

Aufgabe 1

Aufgabe 1 a)

Die Klasse nutzt folgende 13 Klassen:

```
java.io.BufferedInputStream;  
java.io.ByteArrayInputStream;  
java.io.ByteArrayOutputStream;  
java.io.IOException;  
java.io.InputStream;  
java.util.jar.JarEntry;  
java.util.jar.JarFile;  
java.util.jar.Manifest;  
java.util.zip.ZipEntry;  
java.util.zip.ZipException;  
java.util.zip.ZipInputStream;  
sun.misc.JarIndex;  
sun.security.util.ManifestEntryVerifier;
```

Und von den folgenden 7 Klassen wird ein Objekt erzeugt:

```
ByteArrayOutputStream  
ByteArrayInputStream  
BufferedInputStream  
doneEntryVerifier  
JarVerifier  
JarEntry  
Manifest
```

Aufgabe 1 b)

Per Definition ist eine Klasse von einer anderen abhängig, wenn sie sie direkt oder indirekt benötigt. Das ist hier der Fall. Würde IOException nicht existieren, so würde das Programm nicht compilieren.

Aufgabe 2

Aufgabe 2 a

$M(\text{LinkedList}) = 10$

$F(\text{LinkedList}) = 3$

$MF(\text{head}) = 4$

$MF(\text{size}) = 8$

$MF(\text{tail}) = 6$

$LCOM(\text{LinkedList}) = 0.4$

$M(\text{Node}) = 7$

$F(\text{Node}) = 2$

$MF(\text{element}) = 4$

$MF(\text{next}) = 4$

$LCOM(\text{Node}) = 0.429$

Aufgabe 2 b

In LinkedList sind die problematischen Methoden bezüglich der Kohäsion:

`Remove(Node<V>),`

`getFirst(),`

`getLast(),`

`getSize(),`

Da diese Methoden von den drei Instanzfeldern jeweils nur eines verwenden.

In Node sind die problematischen Methoden bezüglich der Kohäsion:

getElement(),

getNext(),

setElement(),

setNext(),

welche jeweils nur eine der beiden Instanzfelder verwenden und

Node()

Welche keines verwendet.

Aufgabe 2 c

In LinkedList wäre die Alternative um Kohäsion zu erhöhen eine eigene jeweils eigene Klassen für die Methoden welche size und tail bzw. size und head verwenden anzulegen.

Dies würde aber die logische Kohäsion, sowie die Übersichtlichkeit stark beeinträchtigen.

In Node wäre die Alternative eigene Klassen anzulegen, welche nur die jeweiligen getter und setter als Methoden besitzen. Dies ist offensichtlich keine sinnvolle Designentscheidung.

Aufgabe 3

Aufgabe 1 c)

Kopplung von A ist größer als die von A1 und A2, da A von zwei Klassen abhängig ist während A1 und A2 jeweils nur zu einer Klasse abhängig sind.

Die Kohäsion hängt davon ab, wie Klassen B und C von A verwendet werden,

gibt es in A ausschließlich Methoden die jeweils nur B oder C verwenden

und diese sind nach der Aufteilung entsprechend in A1 bzw. A2 zu finden,

so hat sich die Kohäsion von A1 bzw. A2 gegenüber A verbessert, da man

offensichtlich zwei verschiedene Aufgaben der Klasse A problemlos

aufteilen kann. Andernfalls ist Code-Duplikation zum Aufspalten unvermeidlich und die Kohäsion verschlechtert sich. Die Kohäsion, wie auch die Kopplung von B und C werden durch die Aufteilung nicht beeinflusst.