



Software Engineering im Wintersemester 2021/2022

Prof. Dr. Martin Leucker, Malte Schmitz, Stefan Benox, Julian Schulz, Benedikt Stepanek, Friederike Weilbeer, Tom Wetterich

Übungszettel 7 (Lösungsvorschlag)

07.12.2021

Abgabe bis Donnerstag, 9. Dezember um 23:59 Uhr online im Moodle.

Aufgabe 7.1: Netzplan

4 Punkte, mittel

Gegeben ist folgendes Software-Projekt zur Entwicklung eines Routing-Systems im Client-Server-Betrieb. Die Vorgangsdauer wird in Tagen angegeben und die Modellierung soll mit Tag 1 beginnen.

- Zum Projektstart (Tag 1) wird zuerst eine gemeinsame *Schnittstelle* für die Kommunikation zwischen Client und Server festgelegt (1 Tag).
- Nachdem die Schnittstelle definiert wurde, müssen sowohl Client als auch Server zuerst einmal entworfen werden. Der *Client-Entwurf* braucht 3 Tage, der davon unabhängige *Server-Entwurf* braucht 2 Tage.
- Wenn beide Entwürfe abgeschlossen wurden, dann kann das Test-Team erste *Testfälle* erstellen (5 Tage).
- Parallel zu den Testfällen können nach den jeweiligen Entwürfen der Client und der Server implementiert werden. Der *Client* braucht 7 Tage. Die Implementierung des Servers kann aufgeteilt werden in die Implementierung der *Datenstrukturen* (3 Tage) und die Implementierung der *Algorithmen* (5 Tage).

- Wenn die Implementierung des Clients und der Entwurf der Testfälle abgeschlossen wurde, dann wird die *Usability* getestet (2 Tage).
- Wenn die Implementierung der beiden Server-Komponenten und der Entwurf der Testfälle abgeschlossen wurde, dann wird die *Performanz* ausführlich analysiert (5 Tage).
- Zum *Abschluss* des Projekts werden die Ergebnisse aufbereitet und dem Kunden vorgestellt (2 Tage).

Hinweis. Wir betrachten in dieser Aufgabe nur Vorgangsknoten-Netzpläne. Es gibt auch noch andere Arten, zum Beispiel Ereignisknoten-Netzpläne und Vorgangspfeil-Netzpläne, die hier aber nicht behandelt werden.

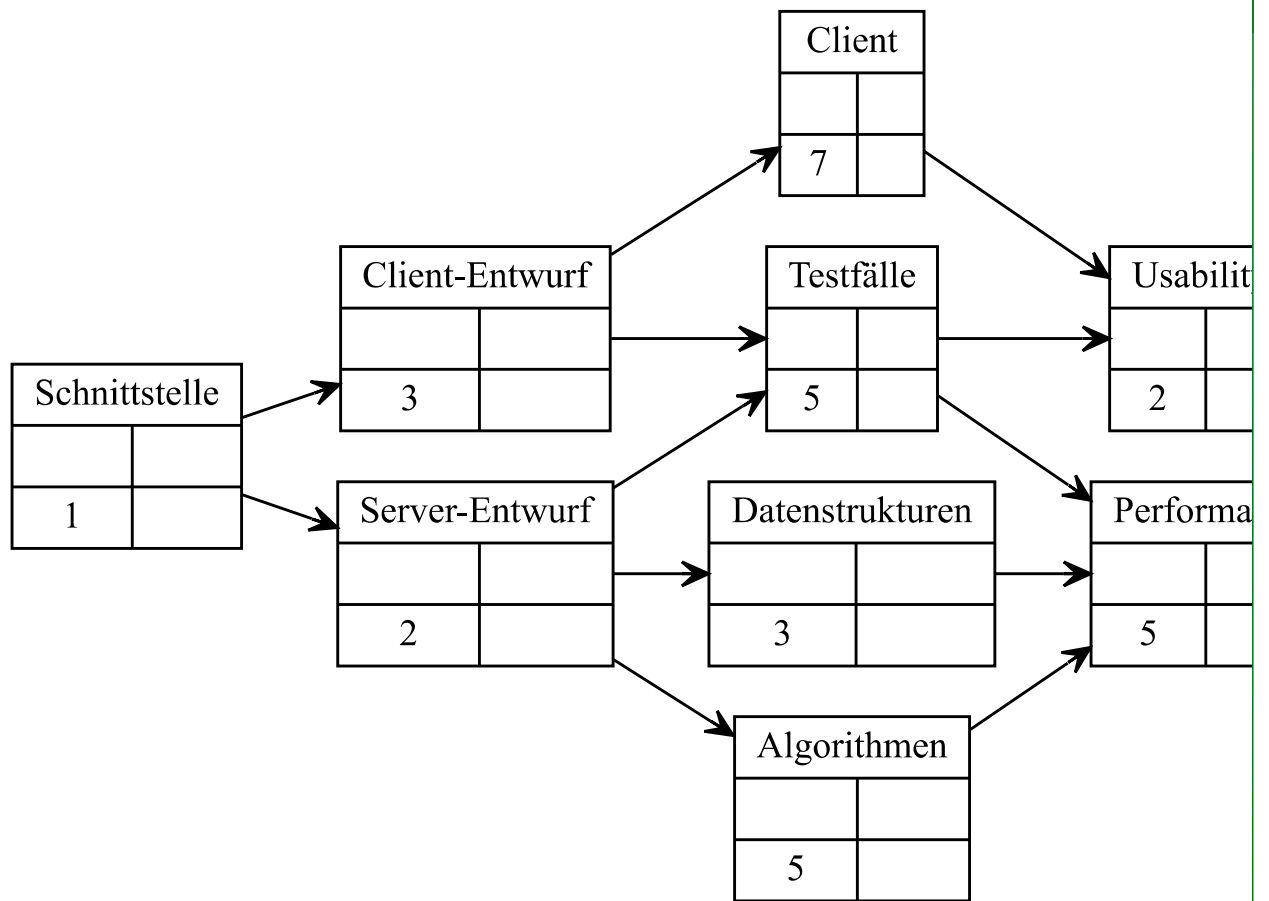
Die Vorgangsknoten sollen folgende Aufteilung haben:

Name des Vorgangs	
Min. Start	Max. Start
Dauer	Puffer

► Quelltext des Diagramms

1. Stellen Sie die gegebenen Aufgaben in einem Netzplan dar. Die (gerichteten) Kanten entsprechenden dabei den Abhängigkeiten. (2 Punkte)

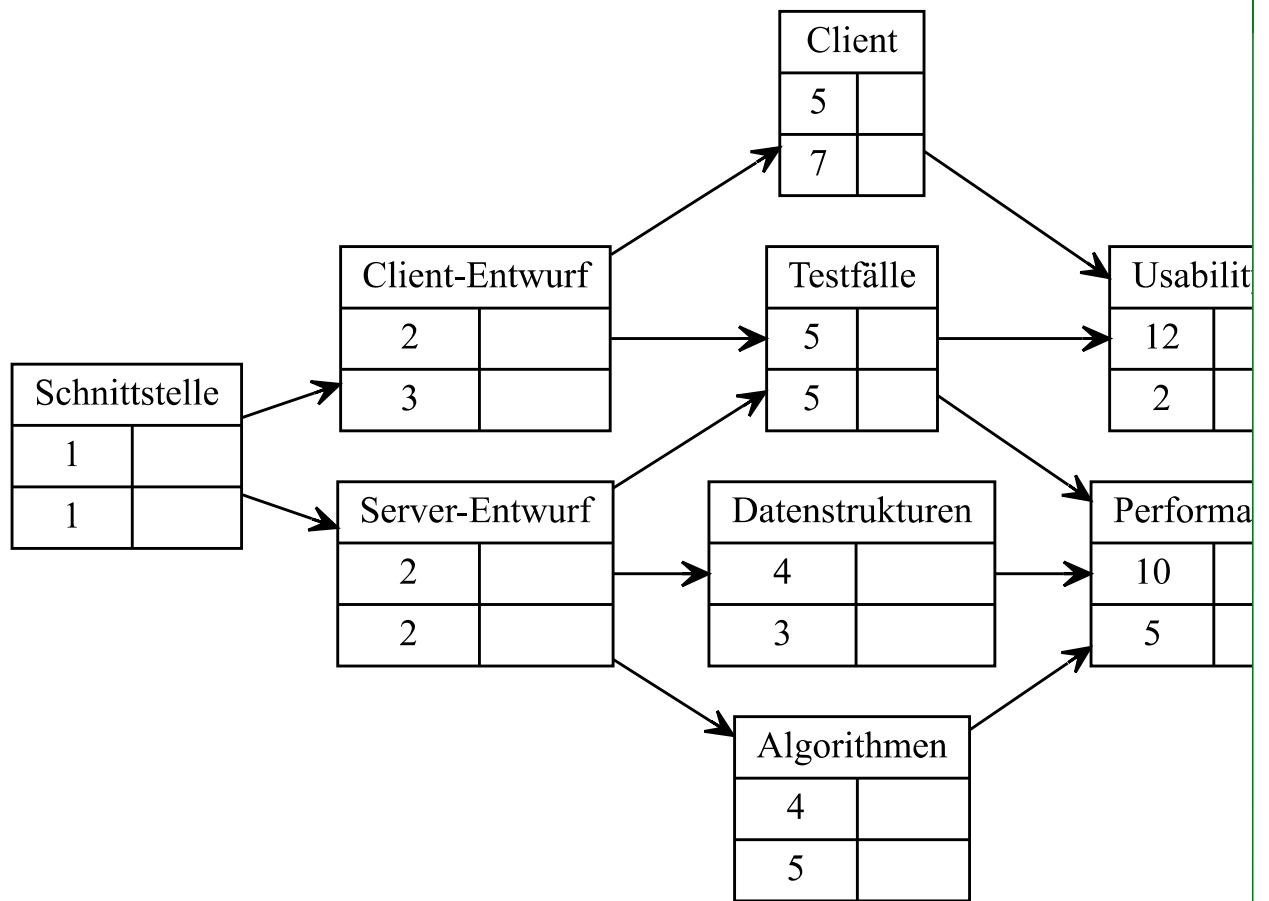
▼ Lösungsvorschlag



► Quelltext des Diagramms

2. Rechnen Sie vorwärts: Berechnen Sie, wann die Vorgänge frühestens begonnen werden können und annotieren Sie diese Werte im Netzplan. (0,5 Punkte)

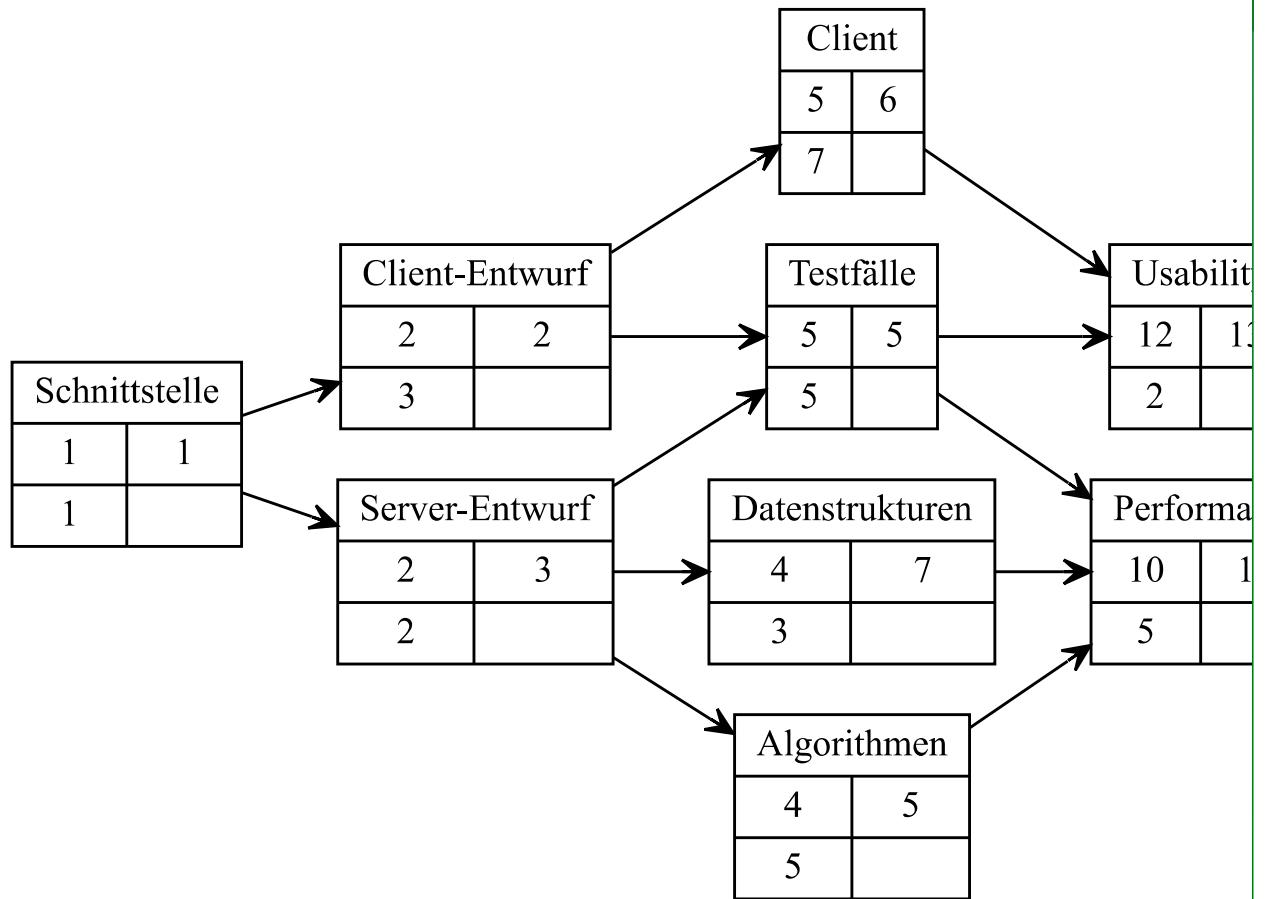
▼ Lösungsvorschlag



► Quelltext des Diagramms

3. Rechnen Sie rückwärts: Berechnen Sie ausgehend vom Projektende, wann die Vorgänge spätestens begonnen werden müssen (ohne dass sich das Projektende verändert) und annotieren Sie diese Werte im Netzplan. (0,5 Punkte)

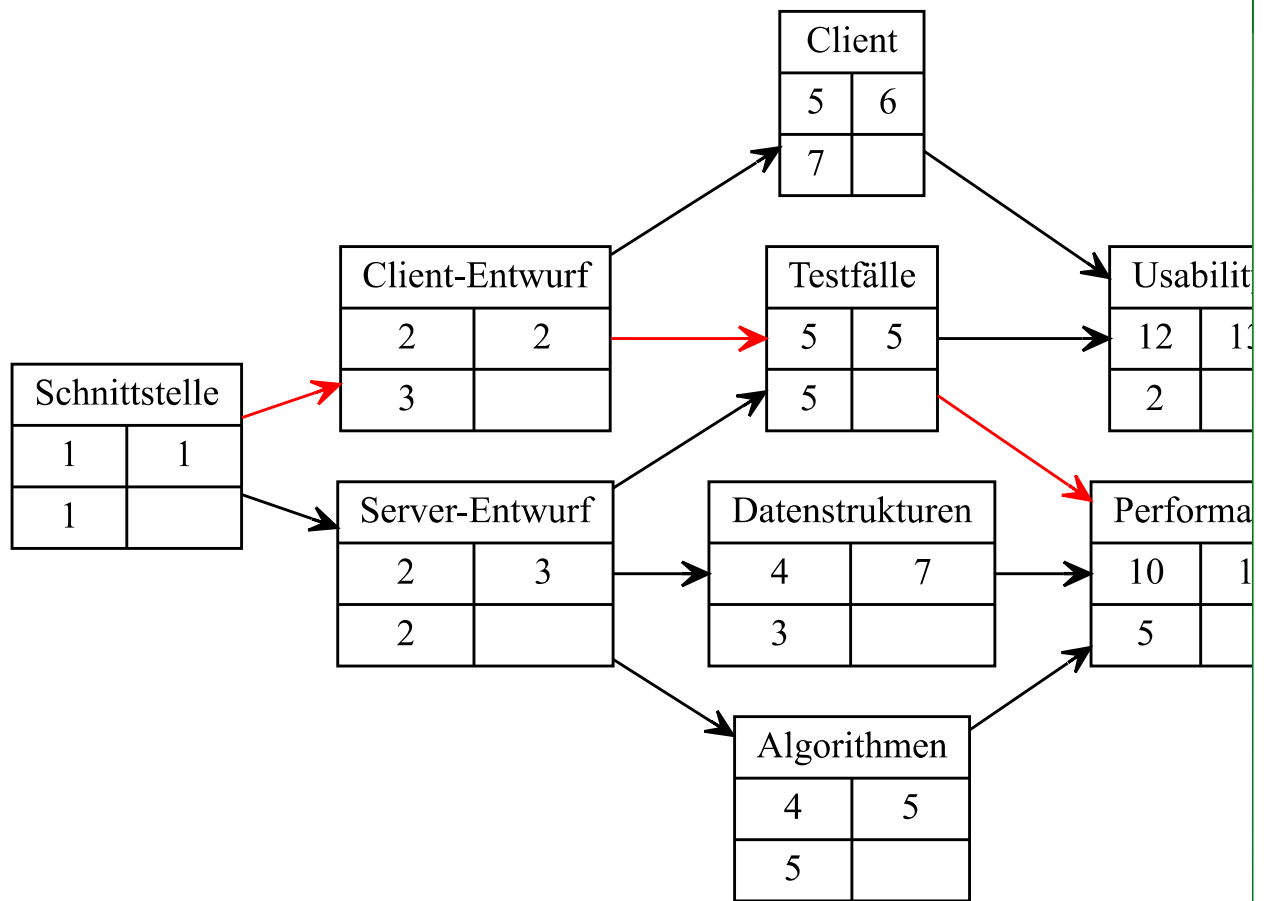
▼ Lösungsvorschlag



► Quelltext des Diagramms

4. Geben Sie einen kritischen Pfad an. (0,5 Punkte)

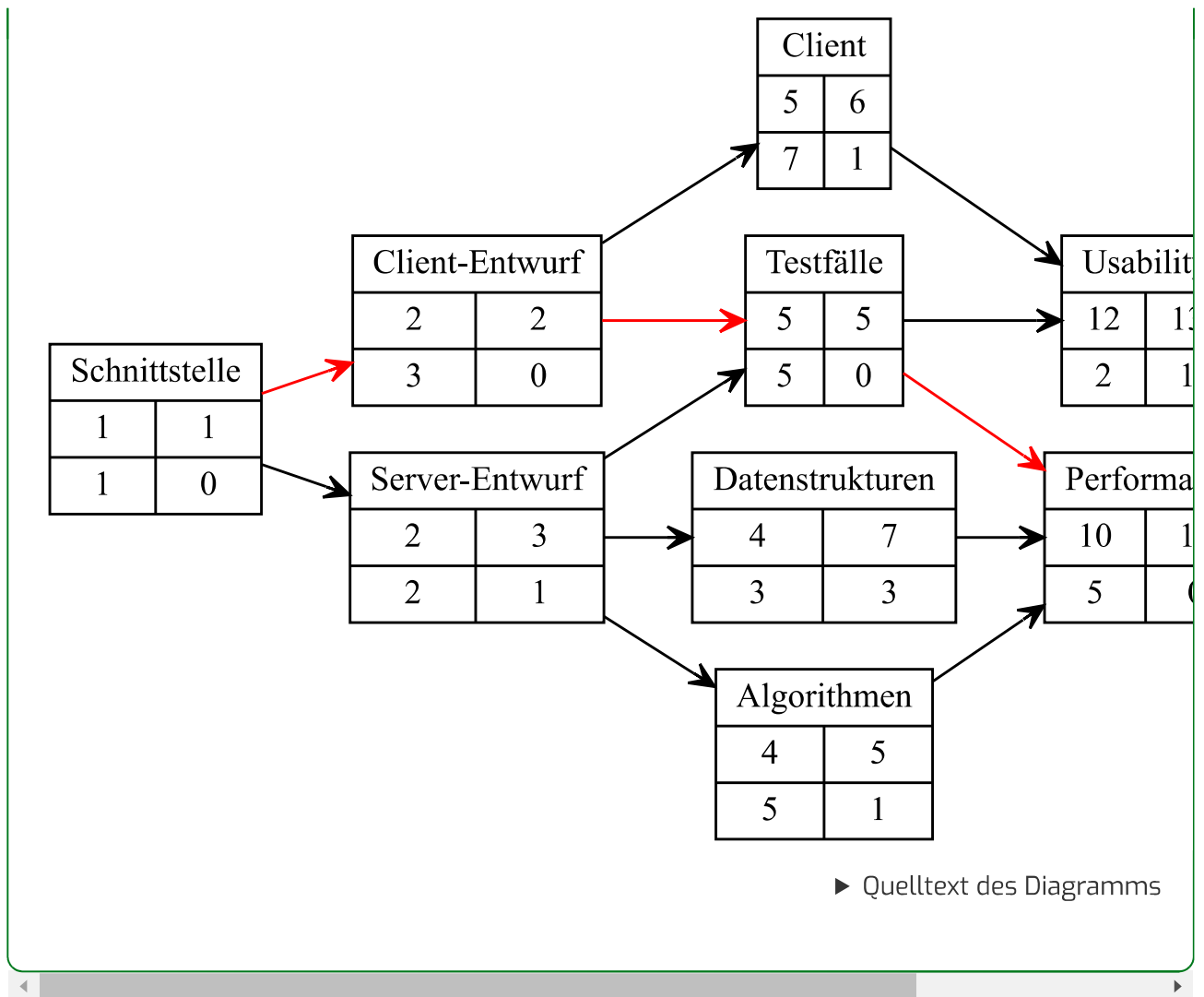
▼ Lösungsvorschlag



► Quelltext des Diagramms

5. Bestimmen Sie sämtliche Pufferzeiten. (0,5 Punkte)

▼ Lösungsvorschlag



Aufgabe 7.2: Gantt-Diagramm

4 Punkte, mittel

Erstellen Sie zu dem Netzplan aus der vorherigen Aufgabe ein optimales Gantt-Diagramm. Gehen Sie dabei davon aus, dass nur drei Personen in Ihrem Team sind, wobei jede Person pro Tag nur eine Aufgabe bearbeiten kann und jede Aufgabe auch nur von einer Person bearbeitet werden kann.

Verwenden Sie in Ihrem Diagramm zudem auch die folgenden drei Meilensteine:

- **M1.** Implementierung des Clients und Entwurf der Testfälle wurden abgeschlossen.
- **M2.** Implementierung der beiden Server-Komponenten und Entwurf der Testfälle wurden abgeschlossen.
- **M3.** Das Projekt wurde insgesamt abgeschlossen.

▼ Lösungsvorschlag

Folgendes Diagramm zeigt die Aufteilung der Aufgaben auf die Personen A, B und C:

Name	Dauer	1	2	3	4	5	6	7	8
Schnittstelle	1	A							
Client-Entwurf	3		A	A	A				
Server-Entwurf	2		B	B					
Testfälle	5					A	A	A	A
Client	7						C	C	C
Datenstrukturen	3				B	B	B		
Algorithmen	5				C	C		B	B
Usability	2								
Performanz	5								
Abschluss	2								

Die Aufteilung auf konkrete Ressourcen, also Personen, war in der Aufgabe nicht verlangt und dient hier nur der Illustration, dass nie mehr als drei Personen gleichzeitig am Projekt arbeiten.

Bei der Erstellung des Gantt-Diagramms muss die Begrenzung auf maximal drei parallel aktive Vorgänge berücksichtigt werden. Bei der Vorwärts- und Rückwärtsrechnung im Netzplan ging es nur darum, den kritischen Pfad und die Pufferzeiten zu finden, um herauszufinden, welche Arbeitspakete besonders kritisch sind in Bezug auf die Abhängigkeiten. In dieser Aufgabe geht es darüber hinaus auch um die Ressourcen-Planung, also die Aufteilung der Arbeitspakete auf Personen. Da wir hier davon ausgehen, dass wir nicht mehr als drei Personen haben, können schlicht nicht vier Personen gleichzeitig arbeiten. Deswegen muss der Vorgang *Algorithmen* unterbrochen werden. Dadurch, dass der Vorgang *Client* einen Tag später als zum frühest möglichen Termin startet, reicht es aus, den Vorgang *Algorithmen* für einen Tag zu unterbrechen. Da beide Vorgänge im

Netzgraph einen Puffer von 1 haben, hat weder die Unterbrechung noch die Verschiebung eine Auswirkungen auf die Gesamtdauer des Projektes.

Da das Projekt insgesamt nicht länger dauert als im Netzplan ermittelt, können wir sicher sein, dass das Gantt-Diagramm optimal ist. Kürzer als im Netzplan ermittelt kann die Projektlaufzeit unter Einhaltung der Abhängigkeiten nicht werden. Das die im Netzplan ermittelte optimale Projektlaufzeit auch mit den gegebenen Ressourcen noch eingehalten werden kann, ist allerdings nicht immer gegeben. Wenn wir zum Beispiel nur zwei Personen zur Verfügung hätten, würde sich die Projektlaufzeit signifikant verlängern.

Obige Tabelle entspricht nur grob einem Gantt-Diagramm. Beachten Sie die Hinweise zur Darstellung von Gantt-Diagrammen am Ende dieses Dokuments!

Aufgabe 7.3: GUI-Programmierung mit Java

4 Punkte, mittel

Machen Sie diese Aufgabe gründlich und stellen Sie sicher, dass der Code auf Ihrem Computer funktioniert. Wir werden im Laufe des Semesters auf dieses Beispiel zurückkommen.

Erweitern Sie Ihr Programm aus Aufgabe 2.3 um eine GUI mit [JavaFX](#). Zur Erinnerung: Das Programm nimmt drei Integer-Zahlen entgegen. Jede Zahl soll dabei als Seitenlänge eines Dreiecks verstanden werden. Das Programm gibt aus, ob das Dreieck mit diesen drei Seitenlängen gleichschenkelig, gleichseitig oder ungleichseitig ist.

Die GUI soll die drei Zahlen in drei mit *a*, *b* und *c* beschrifteten Eingabefeldern entgegen nehmen und die Ausgabe in einem Label anzeigen.

Nutzen Sie [Maven](#) als Build-System und Java-Version 11 als Source- und Target-Version, sowie das [Maven plugin for JavaFX](#), sodass sich ihr Projekt mit `mvn javafx:run` starten und mit `mvn javafx:jlink` bauen lässt.

Reichen Sie Ihren Code als Zip-Archiv im Moodle ein. Achten Sie dabei darauf, tatsächlich nur den Quellcode und keine gebauten Artefakte mit abzugeben.

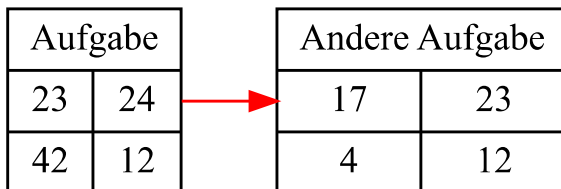
▼ Lösungsvorschlag

Siehe `triangle-gui.zip`.

Hinweise zum Erstellen der Diagramme

Netzpläne

Netzpläne können leicht mit leicht mit [GraphViz](#) gesetzt werden. GraphViz ist in [PlantUML](#) enthalten, was wiederum in [CodiMD](#) (oder HedgeDoc) direkt eingebunden ist.



► Quelltext des Diagramms

Gantt-Diagramme

Gantt-Diagramme können als einfache Tabelle gesetzt werden:

Folgendes Diagramm zeigt die Aufteilung der Aufgaben auf die Personen A, B und C:

Name	Dauer	1	2	3	4
Schnittstelle	1	A			
Client-Entwurf	3		A	A	A
Server-Entwurf	2		B	B	

Obige Tabelle entspricht nur grob einem Gantt-Diagramm. In einem richtigen Gantt-Diagramm würde man die Bearbeitung einer Aufgabe durch einen kontinuierlichen Balken darstellen. Man könnte die Kanten des Netzplans noch zusätzlich im Gantt-Diagramm einzeichnen, allerdings werden Gantt-Diagramme mit zu vielen Informationen auch schnell unübersichtlich. Darüber hinaus lassen sich entsprechende Pfeile in tabellarische Darstellung meist nur umständlich ergänzen. Spezialisierte Software wie zum Beispiel Microsoft Project kann hier weiter helfen. Die

Gantt-Diagramme von PlantUML sind leider für diese Aufgabe nutzlos, da Unterbrechungen von Aufgaben zum Beispiel aus Mangel an Ressourcen hier nicht vernünftig abgebildet werden können.

Hinweise zur Verwendung von Maven und JavaFX

Verwenden Sie das im Moodle bereitgestellte JavaFX-Template: Das Maven-Projekt ist ein sehr einfaches JavaFX-Beispiel. Das Beispiel nutzt das [Maven plugin for JavaFX](#). Über folgenden Befehl können Sie das Projekt starten:

```
mvn javafx:run
```

Darüber hinaus können Sie über folgenden Befehl mit JLink ein angepasstes Java Runtime Image bauen:

```
mvn javafx:jlink
```

Die kompilierte Applikation kann in diesem Image ausgeführt werden über folgenden Aufruf:

```
./target/example/bin/example
```

oder unter Windows:

```
target\example\bin\example.bat
```

Der Name `example` ist dabei als `jlinkImageName` in der Maven-Konfigurationsdatei `pom.xml` festgelegt und kann natürlich durch einen sinnvollere Namen ersetzt werden.

Unter `target` entsteht darüber hinaus ein Zip-Archiv des Images, das weitergegeben und auf anderen Computern ohne weitere Abhängigkeiten ausgeführt werden kann.