

## 5.5\_Refactoring Steps

```
public class MyMathTestRefactored {

    private double x;
    private double y;
    private MyMath mm;
    private Fraction f;

    @BeforeEach
    public void setup () {
        mm = new MyMath();
        x = 10d;
        y = 5d;
        f = new Fraction();
    }
}
```

Als erstes, Variablen erstellt, die für das Testen gebraucht werden und anschließend ein Setup erstellt, das vor jedem Test ausgeführt wird.

```
@Test
public void ifXIs10PlusY5_ThenReturn15() { assertEquals( expected: 15d, (double) mm.add(x,y)); }

@Test
public void ifXIs10MinusY5_ThenReturn5() { assertEquals( expected: 5d, (double) mm.sub(x,y)); }

@Test
public void ifXIs10MultipliedByY5_ThenReturn50() { assertEquals( expected: 50d, (double) mm.mul(x,y)); }

@Test
public void ifXIs10DividedByY5_ThenReturn2() { assertEquals( expected: 2d, (double) mm.div(x,y)); }
```

Die verschiedenen Tests von x1 separiert und den Code verkürzt, unnötige Variablen entfernt. Richtige Namensverteilung. Wiederkehrende Variablen als Setup angelegt

```
@Test
public void ifNumeratorIs1AndDenominatorIs1Reduce_ThenReturn1And1(){
    f = new Fraction( numerator: 1, denominator: 1);
    assertEquals(new Fraction( numerator: 1, denominator: 1),mm.reduce(f));
}

@Test
public void ifNumeratorIs10AndDenominatorIs6Reduce_ThenReturn5And3(){
    f = new Fraction( numerator: 10, denominator: 6);
    assertEquals(new Fraction( numerator: 5, denominator: 3),mm.reduce(f));
}

@Test
public void ifNumeratorIs10AndDenominatorIs5Reduce_ThenReturn2And1(){
    f = new Fraction( numerator: 10, denominator: 5);
    assertEquals(new Fraction( numerator: 2, denominator: 1),mm.reduce(f));
}

@Test
public void ifNumeratorIs10AndDenominatorIs5ToDouble_ThenReturn2(){
    f = new Fraction( numerator: 10, denominator: 5);
    assertEquals( expected: 2d, (double) mm.toDouble(f));
}

@Test
public void ifNumeratorIs10AndDenominatorIs4ToDouble_ThenReturn2_5(){
    f = new Fraction( numerator: 10, denominator: 4);
    assertEquals( expected: 2.5d, (double) mm.toDouble(f));
}
```

Bei x2 die Tests vereinfacht und verkürzt, unnötige Variablen entfernt, wiederkehrende Variablen im Setup angelegt, assertEquals umgeändert um Variablen loszuwerden. Anstelle von getNumerator und getDenominator es gleich als Fraction dargestellt. Nicht nur bei Actual, sondern auch bei Expected. anstelle von neue Double Elemente die Zahl gleich als double darstellen.