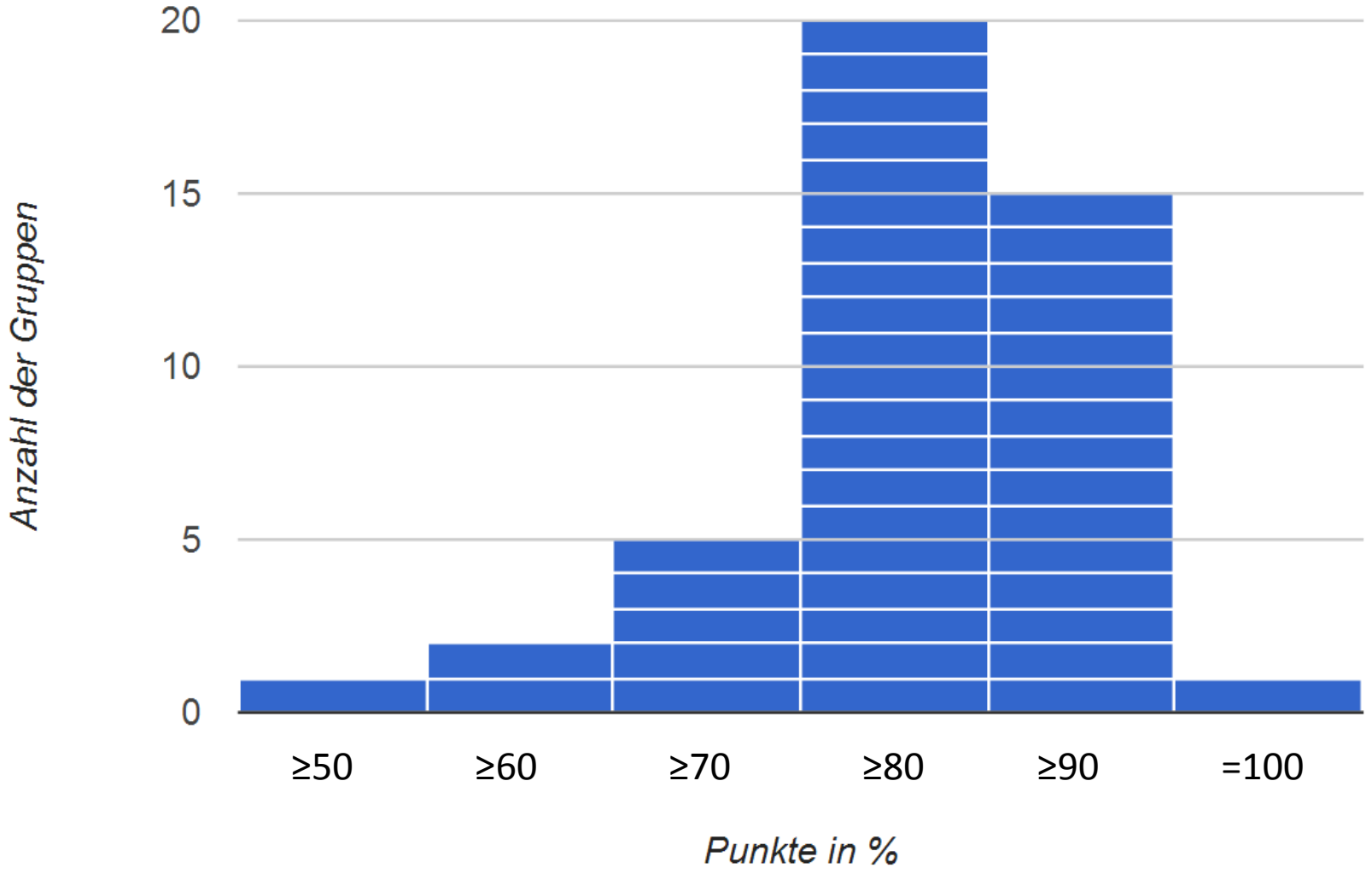
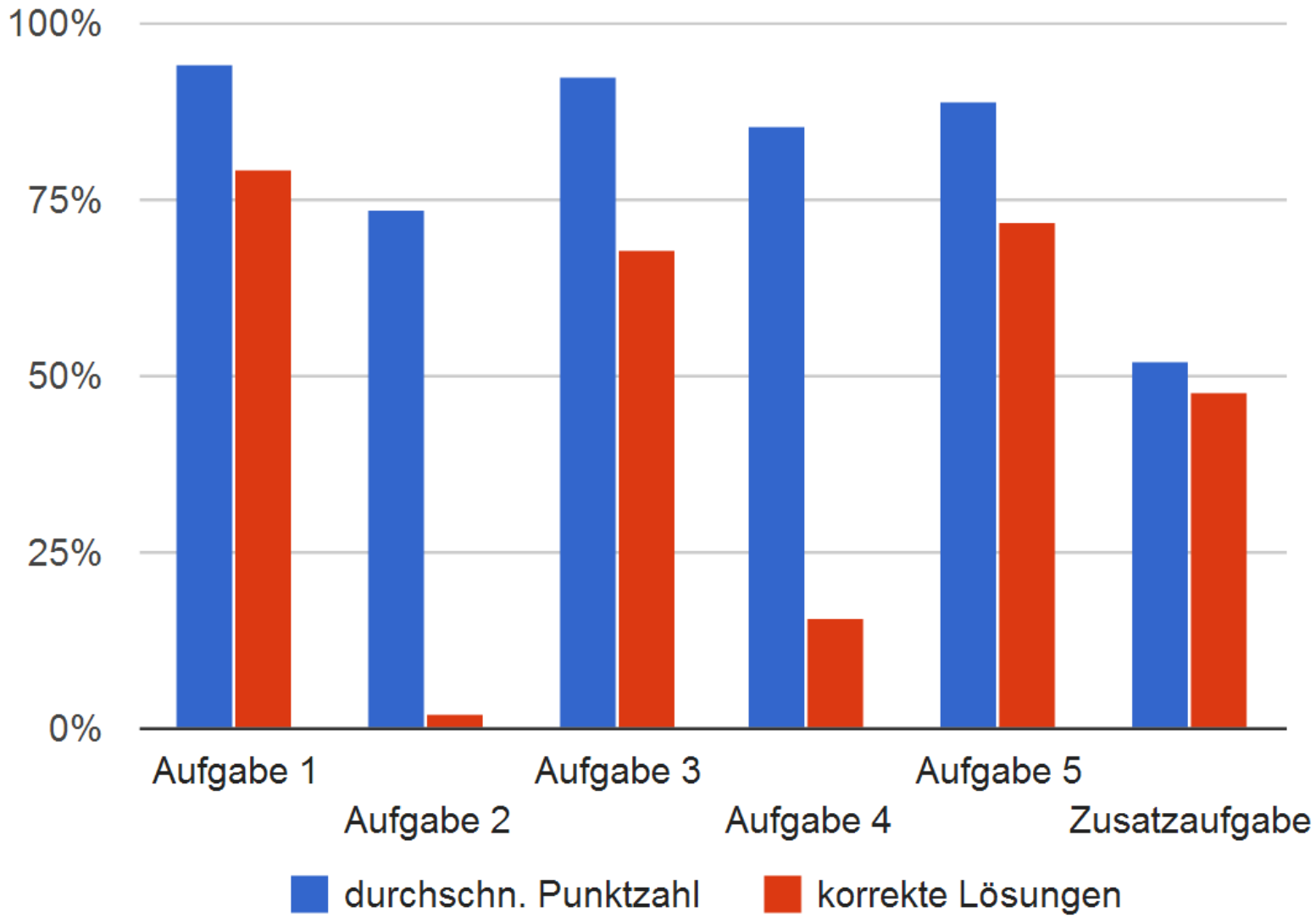
Übungsblatt 1 - Auswertung

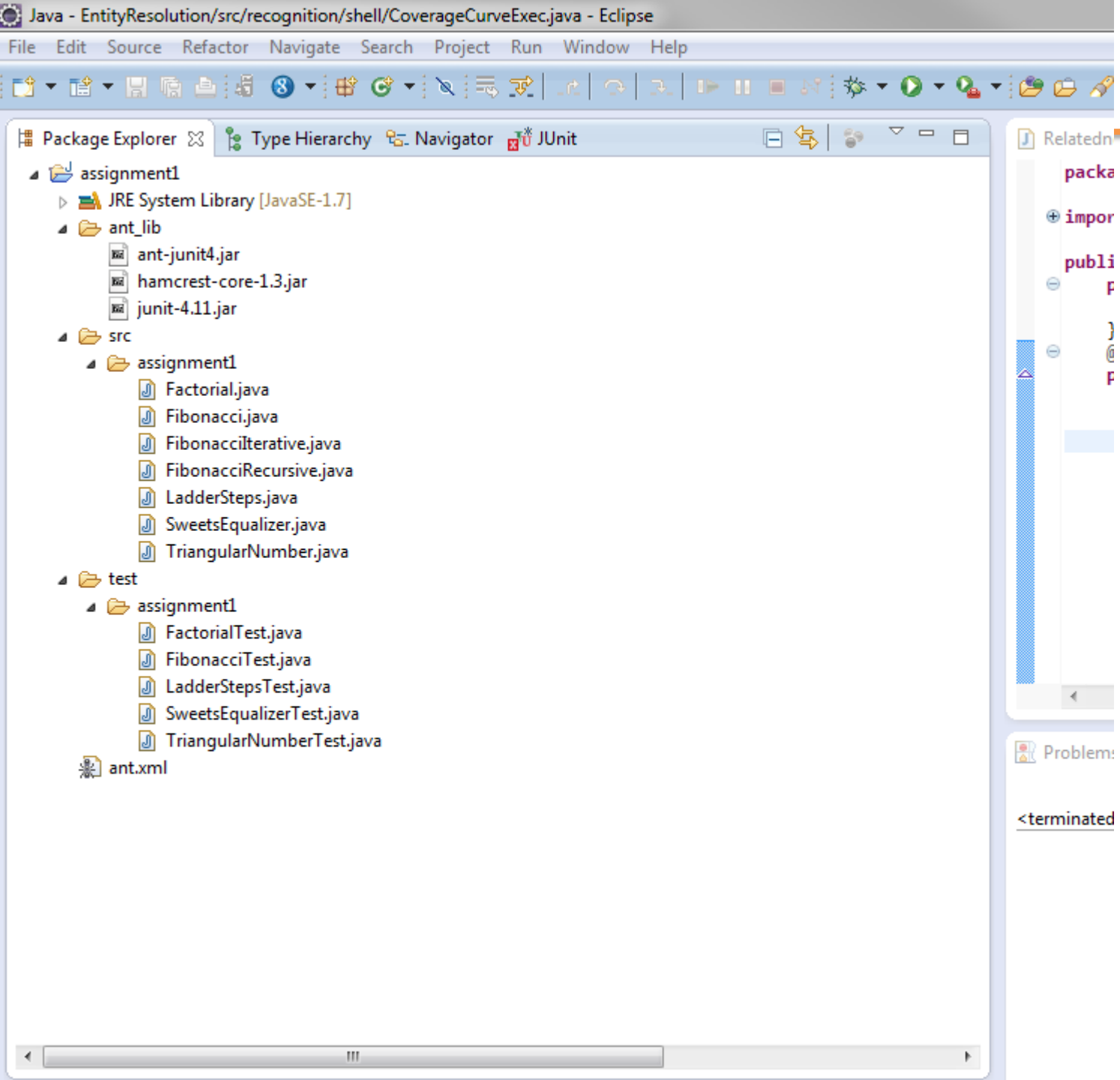
Punktverteilung (Gesamt)

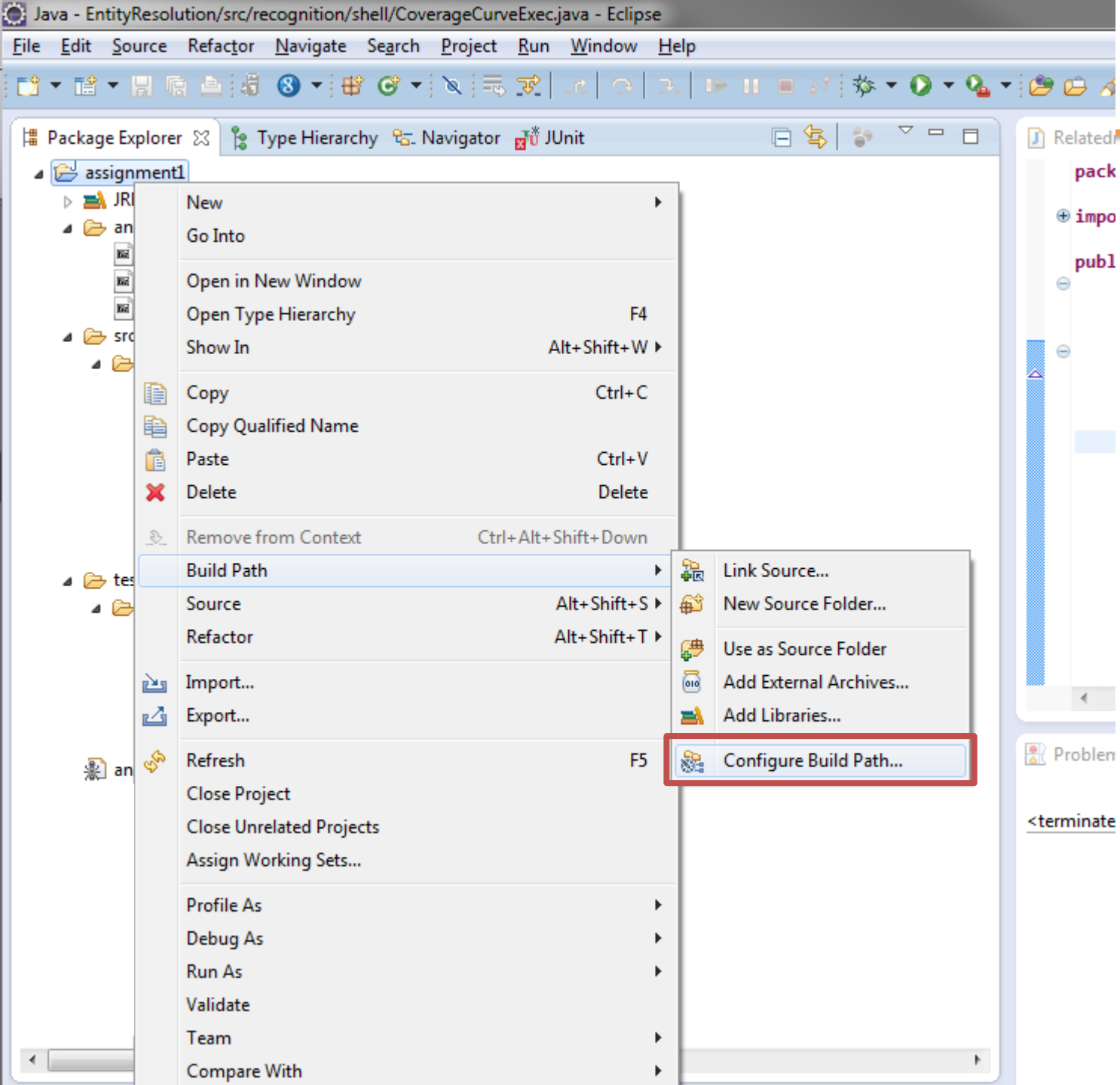


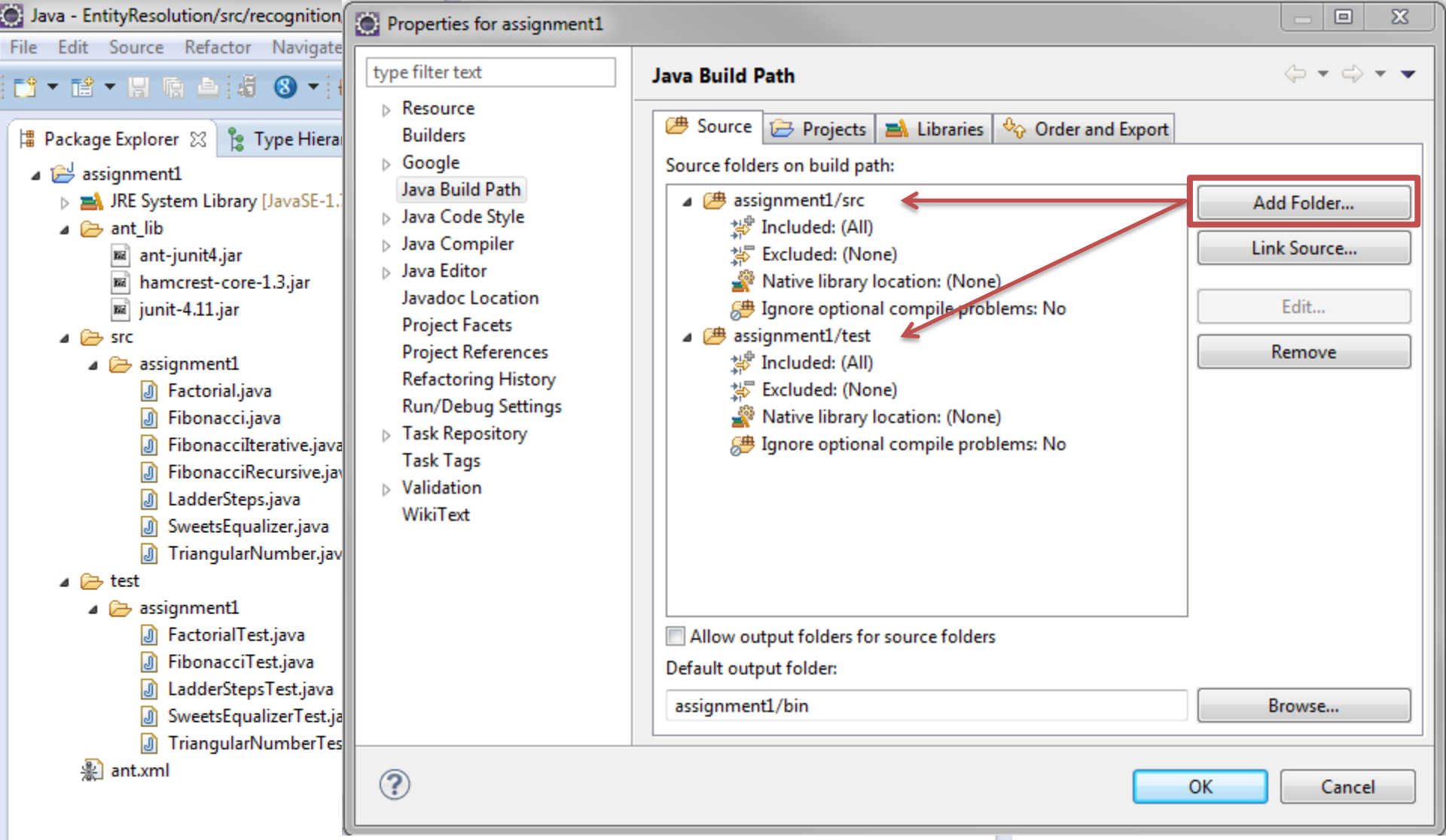
Ergebnisse pro Aufgabe

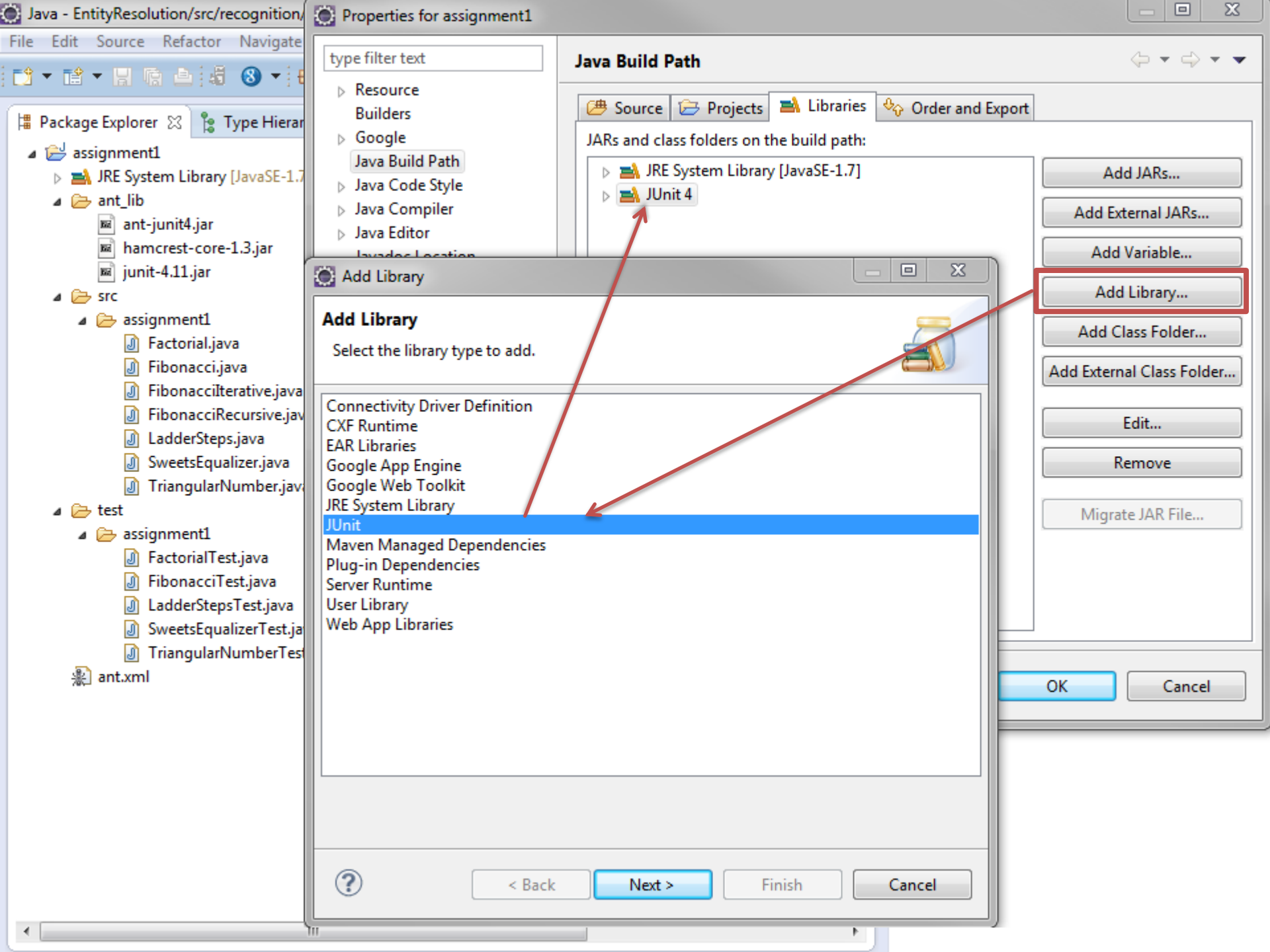


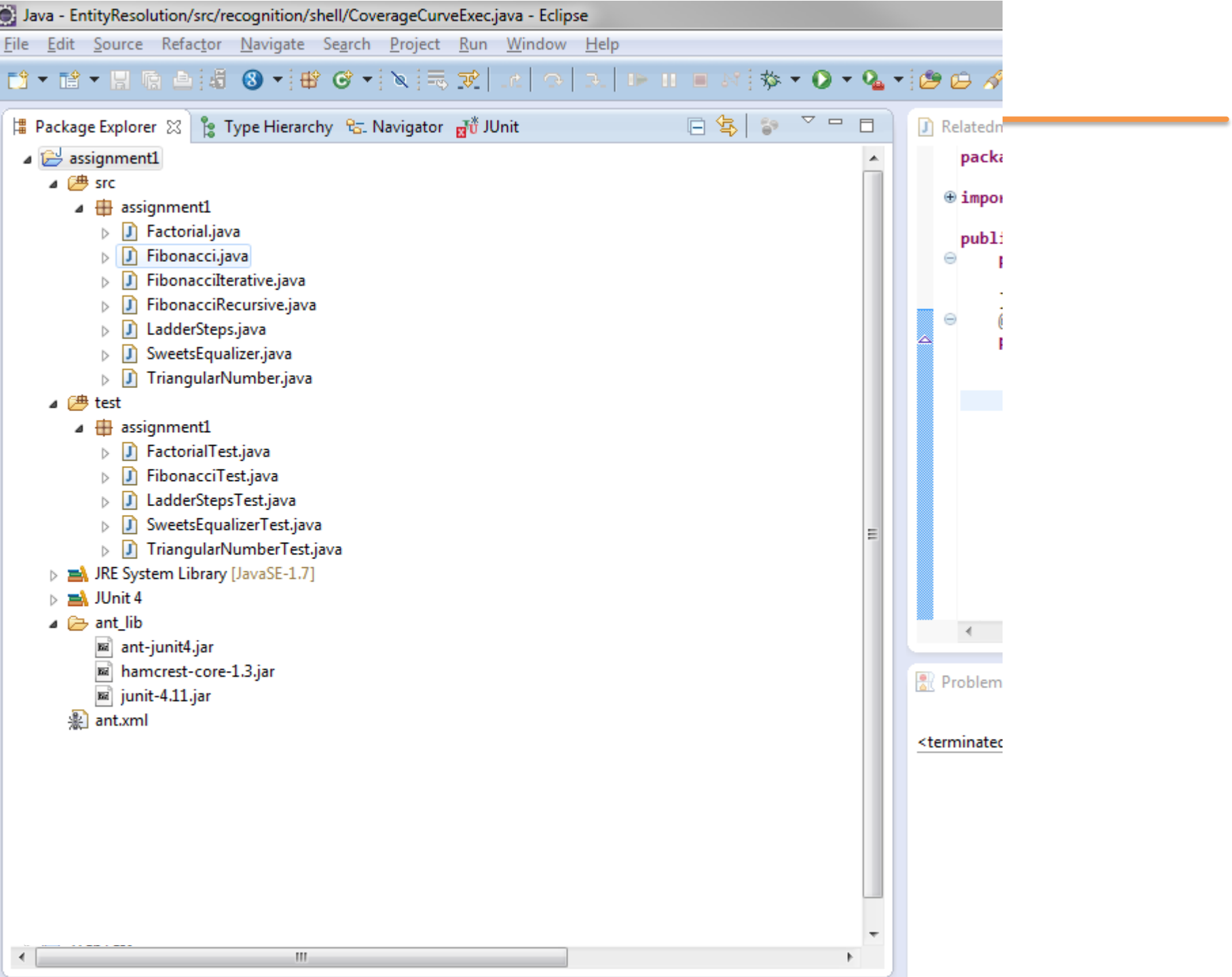
Projektstruktur (Eclipse)


Java - EntityResolution/src/recognition/shell/CoverageCurveExec.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Type Hierarchy
Navigator
JUnit
assignment1
JRE System Library [JavaSE-1.7]
ant_lib
ant-junit4.jar
hamcrest-core-1.3.jar
junit-4.11.jar
src
assignment1
Factorial.java
Fibonacci.java
FibonacciIterative.java
FibonacciRecursive.java
LadderSteps.java
SweetsEqualizer.java
TriangularNumber.java
test
assignment1
FactorialTest.java
FibonacciTest.java
LadderStepsTest.java
SweetsEqualizerTest.java
TriangularNumberTest.java
ant.xml
Problems
<terminated


Java - EntityResolution/src/recognition/shell/CoverageCurveExec.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Type Hierarchy
Navigator
JUnit
assignment1
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Profile As
Debug As
Run As
Validate
Team
Compare With
Link Source...
New Source Folder...
Use as Source Folder
Add External Archives...
Add Libraries...
Configure Build Path...
pack
impo
publ
<terminate

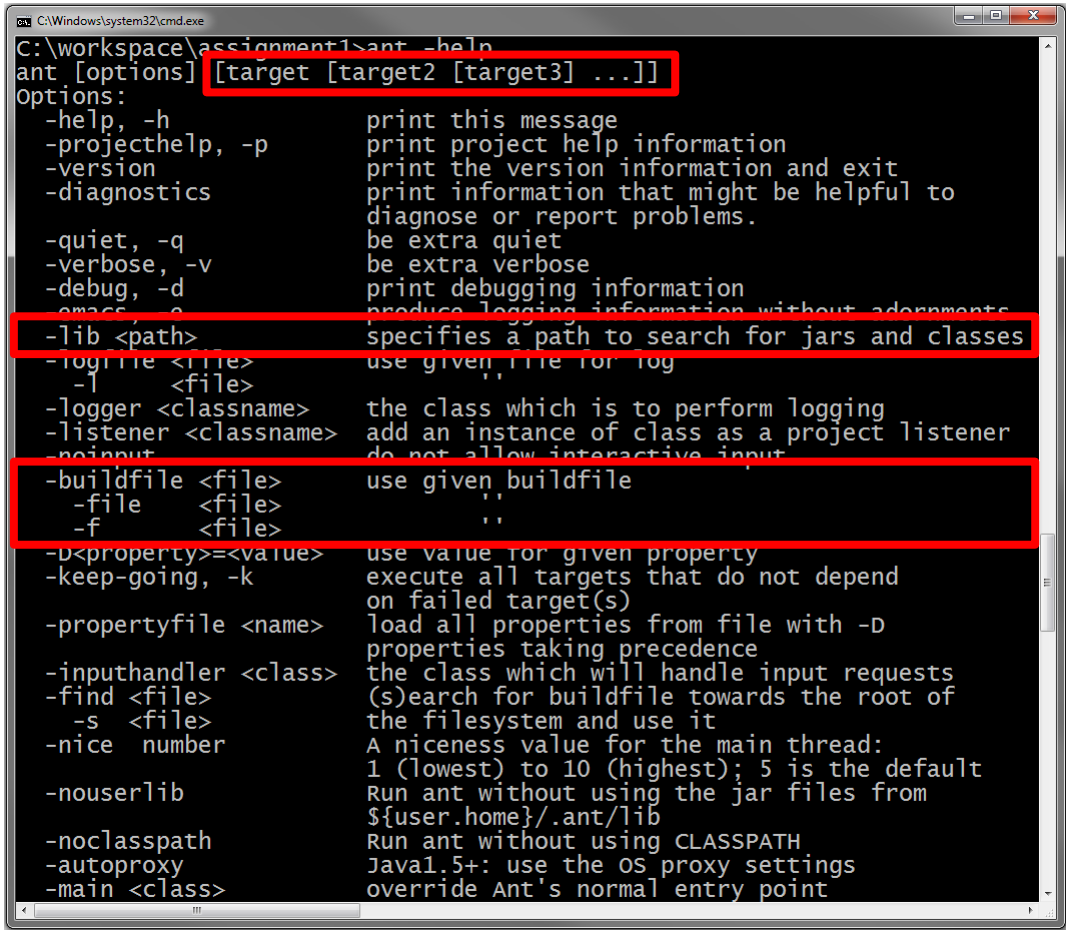

Java - EntityResolution/src/recognition
File Edit Source Refactor Navigate
Package Explorer
Type Hiera
assignment1
JRE System Library [JavaSE-1.
ant_lib
ant-junit4.jar
hamcrest-core-1.3.jar
junit-4.11.jar
src
assignment1
Factorial.java
Fibonacci.java
FibonacciIterative.java
FibonacciRecursive.ja
LadderSteps.java
SweetsEqualizer.java
TriangularNumber.jav
test
assignment1
FactorialTest.java
FibonacciTest.java
LadderStepsTest.java
SweetsEqualizerTest.ja
TriangularNumberTes
ant.xml
Properties for assignment1
type filter text
Resource
Builders
Google
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
Project Facets
Project References
Refactoring History
Run/Debug Settings
Task Repository
Task Tags
Validation
WikiText
Java Build Path
Source
Projects
Libraries
Order and Export
Source folders on build path:
assignment1/src
Included: (All)
Excluded: (None)
Native library location: (None)
Ignore optional compile problems: No
assignment1/test
Included: (All)
Excluded: (None)
Native library location: (None)
Ignore optional compile problems: No
Add Folder...
Link Source...
Edit...
Remove
Allow output folders for source folders
Default output folder:
assignment1/bin
Browse...
OK
Cancel


Java - EntityResolution/src/recognition/
File Edit Source Refactor Navigate
Package Explorer
Type Hierar
assignment1
JRE System Library [JavaSE-1.7
ant_lib
ant-junit4.jar
hamcrest-core-1.3.jar
junit-4.11.jar
src
assignment1
Factorial.java
Fibonacci.java
FibonacciIterative.java
FibonacciRecursive.jav
LadderSteps.java
SweetsEqualizer.java
TriangularNumber.jav
test
assignment1
FactorialTest.java
FibonacciTest.java
LadderStepsTest.java
SweetsEqualizerTest.ja
TriangularNumberTest
ant.xml
Properties for assignment1
type filter text
Resource
Builders
Google
Java Build Path
Java Code Style
Java Compiler
Java Editor
Java Build Path
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
JRE System Library [JavaSE-1.7]
JUnit 4
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
OK
Cancel
Add Library
Add Library
Select the library type to add.
Connectivity Driver Definition
CXF Runtime
EAR Libraries
Google App Engine
Google Web Toolkit
JRE System Library
JUnit
Maven Managed Dependencies
Plug-in Dependencies
Server Runtime
User Library
Web App Libraries
< Back
Next >
Finish
Cancel


Java - EntityResolution/src/recognition/shell/CoverageCurveExec.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Type Hierarchy
Navigator
JUnit
assignment1
src
assignment1
Factorial.java
Fibonacci.java
FibonacciIterative.java
FibonacciRecursive.java
LadderSteps.java
SweetsEqualizer.java
TriangularNumber.java
test
assignment1
FactorialTest.java
FibonacciTest.java
LadderStepsTest.java
SweetsEqualizerTest.java
TriangularNumberTest.java
JRE System Library [JavaSE-1.7]
JUnit 4
ant_lib
ant-junit4.jar
hamcrest-core-1.3.jar
junit-4.11.jar
ant.xml

Aufgabe 1 - Ant-File

- ANT zum OS-Suchpfad (PATH) hinzufügen
- Kommandozeilen-parameter verstehen
- Ordner mit den jar-Dateien erstellen (z.B. ./ant_lib/)



build-file

ant-libs

ant-target: junit

C:\Windows\system32\cmd.exe

```
C:\workspace\assignment1>ant -f ant.xml -lib ./ant_lib/ junit
Buildfile: C:\workspace\assignment1\ant.xml

clean:
   [delete] Deleting directory C:\workspace\assignment1\build

compile:
    [mkdir] Created dir: C:\workspace\assignment1\build
    [javac] Compiling 7 source files to C:\workspace\assignment1\build
    [javac] Compiling 5 source files to C:\workspace\assignment1\build

junit:
    [junit] Running assignment1.FactorialTest
    [junit] Tests run: 3, Failures: 0, Errors: 0, Time elapsed: 0.008 sec
    [junit] Running assignment1.FibonacciTest
    [junit] Tests run: 6, Failures: 0, Errors: 0, Time elapsed: 54.49 sec
    [junit] Running assignment1.LadderStepsTest
    [junit] Tests run: 3, Failures: 0, Errors: 0, Time elapsed: 0.002 sec
    [junit] Running assignment1.SweetsEqualizerTest
    [junit] Tests run: 6, Failures: 0, Errors: 0, Time elapsed: 0.002 sec
    [junit] Running assignment1.TriangularNumberTest
    [junit] Tests run: 2, Failures: 0, Errors: 0, Time elapsed: 0.001 sec

BUILD SUCCESSFUL
Total time: 55 seconds
```

```
<project default="junit">

  <target name="clean">
    <delete dir="build"/>
  warning: 'includeantruntime' was not set, defaulting to
  build.sysclasspath=last; set to false for repeatable builds
  <target name="compile">
    <mkdir dir="build"/>
    <javac srcdir="src" destdir="build"/>
    <javac srcdir="test" destdir="build"/>
  </target>

  <target name="junit" depends="clean,compile">
    <junit printsummary="yes">
      <classpath path="build"/>
      <batchtest>
        <fileset dir="test" includes="**/*Test.java"/>
      </batchtest>
    </junit>
  </target>

</project>
```

```
1  <project default="junit">
2
3    <target name="clean">
4      <delete dir="build"/>
5    </target>
6
7    <target name="compile">
8      <mkdir dir="build"/>
9      <javac
10       srcdir="src" destdir="build"
11       includeantruntime="false"/>
12     <javac
13       srcdir="test" destdir="build"
14       includeantruntime="true"/>
15   </target>
16
17   <target name="junit" depends="clean,compile">
18     <junit printsummary="yes">
19       <classpath path="build"/>
20       <batchtest>
21         <fileset dir="test" includes="**/*Test.java"/>
22       </batchtest>
23     </junit>
24   </target>
```

Aufgabe 2 - Dreieckszahl

- Quasi trivial lösbar durch
  - Iteration $\Delta(n) = \sum_{i=1}^{n} i$ oder
  - Gaußsche Summenformel $\Delta(n) = \frac{n \cdot (n+1)}{2}$

- Ausnahme: $n \geq 2^{16}$

  → **int** overflow ($2^{31} \leq i \leq 2^{31}-1$)

- Lösung: Rechnen mit **long**
- hexadezimal: Long.toHexString(n)

Aufgabe 3 - Fakultät

- Lösung durch Iteration $n! = \prod_{i=1}^{n} i$
- Problem
  - **int** overflow bei *n ≥ 13*
  - **long** overflow bei *n ≥ 21*
- Lösung
  - Rechnen mit **BigInteger**:

```
1 BigInteger prod = BigInteger.ONE;
2 prod = prod.multiply(BigInteger.valueOf(2));
3 //...
4 return prod.toString(16);
```

Aufgabe 4 - Fibonacci (naiv-rekursiv)

- Lösung: straightforward
- Problem:
  - naive rekursive Definition führt zu einer exponeniellen Anzahl von Funktionsaufrufen:

$$
\begin{aligned}
fib(n) &= \overbrace{fib(n-1)}^{} + \overbrace{fib(n-2)}^{} \\
&= \overbrace{fib(n-2)+fib(n-3)}^{} + \overbrace{fib(n-3)+fib(n-4)}^{} \\
&= \overbrace{fib(n-3)+fib(n-4)}^{} + \overbrace{}^{} \\
&\quad fib(n-4)+fib(n-5) + \\
&\qquad\qquad fib(n-4)+fib(n-5) + \\
&\qquad\qquad fib(n-5)+fib(n-6) \\
&= \ldots
\end{aligned}
$$

Aufgabe 2 – Fibonacci (iterativ)

- Idee: Anwendung der Endrekursion
  - Beginne einmalig am Ende der Rekursion
    fib(2) (letzer Wert) und *fib(1)* (vorletzter Wert)
  - Summiere n-2 mal den letzten / vorletzten Wert

```
1  //...
2  long last = 1, nextToLast = 1;
3  for(long l=2;l<n;l++) {
4     final long curr = last + nextToLast;
5     nextToLast = last;
6     last = curr;
7  }
8  return last;
9  }
```

Aufgabe 5 - Leiteraufstieg

- Idee
  - Anzahl der Möglichkeiten für *n* Sprossen ist identisch zur Summe der Möglichkeiten für *n-1* und *n-2* Sprossen
  - Bsp. :

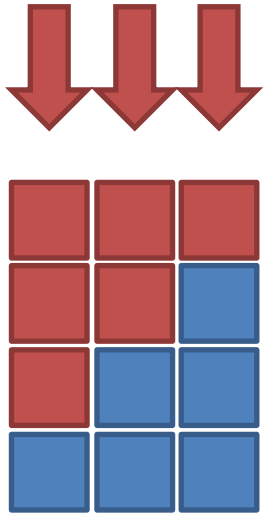
    comb(n=4) = comb(n=3) + comb(n=2)

    comb(n=4) = comb(n=2) + comb(n=1) + comb(n=2)

    comb(n=4) = 2 + 1 + 2 = 5
- Ähnlichkeit zu Fibonacci
- Zur Terminierung des Algorithmus für n=90

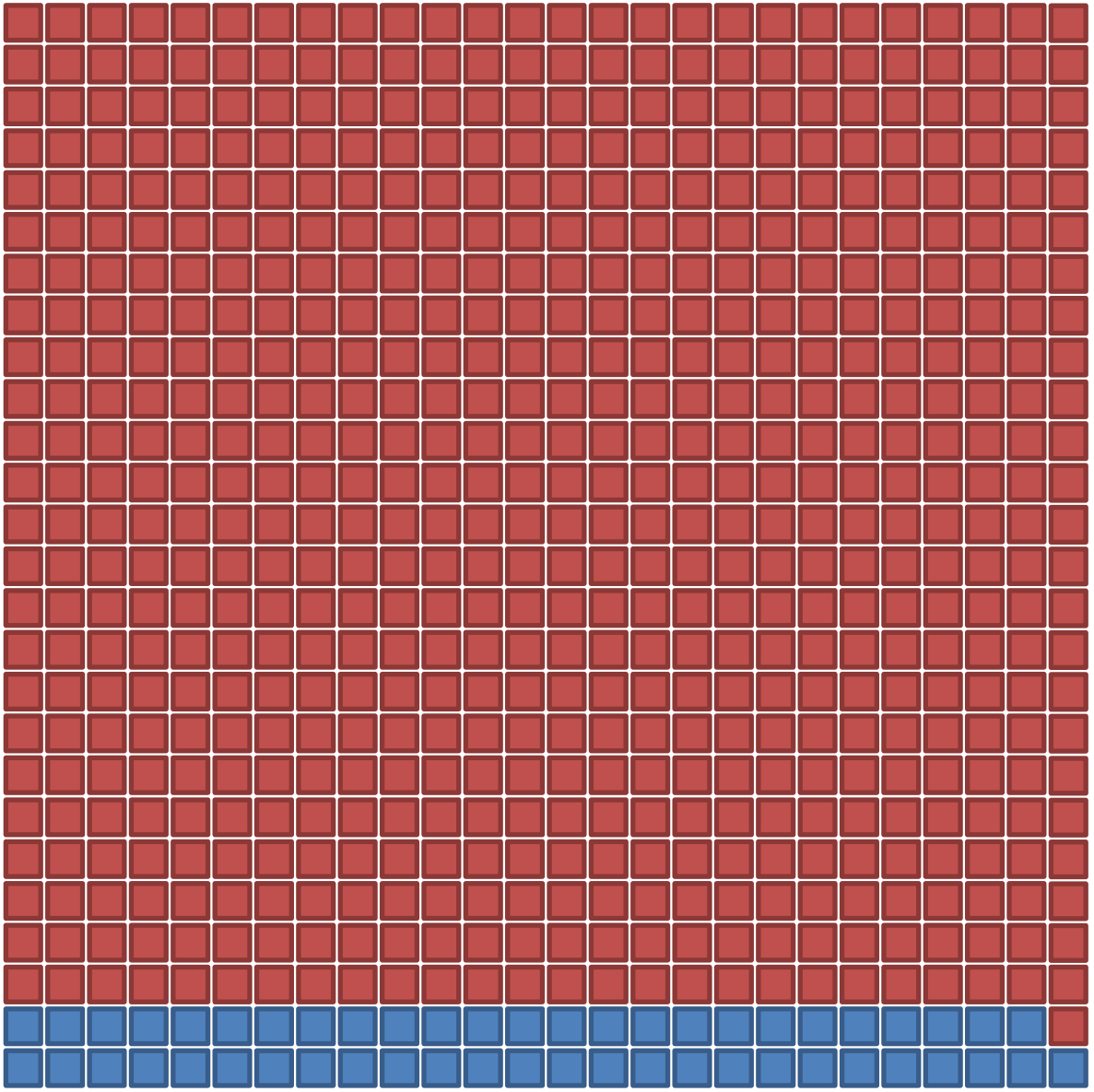
  → Iterative Berechnung

Zusatzaufgabe - Halloween



Zusatzaufgabe – Halloween: Problem

- Simulation ist teuer!
- Wertebereiche:
  - $1 \leq N < 200$
  - $0 \leq s[i] \leq 50\,000$ $(\forall i: 0 \leq i \leq N)$



Zusatzaufgabe – Halloween: Lösung

```
public static int minOperationCount(int[] s) {
    int min = 50001;
    int sum = 0;

    for (int i = 0; i < s.length; i++) {
        min = Math.min(min, s[i]);
        sum += s[i];
    }
    return sum - s.length * min;
}
```