

JUNIT

- 1) Normales Java-Projekt erstellen "JUnit_bsp11" und einfügen eigener Testroutinen

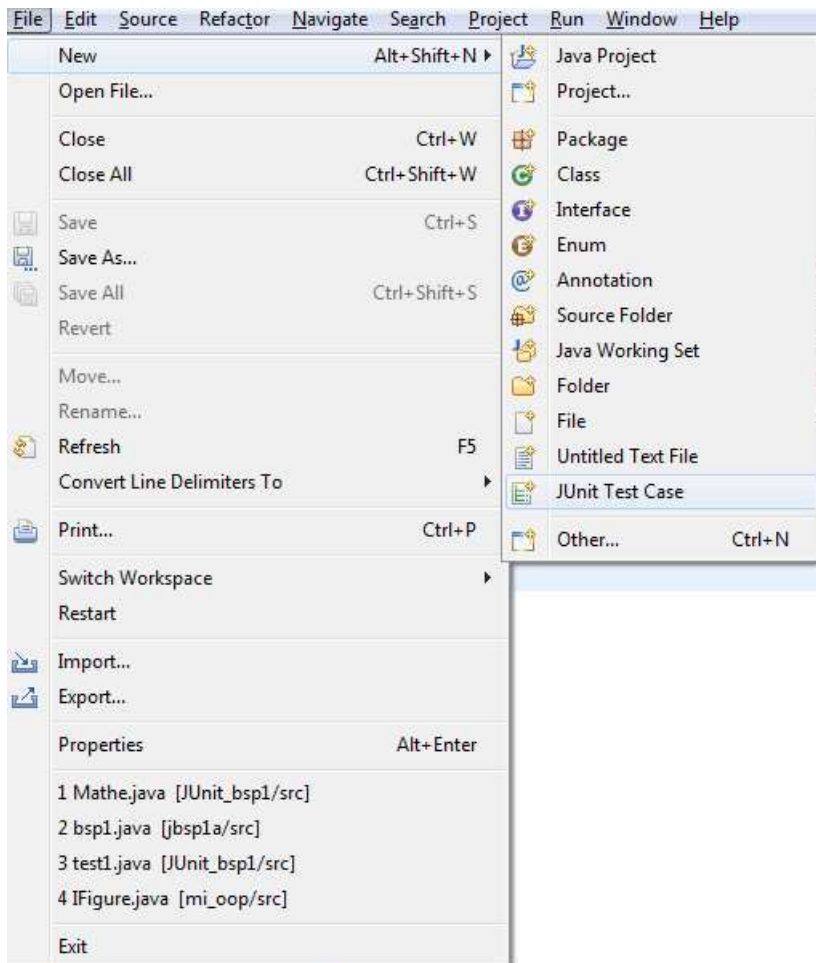
```
public class Mathe {  
  
    public double add(double x, double y) {  
        return x + y;  
    }  
  
    public int add(int x, int y) {  
        return x + y;  
    }  
  
    public void test1() {  
        int a, b, c;  
        Mathe myclass = new Mathe(); // neue Klasse  
  
        c = myclass.add(5, 4);  
        if (c != 9) {  
            System.out.println("Fehler beim 1. Test Add: " + c);  
        }  
    }  
  
    public void test2() {  
        double a, b, c;  
        Mathe myclass = new Mathe(); // neue Klasse  
  
        c = myclass.add(5.033, 4.343);  
        if (c != 9.376) {  
            System.out.println("Fehler beim 2. Test Add: " + c);  
        }  
    } // test2  
  
    public static void main(String[] args) {  
        Mathe ma = new Mathe();  
        ma.test1();  
        ma.test2();  
    }  
}  
  
} // Mathe
```

Eclipse mit jUnit

1) Java Projekt erstellen

```
public class Mathe {  
  
    public double add(double x, double y) {  
        return x + y;  
    }  
  
    public int add(int x, int y) {  
        return x + y;  
    }  
  
    public static void main(String[] args) {  
        System.out.println("in Main von JUnit_bsp2");  
    }  
  
} // Mathe
```

2) JUnit Testsuite hinzufügen



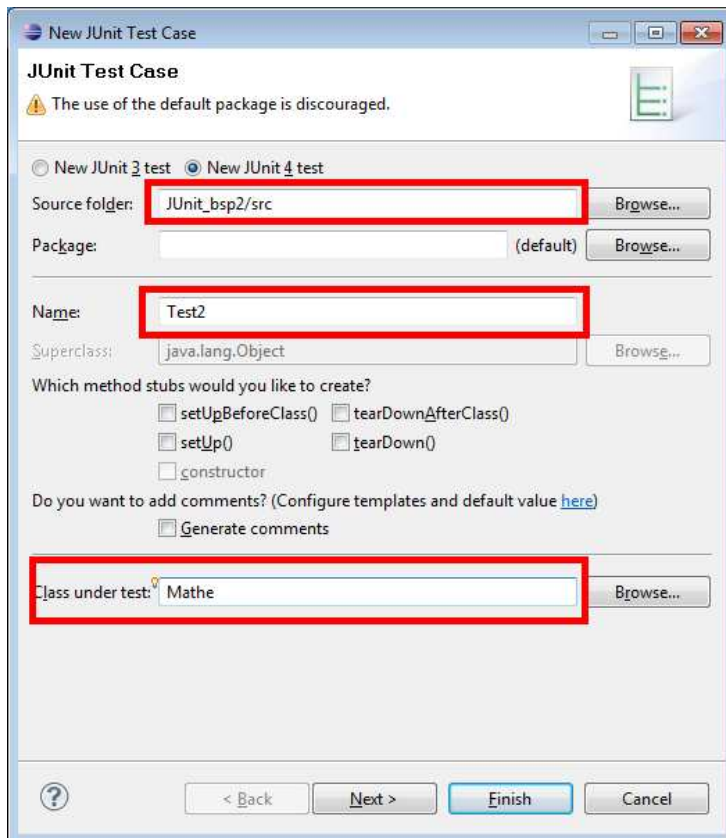


Abbildung 1 Erstellen des Testcase

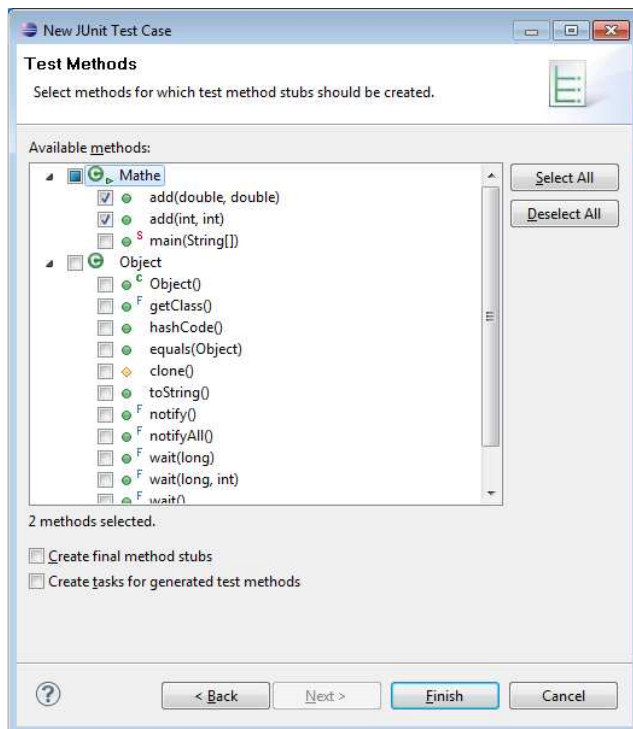
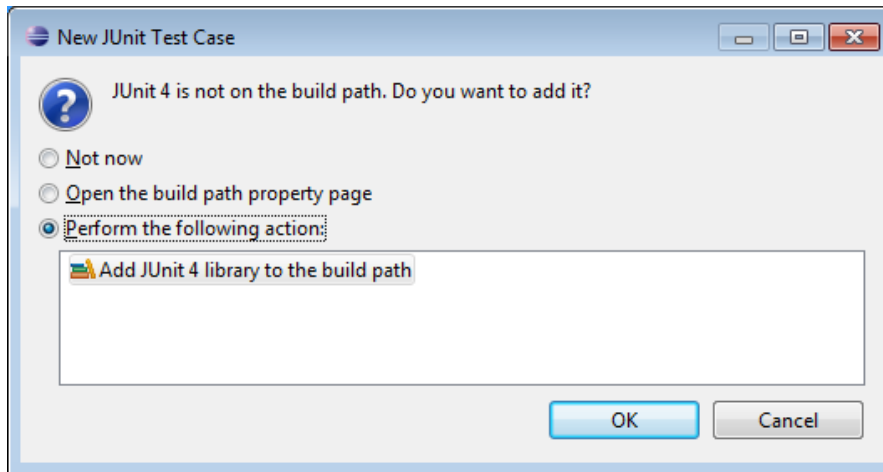


Abbildung 2 Auswahl der Methoden



Testquellcode

```
import static org.junit.Assert.*;

import org.junit.Test;

public class Test2 {

    @Test
    public void testAddDoubleDouble() {
        fail("Not yet implemented1");
    }

    @Test
    public void testAddIntInt() {
        fail("Not yet implemented2");
    }
}
```

Ergänzen des Quellcodes:

```
import static org.junit.Assert.*;

import org.junit.Test;

public class Test2 {

    @Test
    public void testAddInt() {
        int a, b, c;
        Mathe myclass = new Mathe(); // neue Klasse

        c = myclass.add(5, 4);
        assertEquals( c, 9);
    }

    @Test
    public void testAddDouble1() {
        double a, b, c;
        Mathe myclass = new Mathe(); // neue Klasse

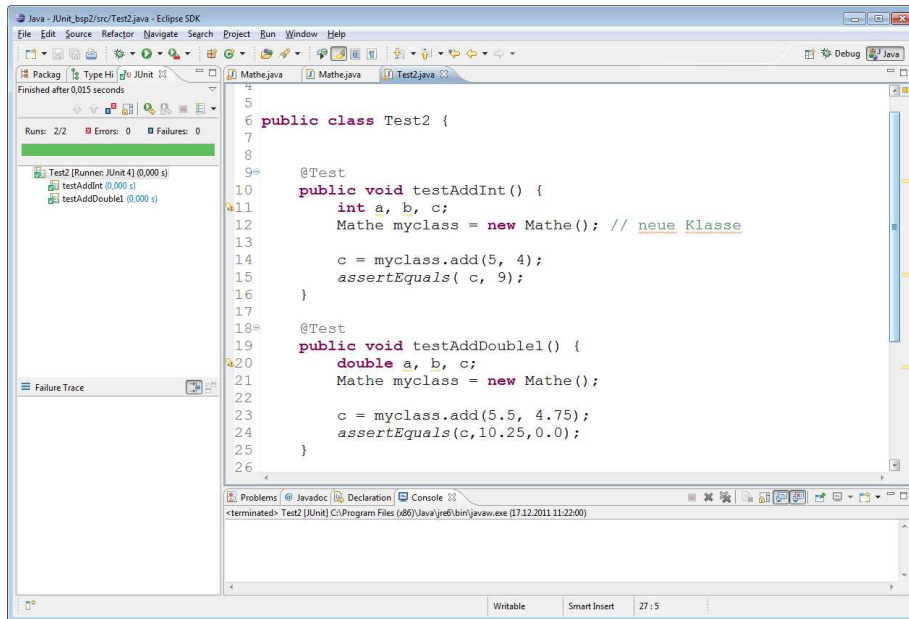
        c = myclass.add(5.5, 4.75);
        assertEquals(c,10.25,0.0);
    }

    // @Test
    public void atestAddDouble2() {
        double a, b, c;
        Mathe myclass = new Mathe();

        c = myclass.add(5.033, 4.343);
        assertEquals(c,9.376,0.0);
    }
}
```

Starten des Tests

- Wechseln ins Register "Test2.java"
- STRG+F11



Test Matrixmultiplikation

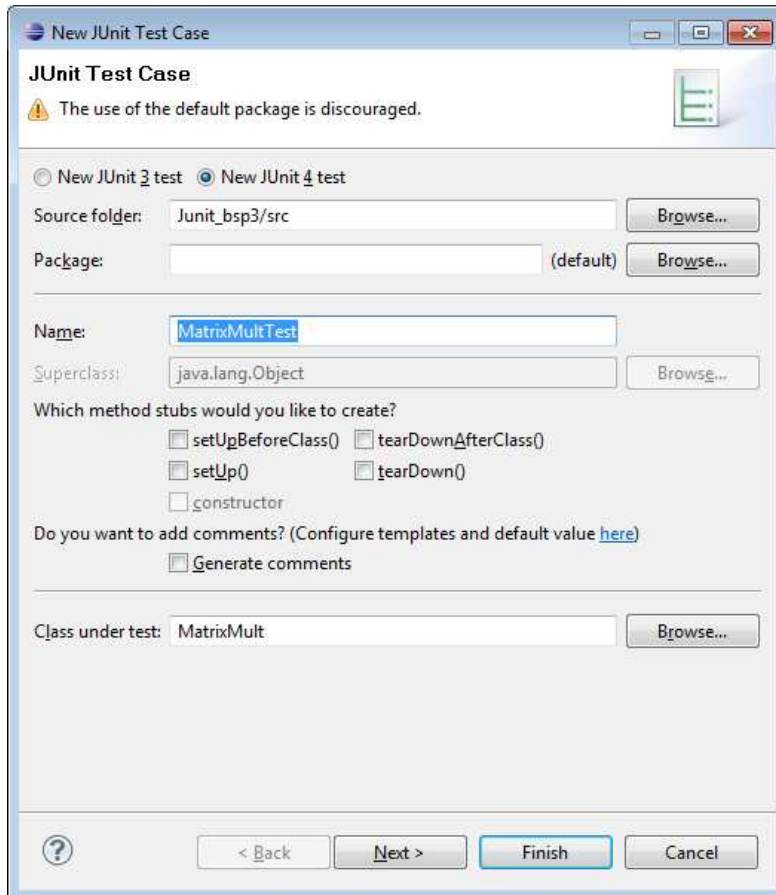
```
import java.util.Vector;

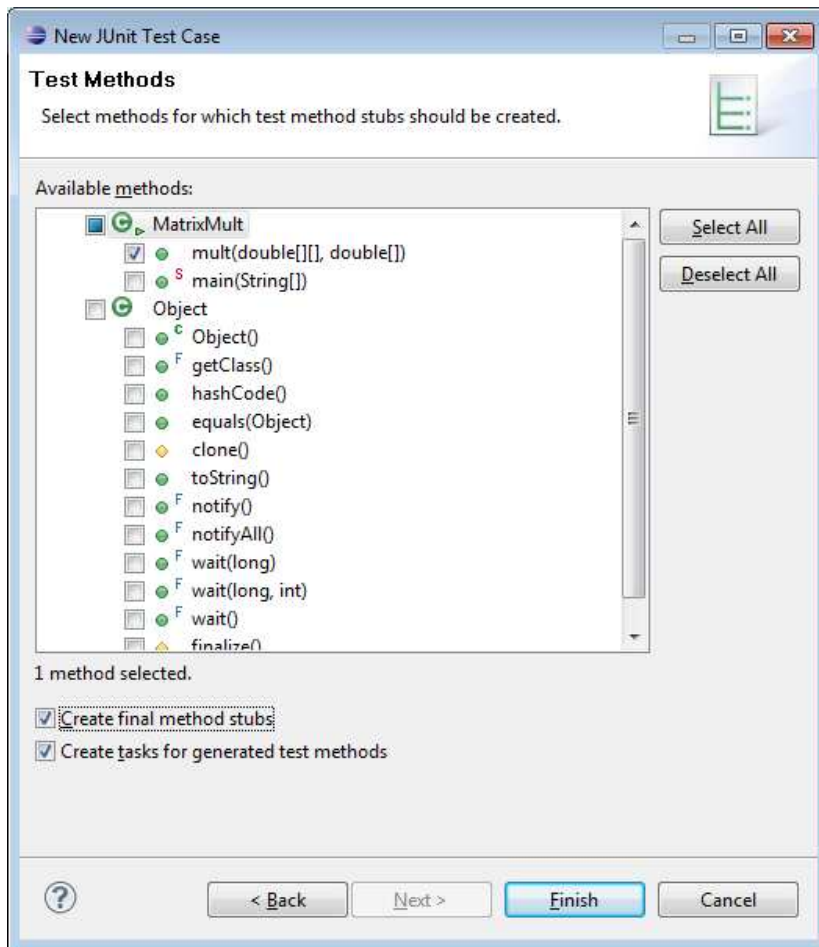
public class MatrixMult {

    /*
    Eingangsgrößen:  Matrix A   m ? n
                   Vektor B    n
    Ausgangsgrößen: Vektor C   m

    Berechnung:      C = A x B
    */
    public double [] mult(double A[][], double B[] ){
        int i,j,m, n;
        double s;
        double C[];
        m = A.length;
        n = B.length;
        C = new double[m];
        for (i=0; i<m; i++ ) {
            s = 0;
            for (j=0; j<B.length; j++) {
                s += A[i][j] * B[j];
            }
            C[i] = s;
        }
        return C;
    }

} // class MatrixMult
```





Quellcode des Tests

```
import static org.junit.Assert.*;

import org.junit.Test;

public class MatrixMultTest {

    @Test
    public final void testMult1() {
        double A[][];
        double B[];
        double C[];
        MatrixMult m;
        A = new double[3][2];
        B = new double[2];
        double[] CErg = {9.00,12,15};

        A[0][0]=1;
        A[1][0]=2;
        A[2][0]=3;
        A[0][1]=4;
        A[1][1]=5;
        A[2][1]=6;

        B[0]=1;
        B[1]=2;
        m = new MatrixMult();
        C = m.mult(A,B);
        for (int i=0; i<C.length; i++ ) {
            System.out.println("Matrix C: "+C[i] );
        }
        for (int i=0; i<C.length; i++ ) {
            assertEquals(
                "Vergleich der Vektoren:
                i"+i,C[i],CErg[i],0.00000001);
        }
        //fail("Not yet implemented"); // TODO
    }
}
```

```

@Test
public final void testMult2() {
    double A[][];
    double B[];
    double C[];
    MatrixMult m;
    A = new double[3][2];
    B = new double[2];
    double[] CErg = {11.0385,14.7048,18.3711};

    A[0][0]=1.123;
    A[1][0]=2.234;
    A[2][0]=3.345;
    A[0][1]=4.456;
    A[1][1]=5.567;
    A[2][1]=6.678;

    B[0]=1.1;
    B[1]=2.2;

    // CErg[0] = 11.0385
    /* 1.123*1.1 + 4.456*2.2 = 1^1.0385 */

    m = new MatrixMult();
    C = m.mult(A,B);
    System.out.println("\n\nTest2: " );
    for (int i=0; i<C.length; i++ ) {
        System.out.println("Matrix C: "+C[i] );
    }
    for (int i=0; i<C.length; i++ ) {
        assertEquals("Vergleich der Vektoren:
            i"+i,C[i],CErg[i],0.0);
    }
} // Test2

```

Kontotest JUnit_bsp4

```
public class Konto {

    private double _betrag;
    private String _name;

    public Konto(double betrag, String name) {
        _betrag = betrag;
        _name = name;
    }
    public void add (double betrag) {
        _betrag += betrag;
    }
    public void sub (double betrag) {
        _betrag -= betrag;
    }

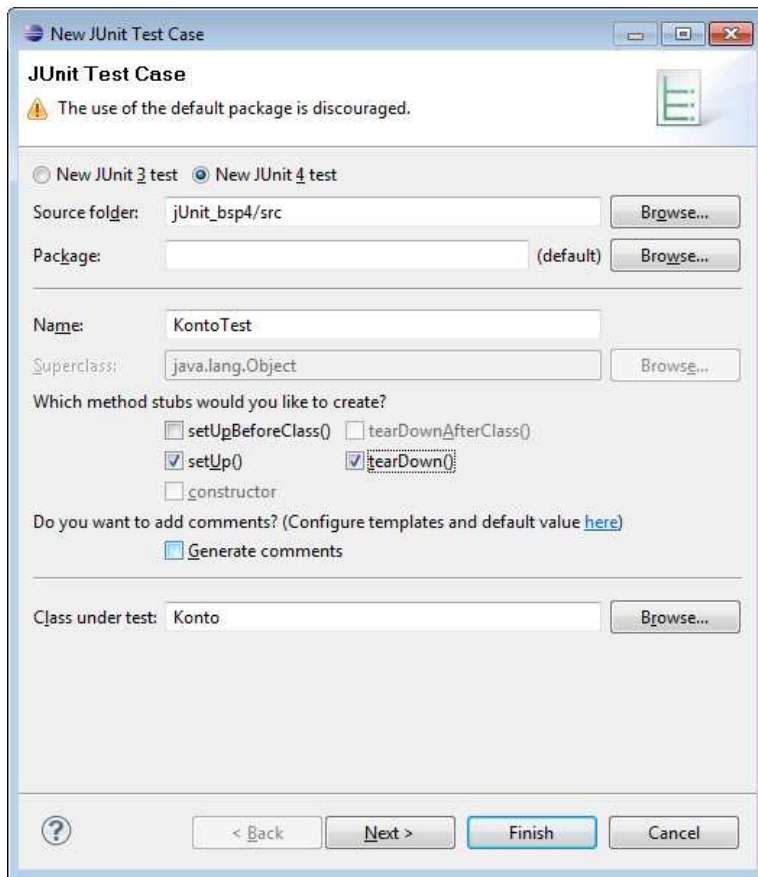
    public double getBetrag() {
        return _betrag;
    }
    public void setBetrag(double betrag) {
        _betrag = betrag;
    }

    public String getName() {
        return _name;
    }

    public String toString() {
        return new String( "[ Name: "+getName()+
            "   Betrag: "+getBetrag()+" ]");
    }

    public static void main(String[] args) {
        System.out.println("In Konto");
    }

}
```



Testquellcode:

```
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

import static org.junit.Assert.*;

public class KontoTest {

    private Konto k1;
    private Konto k2;
    private Konto k3;
    private Konto k4;

    @Test
    public void testAdd() {
        // 100 + 20 = 120
        k1.add(20);
        assertEquals(k1.getBetrag(), 120.0, 0.0001);
    }

    @Test
    public void testPrint() {
        assertEquals("[ Name: Müller   Betrag: 100.0 ]",
            k1.toString());
    }

    @Test
    public void testSub1() {
        // 100 - 20 = 80}
        k1.sub(20);
        assertEquals(k1.getBetrag(), 80.0,0.0);
    }

    @Test
    public void testSub2() {
        // 100 - 20 = 80}
        k1.sub(20);
        assertEquals(k1.getBetrag(), 80.0 , 0.0001);
    }

    @Test
    public void testNumerik() {
        // 0.1+0.1+0.1   = 0.3
        k1.setBetrag(0.0);
        k1.add(0.1);
        k1.add(0.1);
        k1.add(0.1);
        assertEquals(k1.getBetrag(),0.3 , 0.0);
    }
}
```

```

@Test
public void testSummen() {
    double s1, s2;
    s1 = k1.getBetrag() + k2.getBetrag()
        + k3.getBetrag() + k4.getBetrag();
    k1.add(20.0);
    k2.sub(20.0);

    k1.add(60.0);
    k4.sub(60.0);

    k3.add(25.0);
    k2.sub(25.0);

    s2 = k1.getBetrag() + k2.getBetrag()
        + k3.getBetrag() + k4.getBetrag();
    System.out.println("s1: "+s1);
    System.out.println("s2: "+s2);
    assertEquals(s1, s2, 0.0001);
} // testSummen

@Before
public void setUp() throws Exception {
    k1 = new Konto(100.0, "Müller");
    k2 = new Konto(200.0, "Schmidt");
    k3 = new Konto(300.0, "Meyer");
    k4 = new Konto(400.0, "Schulze");
}

@After
public void tearDown() throws Exception {
}
}

```