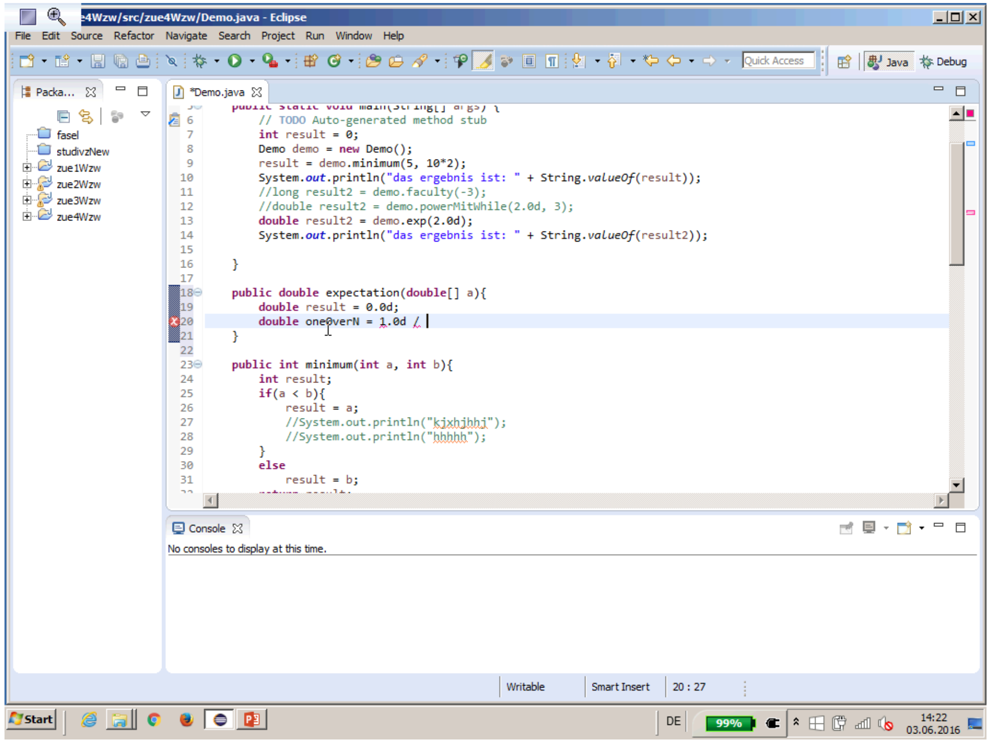
Press 'F2' for focus

No consoles to display at this time.



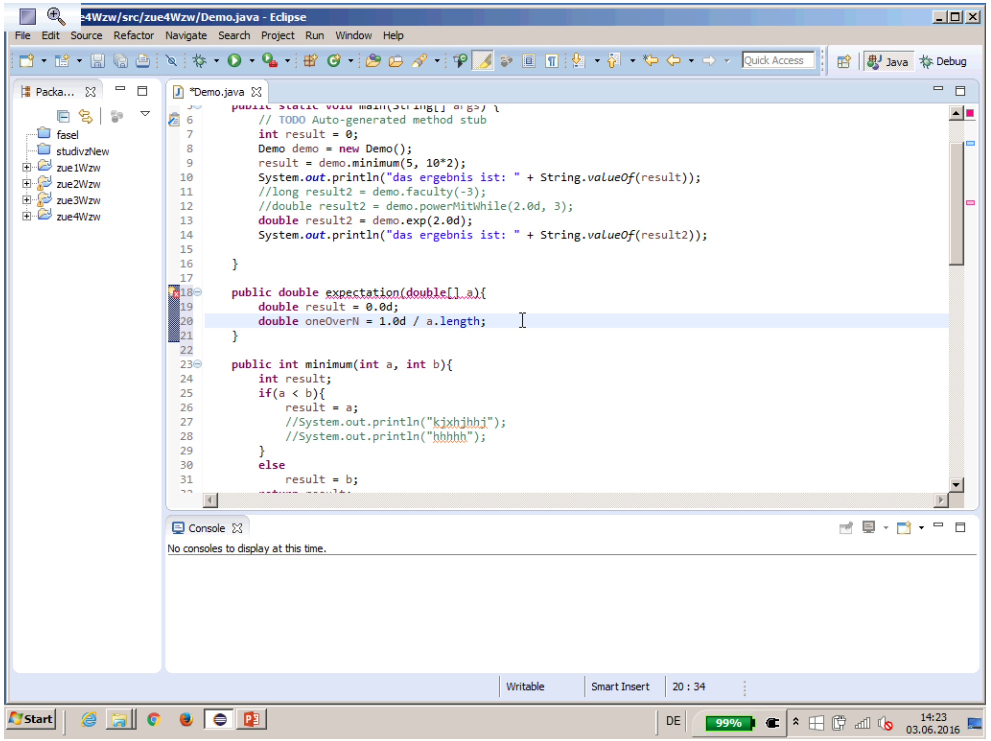
```
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));

    }

    public double expectation(double[] a){
        double result = 0.0d;

    }

    public int minimum(int a, int b){
        int result;
        if(a < b){
            result = a;
            //System.out.println("kjxhjhhj");
            //System.out.println("hhhhh");
        }
        else
            result = b;
        return result;
```

No consoles to display at this time.



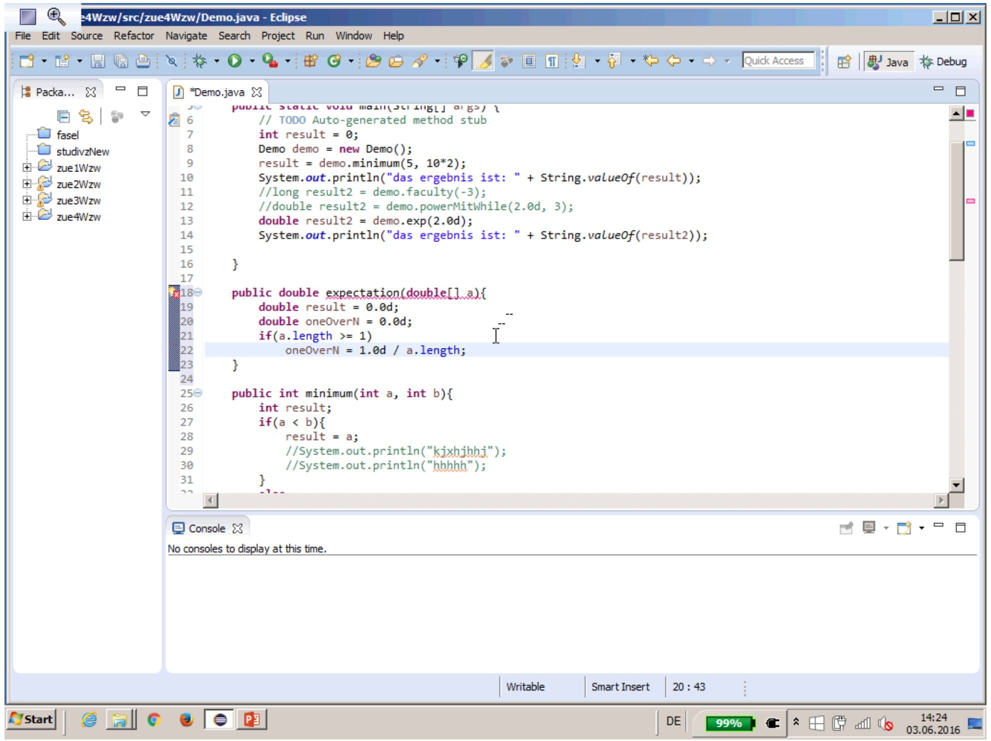
```
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));

    }

    public double expectation(double[] a){
        double result = 0.0d;
        double oneOverN = 1.0d / 
    }

    public int minimum(int a, int b){
        int result;
        if(a < b){
            result = a;
            //System.out.println("kjxhjhhj");
            //System.out.println("hhhhh");
        }
        else
            result = b;
        return result;
```

No consoles to display at this time.



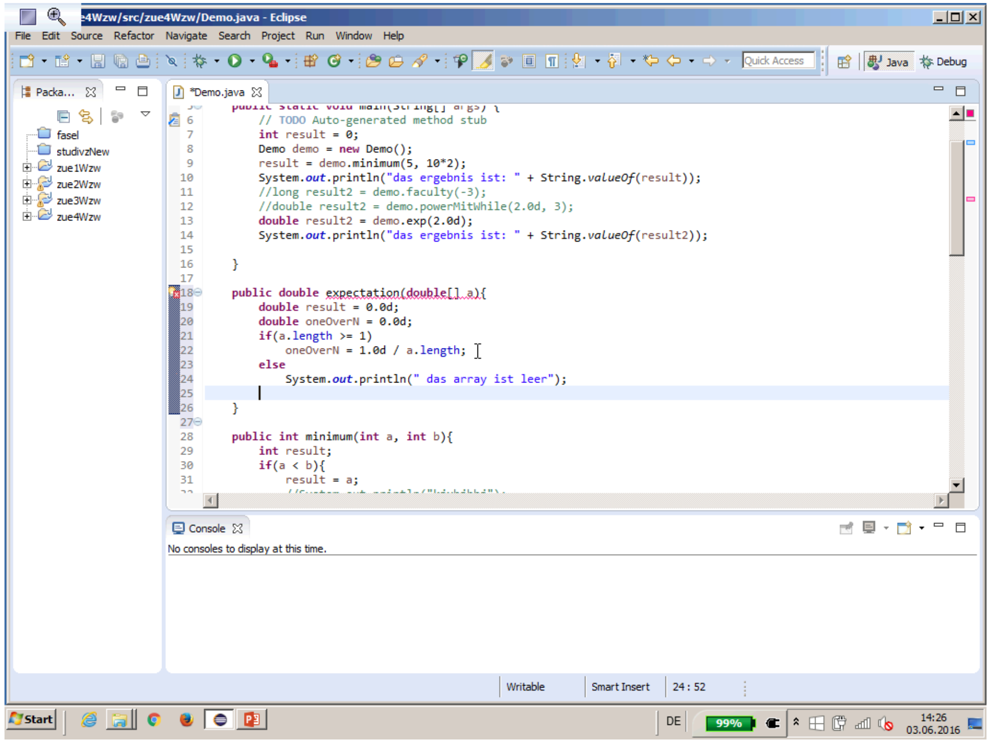
```
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));

    }

    public double expectation(double[] a){
        double result = 0.0d;
        double oneOverN = 1.0d / a.length;
    }

    public int minimum(int a, int b){
        int result;
        if(a < b){
            result = a;
            //System.out.println("kjxhjhhj");
            //System.out.println("hhhhh");
        }
        else
            result = b;
        return result;
```

No consoles to display at this time.



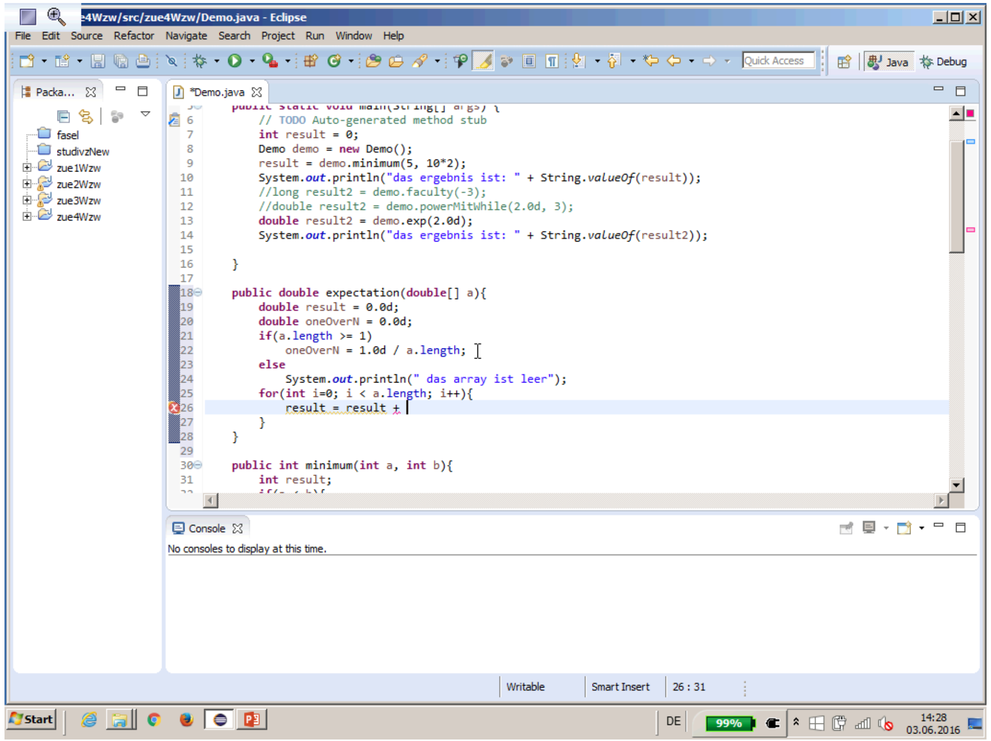
```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));

    }

    public double expectation(double[] a){
        double result = 0.0d;
        double oneOverN = 0.0d;
        if(a.length >= 1)
            oneOverN = 1.0d / a.length;
    }

    public int minimum(int a, int b){
        int result;
        if(a < b){
            result = a;
            //System.out.println("kjxhjhhj");
            //System.out.println("hhhhh");
        }
```

No consoles to display at this time.



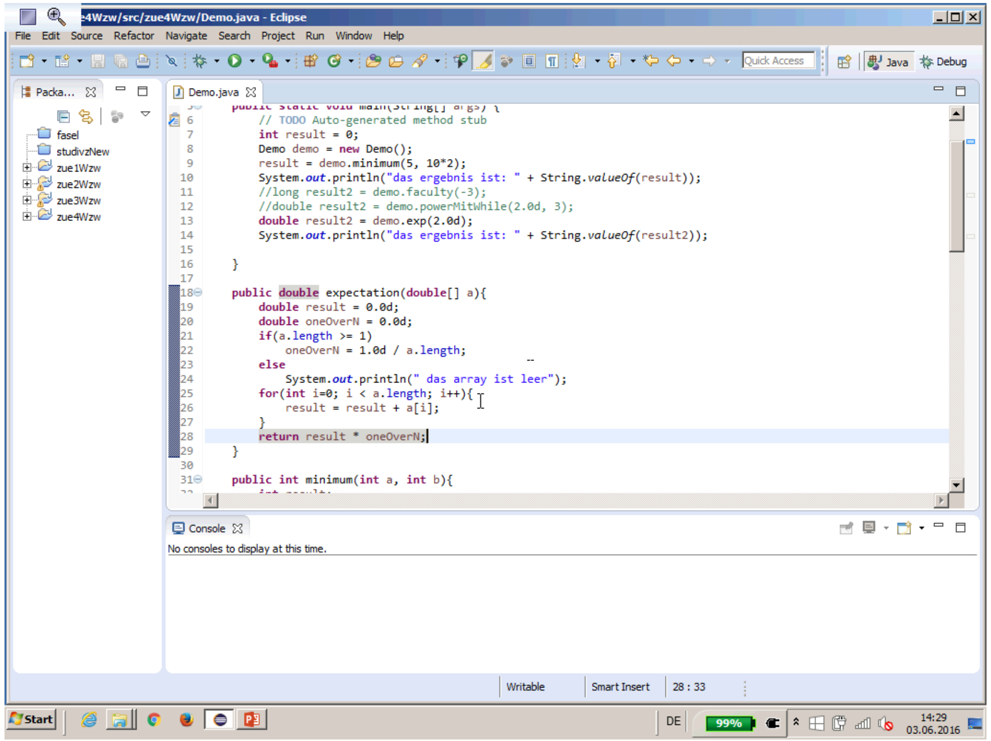
```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));

    }

    public double expectation(double[] a){
        double result = 0.0d;
        double oneOverN = 0.0d;
        if(a.length >= 1)
            oneOverN = 1.0d / a.length;
        else
            System.out.println(" das array ist leer");

    }

    public int minimum(int a, int b){
        int result;
        if(a < b){
            result = a;
```

No consoles to display at this time.



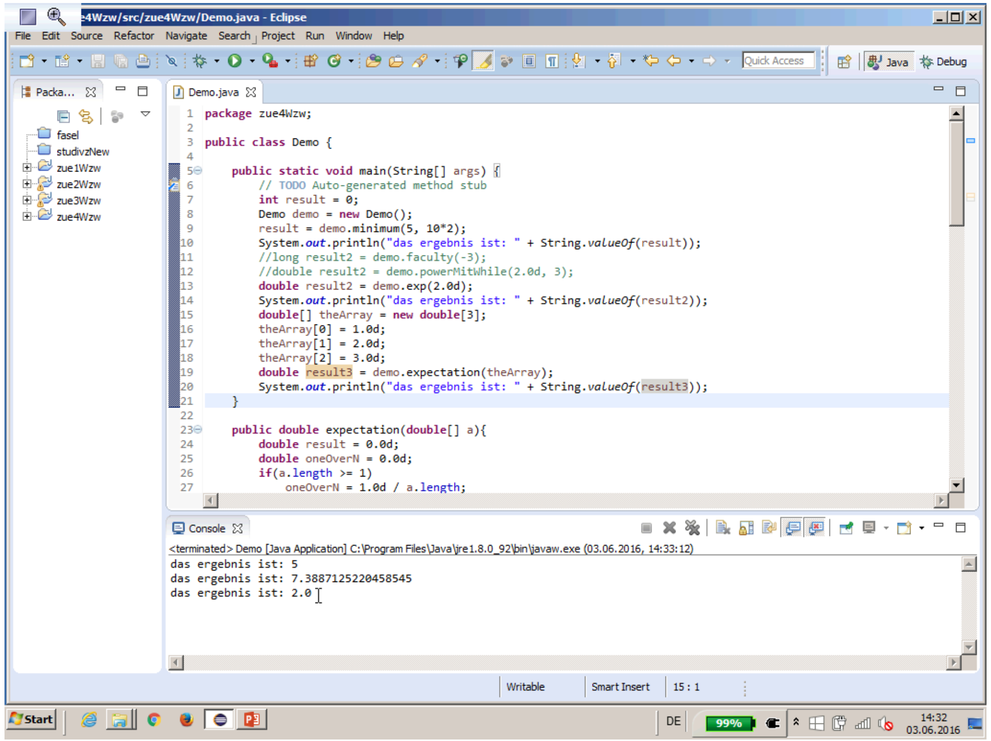
```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));

    }

    public double expectation(double[] a){
        double result = 0.0d;
        double oneOverN = 0.0d;
        if(a.length >= 1)
            oneOverN = 1.0d / a.length;
        else
            System.out.println(" das array ist leer");
        for(int i=0; i < a.length; i++){
            result = result + 
        }
    }

    public int minimum(int a, int b){
        int result;
```

No consoles to display at this time.



```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));

    }

    public double expectation(double[] a){
        double result = 0.0d;
        double oneOverN = 0.0d;
        if(a.length >= 1)
            oneOverN = 1.0d / a.length;
        else
            System.out.println(" das array ist leer");
        for(int i=0; i < a.length; i++){
            result = result + a[i];
        }
        return result * oneOverN;
    }

    public int minimum(int a, int b){
```

No consoles to display at this time.



```java
package zue4Wzw;

public class Demo {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
    }

    public double expectation(double[] a){
        double result = 0.0d;
        double oneOverN = 0.0d;
        if(a.length >= 1)
            oneOverN = 1.0d / a.length;
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 14:33:12)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
```

Aufgabe

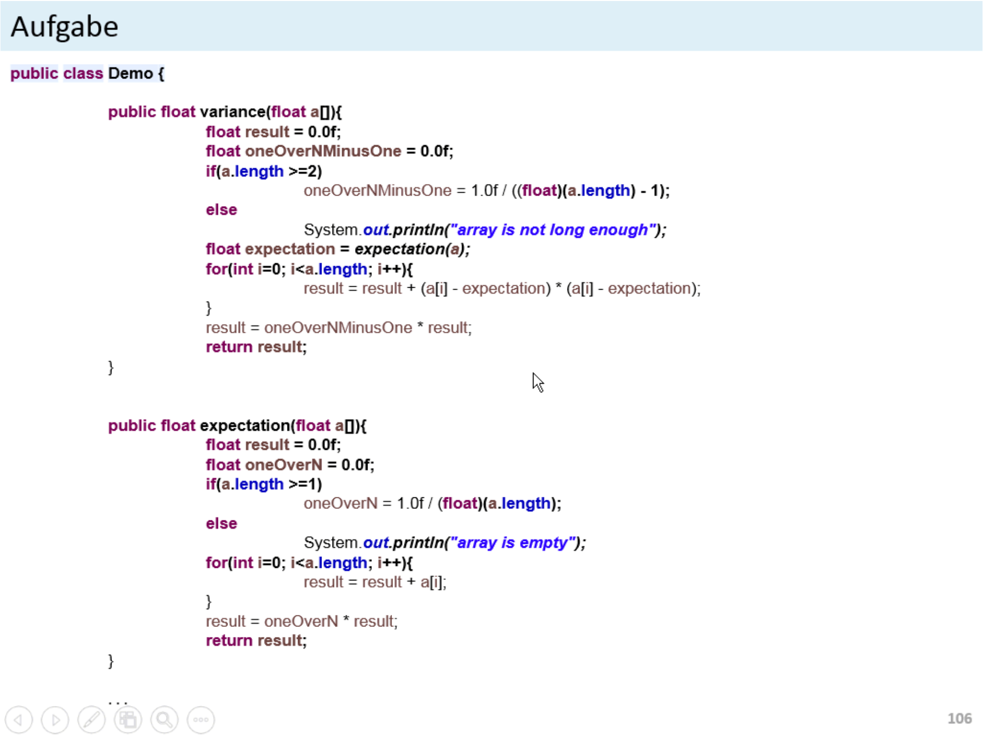
```java
public class Demo {

        public float variance(float a[]){
                float result = 0.0f;
                float oneOverNMinusOne = 0.0f;
                if(a.length >=1)
                        oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
                else
                        System.out.println("array is empty");
                float expectation = expectation(a);
                for(int i=0; i<a.length; i++){
                        result = result + (a[i] - expectation) * (a[i] - expectation);
                }
                result = oneOverNMinusOne * result;
                return result;
        }

        public float expectation(float a[]){
                float result = 0.0f;
                float oneOverN = 0.0f;
                if(a.length >=1)
                        oneOverN = 1.0f / (float)(a.length);
                else
                        System.out.println("array is empty");
                for(int i=0; i<a.length; i++){
                        result = result + a[i];
                }
                result = oneOverN * result;
                return result;
        }

        . . .
```

Aufgabe

```java
public class Demo {

        public float variance(float a[]){
                float result = 0.0f;
                float oneOverNMinusOne = 0.0f;
                if(a.length >=1)
                        oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
                else
                        System.out.println("array is empty");
                float expectation = expectation(a);
                for(int i=0; i<a.length; i++){
                        result = result + (a[i] - expectation) * (a[i] - expectation);
                }
                result = oneOverNMinusOne * result;
                return result;
        }

        public float expectation(float a[]){
                float result = 0.0f;
                float oneOverN = 0.0f;
                if(a.length >=1)
                        oneOverN = 1.0f / (float)(a.length);
                else
                        System.out.println("array is empty");
                for(int i=0; i<a.length; i++){
                        result = result + a[i];
                }
                result = oneOverN * result;
                return result;
        }

        . . .
```

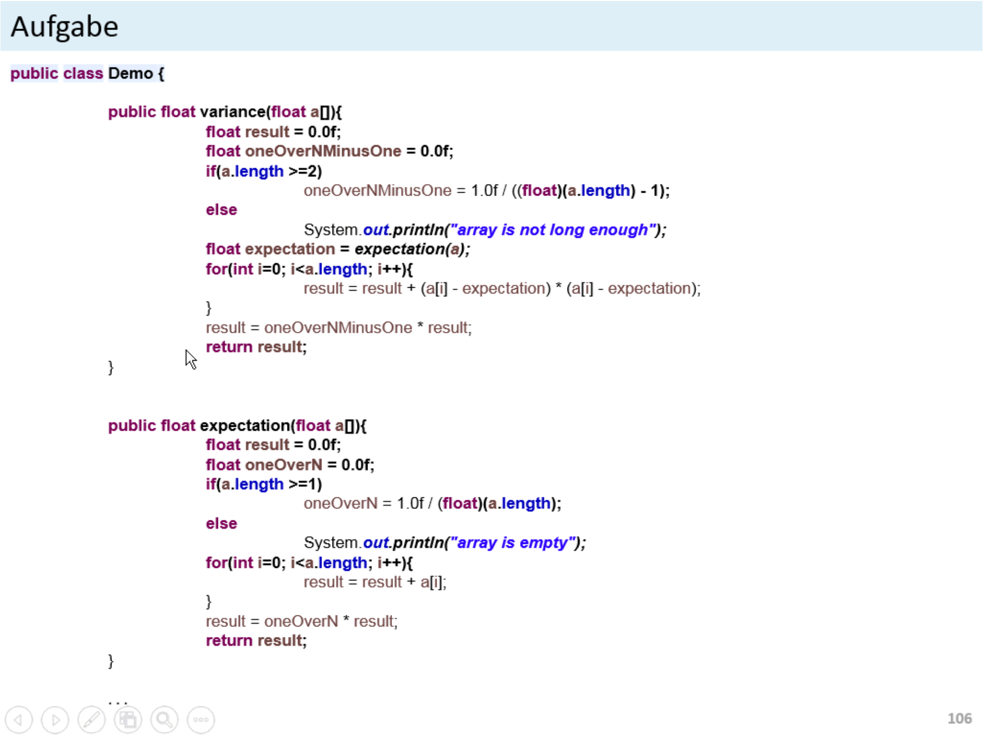
Aufgabe

```java
public class Demo {

        public float variance(float a[]){
                float result = 0.0f;
                float oneOverNMinusOne = 0.0f;
                if(a.length >=1)
                        oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
                else
                        System.out.println("array is empty");
                float expectation = expectation(a);
                for(int i=0; i<a.length; i++){
                        result = result + (a[i] - expectation) * (a[i] - expectation);
                }
                result = oneOverNMinusOne * result;
                return result;
        }

        public float expectation(float a[]){
                float result = 0.0f;
                float oneOverN = 0.0f;
                if(a.length >=1)
                        oneOverN = 1.0f / (float)(a.length);
                else
                        System.out.println("array is empty");
                for(int i=0; i<a.length; i++){
                        result = result + a[i];
                }
                result = oneOverN * result;
                return result;
        }

        . . .
```



Aufgabe

```
public class Demo {

        public float variance(float a[]){
                float result = 0.0f;
                float oneOverNMinusOne = 0.0f;
                if(a.length >=2)
                        oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
                else
                        System.out.println("array is not long enough");
                float expectation = expectation(a);
                for(int i=0; i<a.length; i++){
                        result = result + (a[i] - expectation) * (a[i] - expectation);
                }
                result = oneOverNMinusOne * result;
                return result;
        }


        public float expectation(float a[]){
                float result = 0.0f;
                float oneOverN = 0.0f;
                if(a.length >=1)
                        oneOverN = 1.0f / (float)(a.length);
                else
                        System.out.println("array is empty");
                for(int i=0; i<a.length; i++){
                        result = result + a[i];
                }
                result = oneOverN * result;
                return result;
        }

        . . .
```



Aufgabe

```
public class Demo {

        public float variance(float a[]){
                float result = 0.0f;
                float oneOverNMinusOne = 0.0f;
                if(a.length >=2)
                        oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
                else
                        System.out.println("array is not long enough");
                float expectation = expectation(a);
                for(int i=0; i<a.length; i++){
                        result = result + (a[i] - expectation) * (a[i] - expectation);
                }
                result = oneOverNMinusOne * result;
                return result;
        }


        public float expectation(float a[]){
                float result = 0.0f;
                float oneOverN = 0.0f;
                if(a.length >=1)
                        oneOverN = 1.0f / (float)(a.length);
                else
                        System.out.println("array is empty");
                for(int i=0; i<a.length; i++){
                        result = result + a[i];
                }
                result = oneOverN * result;
                return result;
        }

        . . .
```

106

Aufgabe

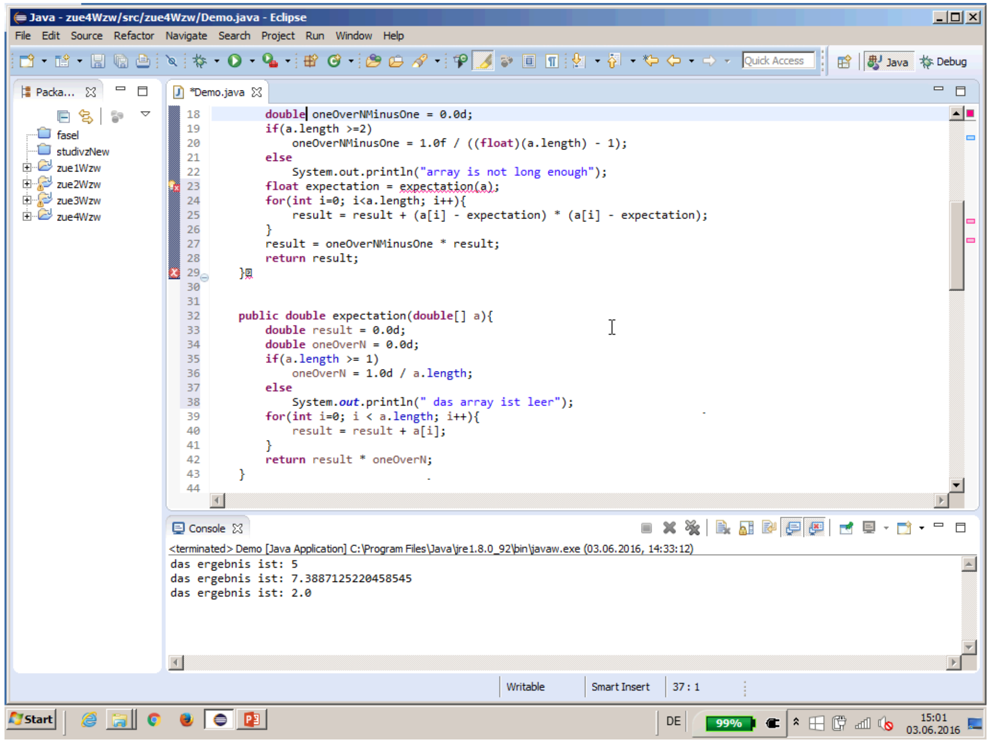
```
public class Demo {

        public float variance(float a[]){
                float result = 0.0f;
                float oneOverNMinusOne = 0.0f;
                if(a.length >=2)
                        oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
                else
                        System.out.println("array is not long enough");
                float expectation = expectation(a);
                for(int i=0; i<a.length; i++){
                        result = result + (a[i] - expectation) * (a[i] - expectation);
                }
                result = oneOverNMinusOne * result;
                return result;
        }


        public float expectation(float a[]){
                float result = 0.0f;
                float oneOverN = 0.0f;
                if(a.length >=1)
                        oneOverN = 1.0f / (float)(a.length);
                else
                        System.out.println("array is empty");
                for(int i=0; i<a.length; i++){
                        result = result + a[i];
                }
                result = oneOverN * result;
                return result;
        }

        . . .
```

106

Aufgabe

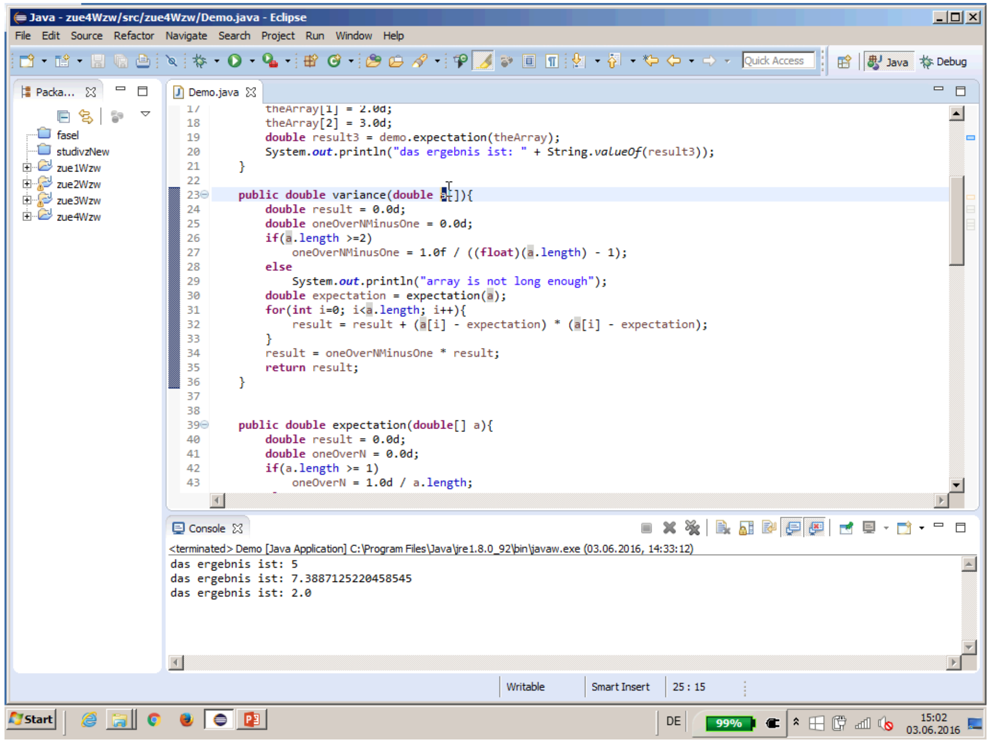
```
public class Demo {

        public float variance(float a[]){
                float result = 0.0f;
                float oneOverNMinusOne = 0.0f;
                if(a.length >=2)
                        oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
                else
                        System.out.println("array is not long enough");
                float expectation = expectation(a);
                for(int i=0; i<a.length; i++){
                        result = result + (a[i] - expectation) * (a[i] - expectation);
                }
                result = oneOverNMinusOne * result;
                return result;
        }


        public float expectation(float a[]){
                float result = 0.0f;
                float oneOverN = 0.0f;
                if(a.length >=1)
                        oneOverN = 1.0f / (float)(a.length);
                else
                        System.out.println("array is empty");
                for(int i=0; i<a.length; i++){
                        result = result + a[i];
                }
                result = oneOverN * result;
                return result;
        }

        . . .
```

106



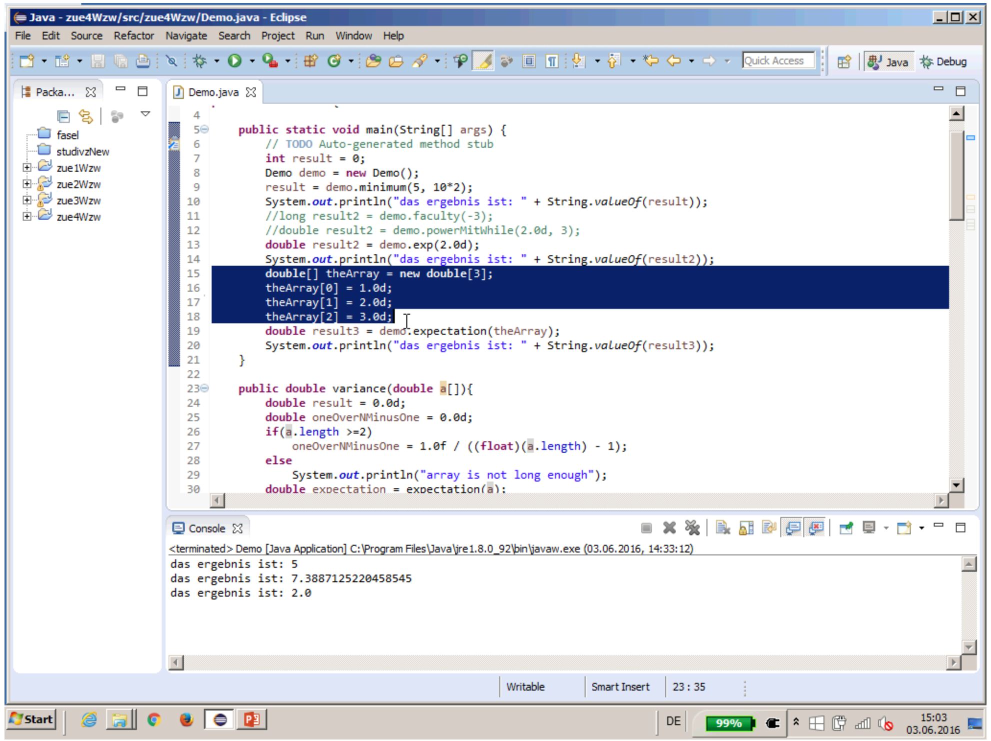
```java
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        float expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
        result = oneOverNMinusOne * result;
        return result;
    }


    public double expectation(double[] a){
        double result = 0.0d;
        double oneOverN = 0.0d;
        if(a.length >= 1)
            oneOverN = 1.0d / a.length;
        else
            System.out.println(" das array ist leer");
        for(int i=0; i < a.length; i++){
            result = result + a[i];
        }
        return result * oneOverN;
    }
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 14:33:12)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
```



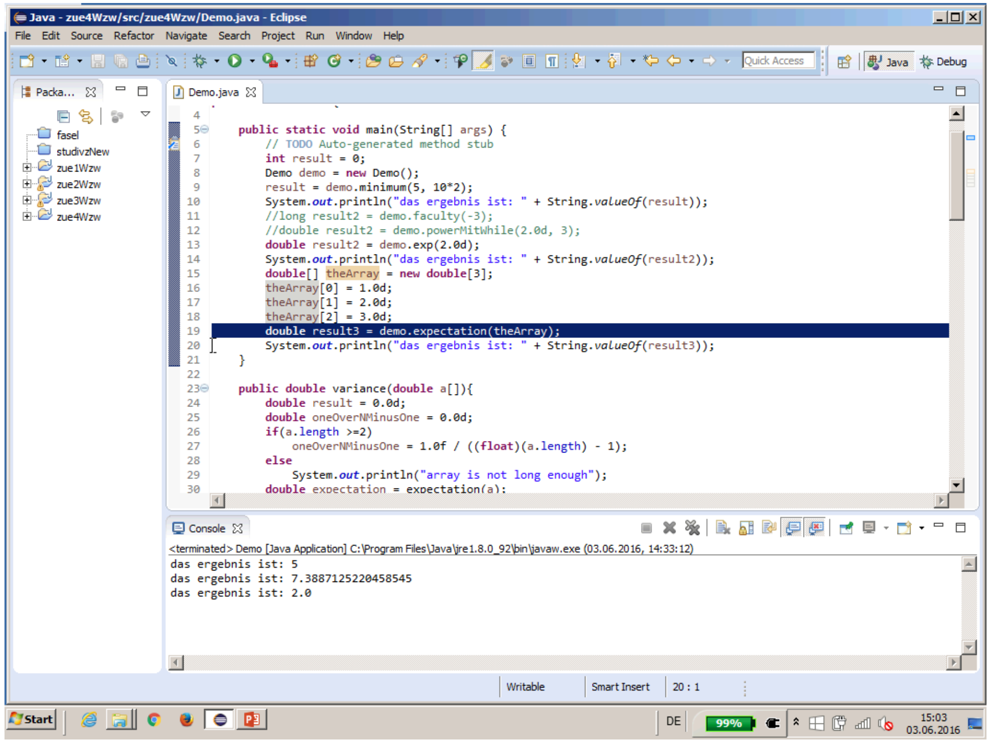
```java
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
        result = oneOverNMinusOne * result;
        return result;
    }


    public double expectation(double[] a){
        double result = 0.0d;
        double oneOverN = 0.0d;
        if(a.length >= 1)
            oneOverN = 1.0d / a.length;
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 14:33:12)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
```



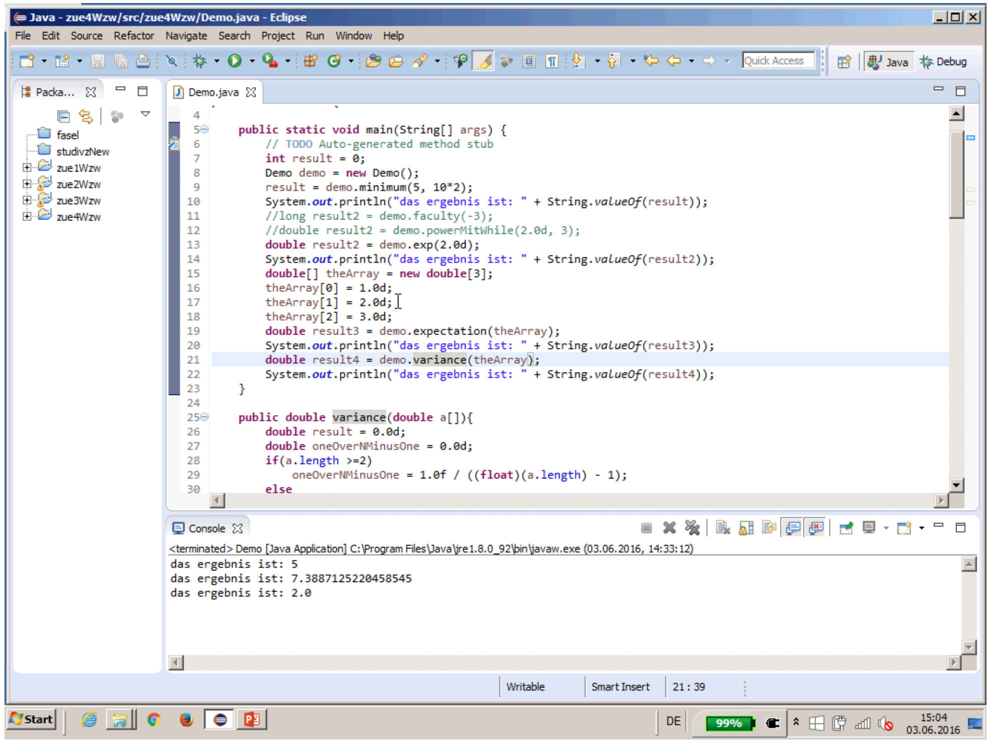
```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 14:33:12)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
```



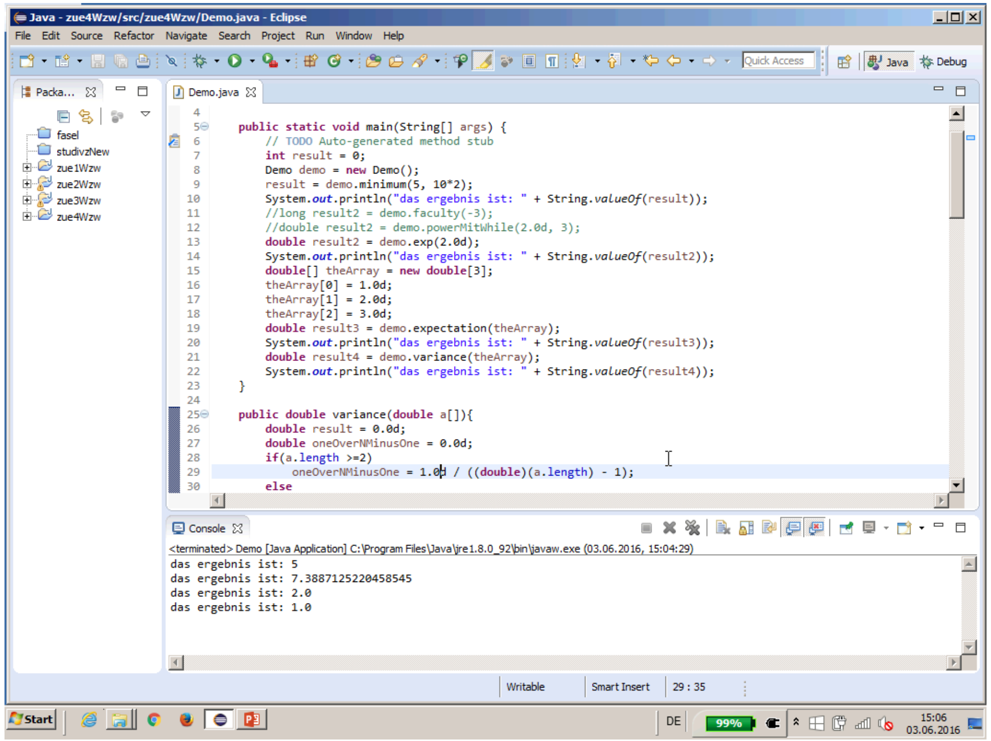
```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 14:33:12)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
```



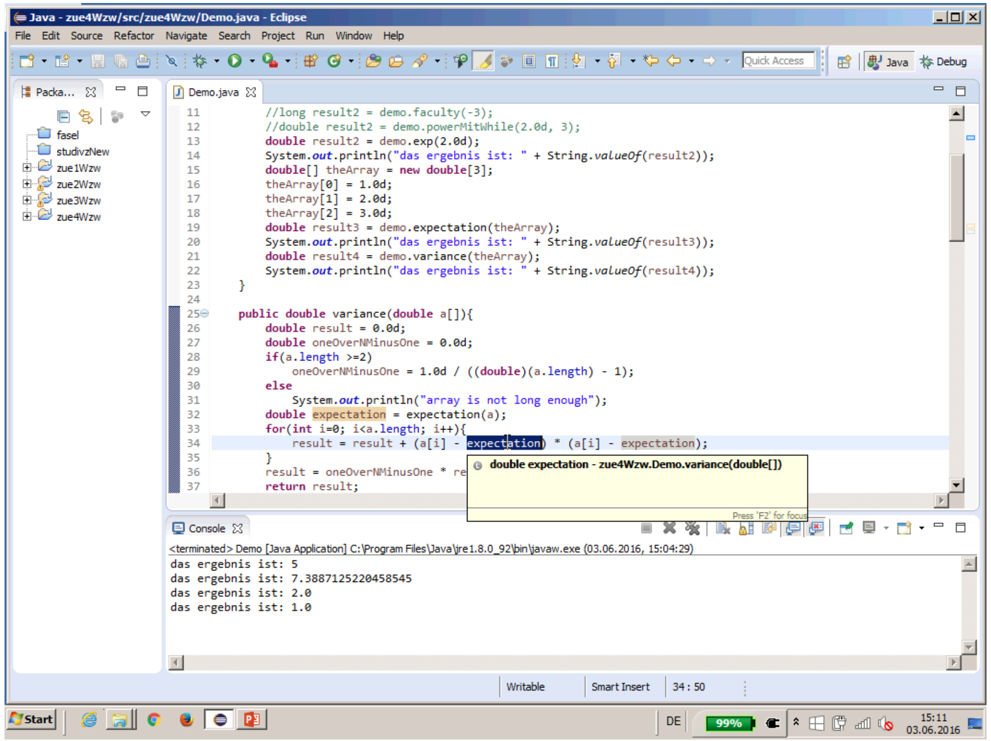
```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
        else
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 14:33:12)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
```



```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int result = 0;
        Demo demo = new Demo();
        result = demo.minimum(5, 10*2);
        System.out.println("das ergebnis ist: " + String.valueOf(result));
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:04:29)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
```

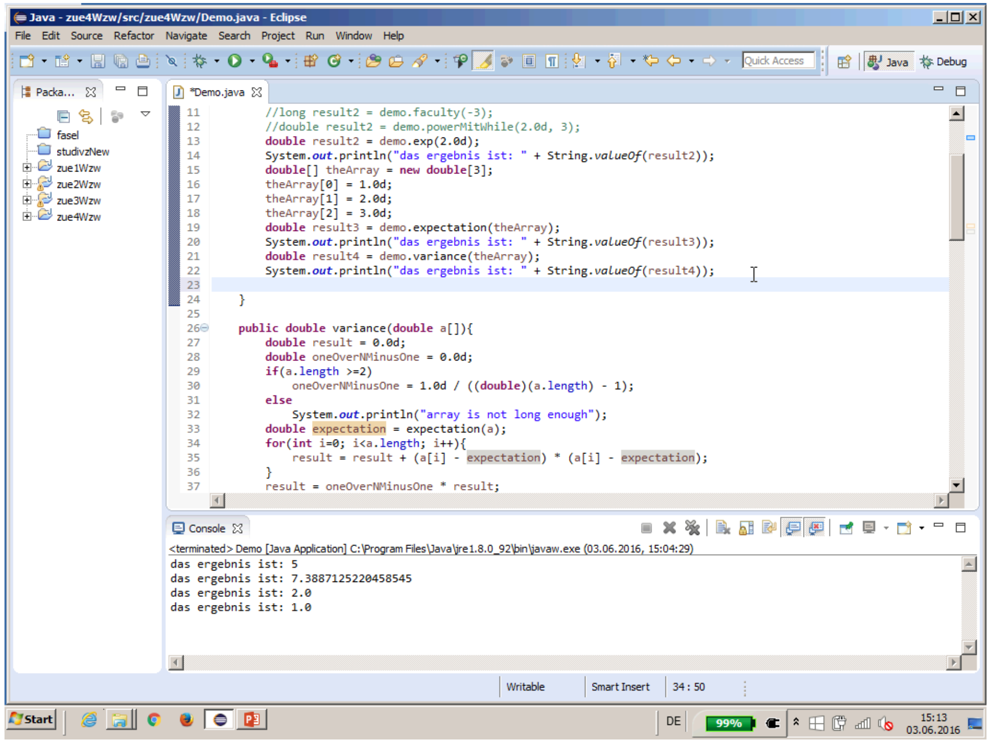


```java
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
        result = oneOverNMinusOne * re
        return result;
```

double expectation - zue4Wzw.Demo.variance(double[])

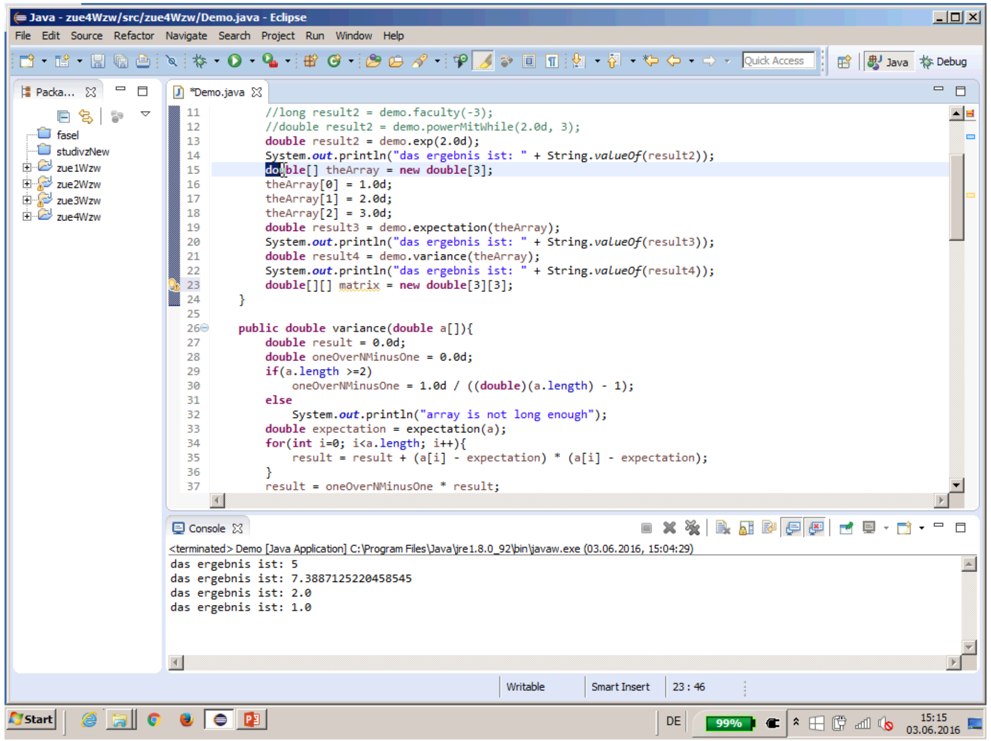
Press 'F2' for focus

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:04:29)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
```



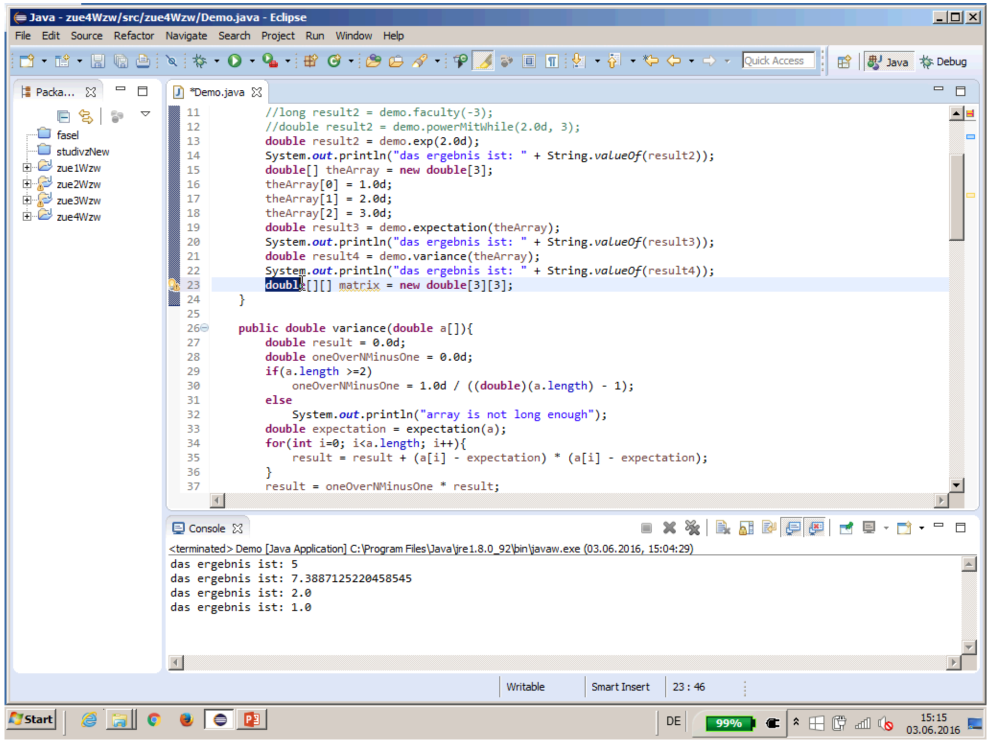
```java
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));

    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
        result = oneOverNMinusOne * result;
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:04:29)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
```



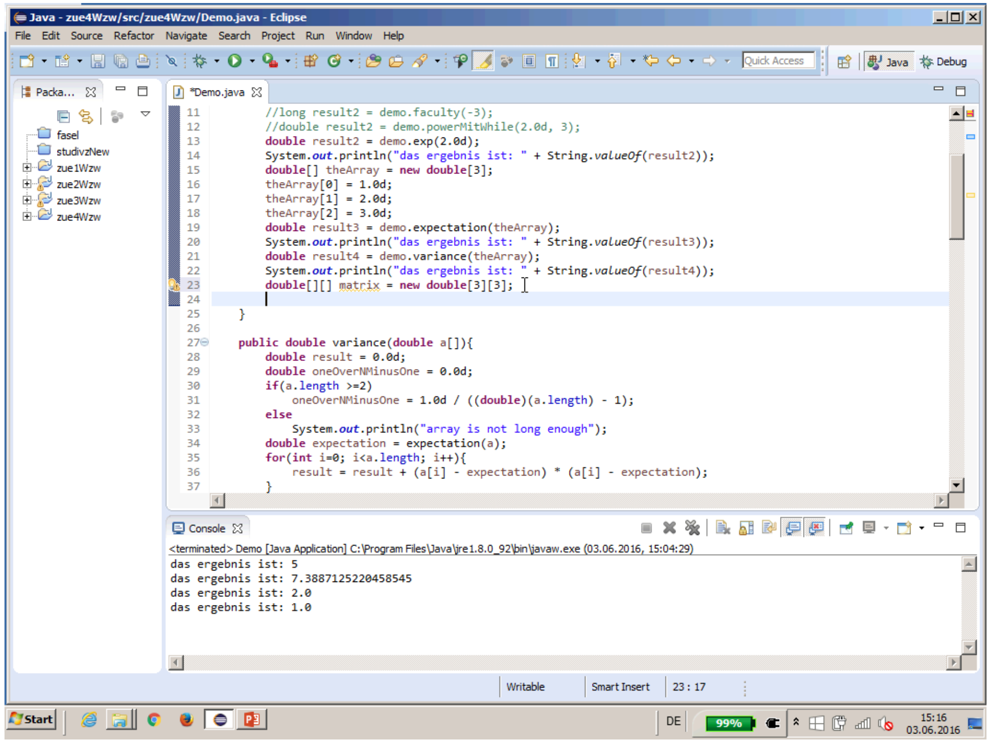
```java
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
        double[][] matrix = new double[3][3];
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
        result = oneOverNMinusOne * result;
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:04:29)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
```



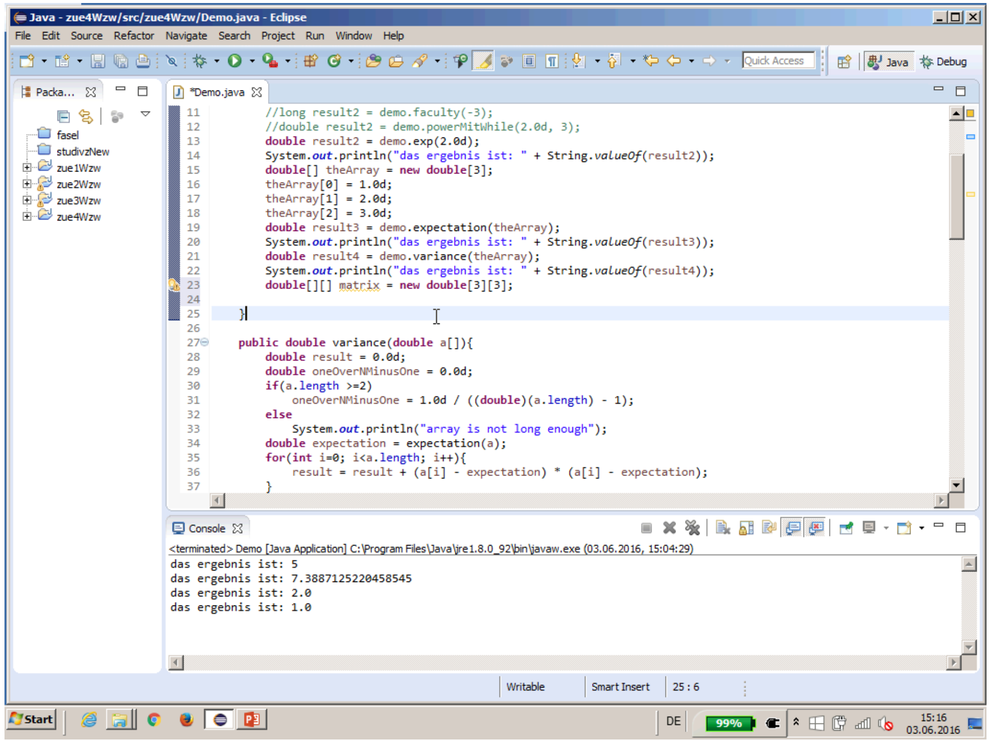
```java
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
        double[][] matrix = new double[3][3];
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
        result = oneOverNMinusOne * result;
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:04:29)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
```



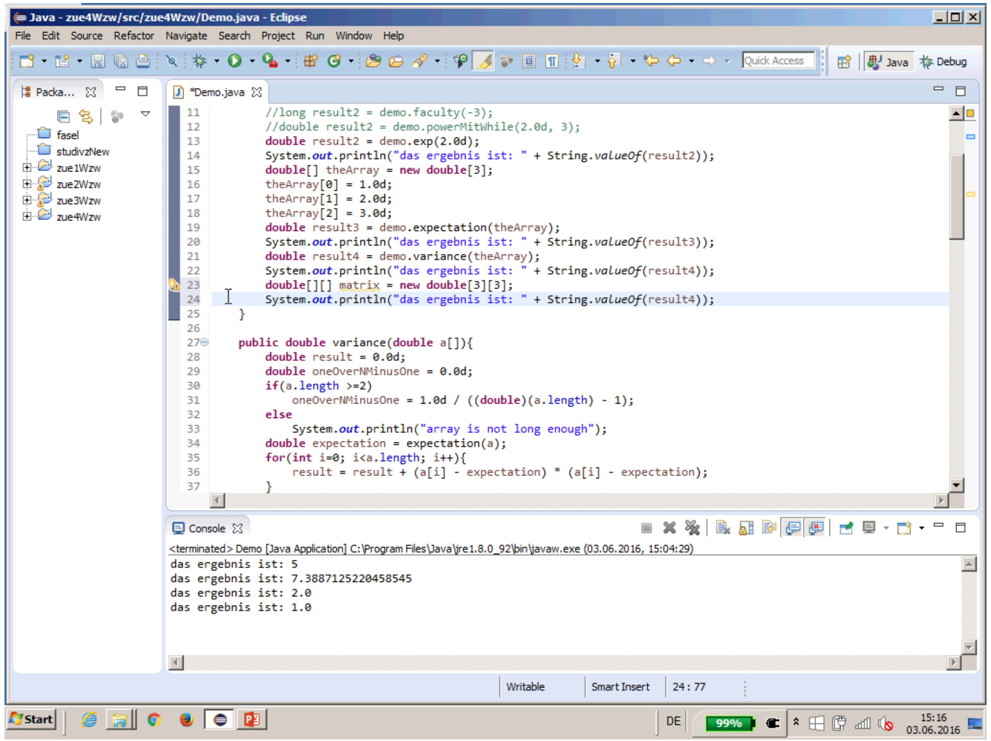
```java
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
        double[][] matrix = new double[3][3];

    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:04:29)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
```



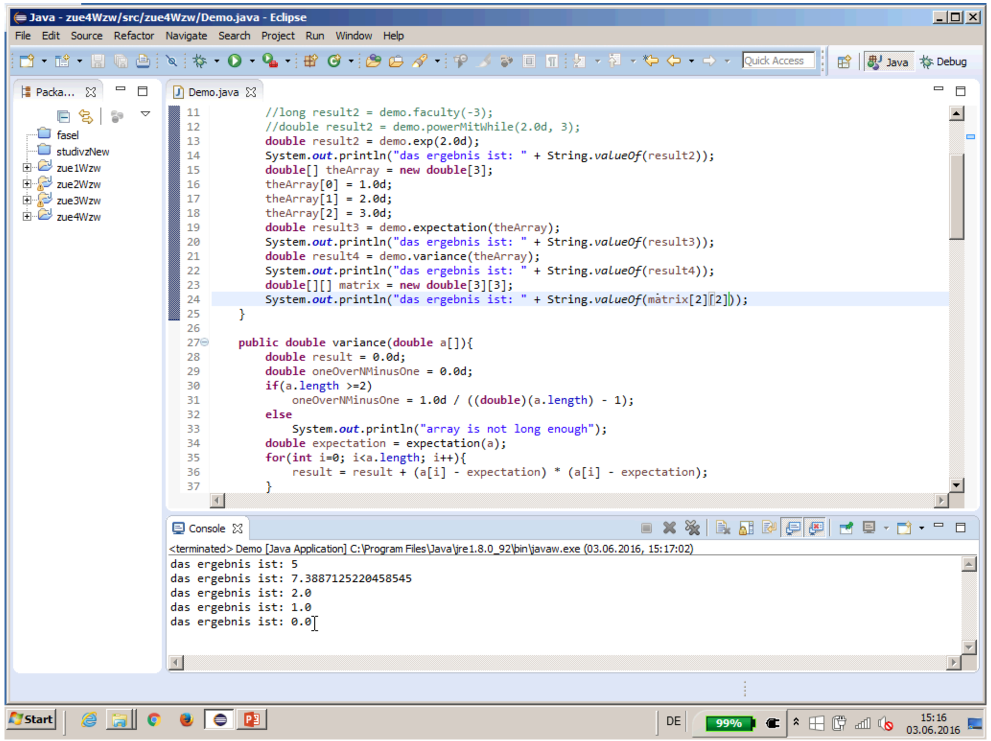
```java
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
        double[][] matrix = new double[3][3];

    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:04:29)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
```



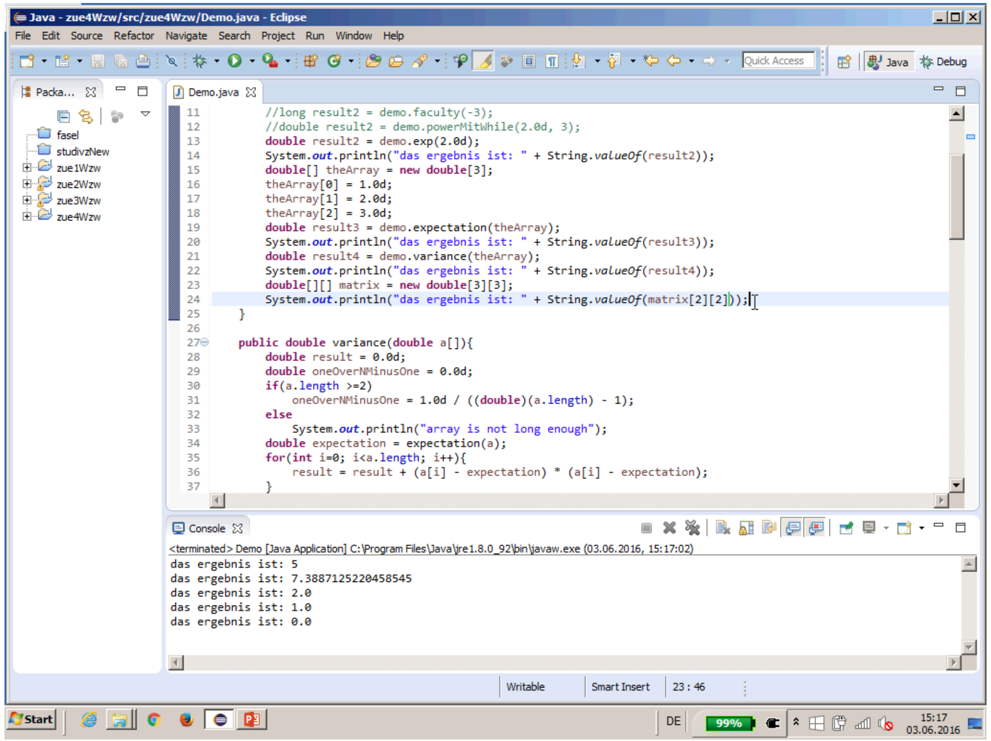
```java
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
        double[][] matrix = new double[3][3];
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:04:29)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
```



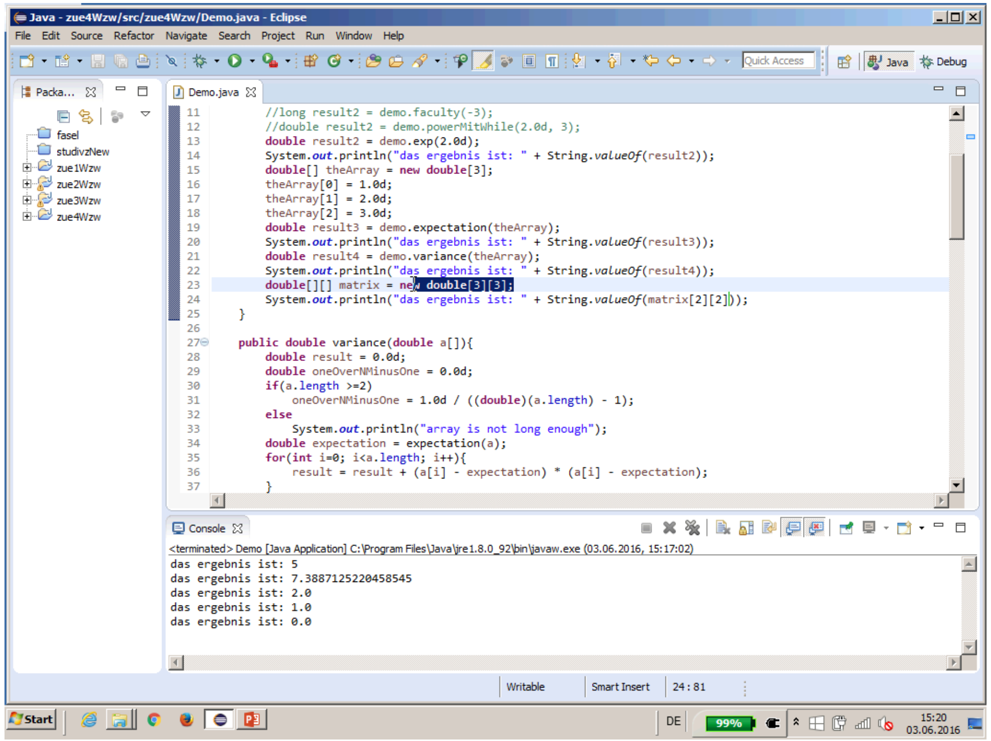
```java
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
        double[][] matrix = new double[3][3];
        System.out.println("das ergebnis ist: " + String.valueOf(matrix[2][2]));
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:17:02)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```



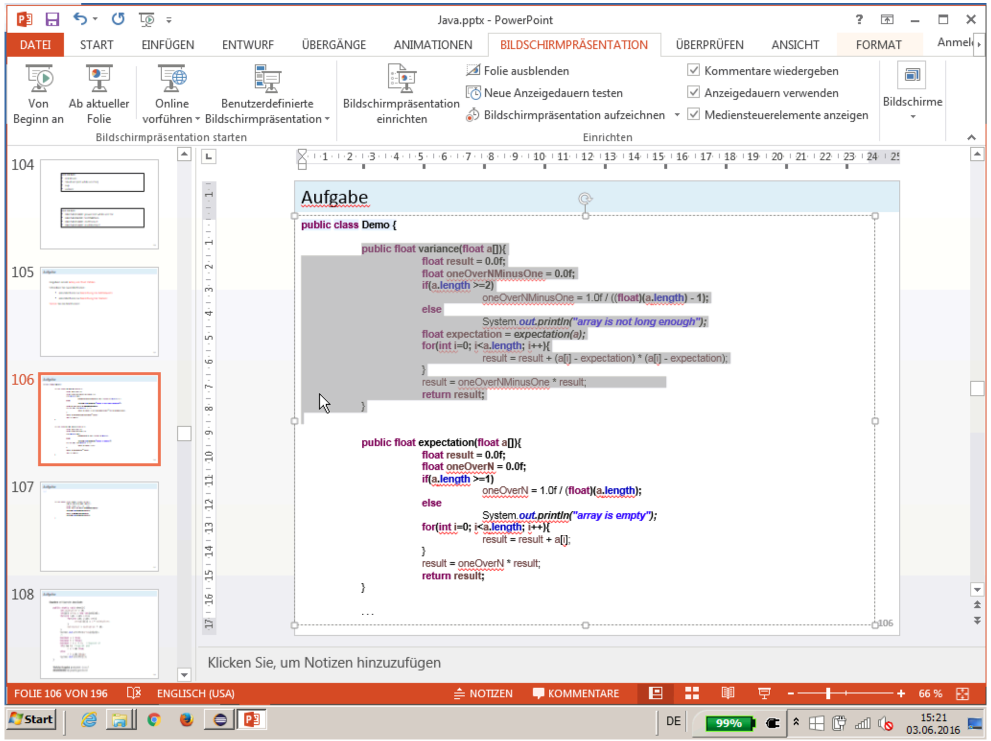
```java
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
        double[][] matrix = new double[3][3];
        System.out.println("das ergebnis ist: " + String.valueOf(matrix[2][2]));
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:17:02)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```



```java
        //long result2 = demo.faculty(-3);
        //double result2 = demo.powerMitWhile(2.0d, 3);
        double result2 = demo.exp(2.0d);
        System.out.println("das ergebnis ist: " + String.valueOf(result2));
        double[] theArray = new double[3];
        theArray[0] = 1.0d;
        theArray[1] = 2.0d;
        theArray[2] = 3.0d;
        double result3 = demo.expectation(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result3));
        double result4 = demo.variance(theArray);
        System.out.println("das ergebnis ist: " + String.valueOf(result4));
        double[][] matrix = new double[3][3];
        System.out.println("das ergebnis ist: " + String.valueOf(matrix[2][2]));
    }

    public double variance(double a[]){
        double result = 0.0d;
        double oneOverNMinusOne = 0.0d;
        if(a.length >=2)
            oneOverNMinusOne = 1.0d / ((double)(a.length) - 1);
        else
            System.out.println("array is not long enough");
        double expectation = expectation(a);
        for(int i=0; i<a.length; i++){
            result = result + (a[i] - expectation) * (a[i] - expectation);
        }
```

```
<terminated> Demo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (03.06.2016, 15:17:02)
das ergebnis ist: 5
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```



Aufgabe

```
public class Demo {

        public float variance(float a[]){
                float result = 0.0f;
                float oneOverNMinusOne = 0.0f;
                if(a.length >=2)
                        oneOverNMinusOne = 1.0f / ((float)(a.length) - 1);
                else
                        System.out.println("array is not long enough");
                float expectation = expectation(a);
                for(int i=0; i<a.length; i++){
                        result = result + (a[i] - expectation) * (a[i] - expectation);
                }
                result = oneOverNMinusOne * result;
                return result;
        }

        public float expectation(float a[]){
                float result = 0.0f;
                float oneOverN = 0.0f;
                if(a.length >=1)
                        oneOverN = 1.0f / (float)(a.length);
                else
                        System.out.println("array is empty");
                for(int i=0; i<a.length; i++){
                        result = result + a[i];
                }
                result = oneOverN * result;
                return result;
        }

        . . .
```

106

Aufgabe

Gegeben sei folgender **Java Code**:

```
public static void demo(){
        int multiplier = 10;
        int[][] array = new int[10][10];
        for(int i=0; i<10; i++){
                for(int j=0; j<10; j++){
                        array[i][j] = j * multiplier;
                }
                multiplier = multiplier * 10;
        }
        System.out.println(array[2][2]);
        //
        boolean a = true;
        boolean b = false;
        boolean c = a || b;  //logical or
        if(a && b) //logical and
                c = c && true;
        else
                c = c && false;
        System.out.println(c);
}
```

Welche Ausgabe produziert demo?
BEGRÜNDEN Sie jeweils ganz kurz!

Aufgabe

Gegeben sei folgender **Java Code**:

```
public static void demo(){
        int multiplier = 10;
        int[][] array = new int[10][10];
        for(int i=0; i<10; i++){
                for(int j=0; j<10; j++){
                        array[i][j] = j * multiplier;
                }
                multiplier = multiplier * 10;
        }
        System.out.println(array[2][2]);
        //
        boolean a = true;
        boolean b = false;
        boolean c = a || b;  //logical or
        if(a && b) //logical and
                c = c && true;
        else
                c = c && false;
        System.out.println(c);
}
```

Welche Ausgabe produziert demo?
BEGRÜNDEN Sie jeweils ganz kurz!

Aufgabe

Gegeben sei folgender **Java Code**:

```
public static void demo(){
        int multiplier = 10;
        int[][] array = new int[10][10];
        for(int i=0; i<10; i++){
                for(int j=0; j<10; j++){
                        array[i][j] = j * multiplier;
                }
                multiplier = multiplier * 10;
        }
        System.out.println(array[2][2]);
        //
        boolean a = true;
        boolean b = false;
        boolean c = a || b;  //logical or
        if(a && b) //logical and
                c = c && true;
        else
                c = c && false;
        System.out.println(c);
}
```

Welche Ausgabe produziert demo?
BEGRÜNDEN Sie jeweils ganz kurz!

Aufgabe

Gegeben sei folgender **Java Code**:

```
public static void demo(){
        int multiplier = 10;
        int[][] array = new int[10][10];
        for(int i=0; i<10; i++){
                for(int j=0; j<10; j++){
                        array[i][j] = j * multiplier;
                }
                multiplier = multiplier * 10;
        }
        System.out.println(array[2][2]);
        //
        boolean a = true;
        boolean b = false;
        boolean c = a || b;  //logical or
        if(a && b) //logical and
                c = c && true;
        else
                c = c && false;
        System.out.println(c);
}
```

Welche Ausgabe produziert demo?
BEGRÜNDEN Sie jeweils ganz kurz!

Aufgabe

Gegeben sei folgender **Java Code**:

```
public static void demo(){
        int multiplier = 10;
        int[][] array = new int[10][10];
        for(int i=0; i<10; i++){
                for(int j=0; j<10; j++){
                        array[i][j] = j * multiplier;
                }
                multiplier = multiplier * 10;
        }
        System.out.println(array[2][2]);
        //
        boolean a = true;
        boolean b = false;
        boolean c = a || b;  //logical or
        if(a && b) //logical and
                c = c && true;
        else
                c = c && false;
        System.out.println(c);
}
```

Welche Ausgabe produziert demo?
BEGRÜNDEN Sie jeweils ganz kurz!

Aufgabe

Gegeben seien zwei Java Methoden, die die Funktion $f(x) = x^n$ berechnen:

```
public static double powerWithWhile(double x, int n)
        double result = 1.0;
        int i = 1;
        while (i <= n) {
                result = □;
                i++;
        }
        return result;
}

public static double powerWithFor(double x, int n)
        double result = 1.0;
        for (□) {
                result = result * x;
        }
        return result;
}
```

Ergänzen Sie die fehlenden Elemente **sinnvoll**!