JAVA 04

Planung und Fehlerbehandlung

JAVA 04 – Planung und Fehlerhandling

1. **Lastenheft und Pflichtenheft**

2. **Designdokument und UML-Diagramme**

3. **Fehlerbehandlung**

1. Lastenheft und Pflichtenheft

Das Lastenheft beschreibt Anforderungen von einem Kunden an die Software. Das Pflichtenheft beschreibt alle Feature der Software, auf die sich Kunde und Entwickler geeinigt haben. Gegebenenfalls sind hier auch Hardwareanforderungen und weitere Details definiert. Beide Dokumente sind sehr umfangreich und meist auch Grundlage für einen Vertrag...

Diese Art der Planung ist ein guter allererster Schritt wenn man einen Auftrag bekommt oder ein großes Projekt durchführen will – ***aber für unsere Zwecke fiel zu weit gegriffen***!

2. Designdokument und UML-Diagramme

Das (Software) Designdokument (**SDD**) ähnelt Lasten und Pflichtenheft, ist aber nicht so umfangreich und bei weiten nicht so strikt reglementiert. In Kombination mit **UML**-Diagrammen sind damit auch kleinen Programme VOR dem programmieren besser vorbereitet und sehr klar definiert. Sind SDD und UML gut, muss man beim Programmieren nicht mehr darüber nachdenken, was man überhaupt machen will.

Die Reihenfolge bei der Erstellung von SDD und UML ist nicht starr, manchmal ergibt sich durch das UML ein Detail im SDD. Übertragungen von Erkenntnisse vom SDD zum UML sind der Regelfall.

Sollte man jedoch erst während des Programmierens zu viele Änderungen an SDD oder in den UML vornehmen müssen, zeugt das von… **suboptimaler Planung** oder unerwarteten Änderungen der Anforderungen…

2. Designdokument und UML-Diagramme

Beispiel: Aufgabe 4 von Teil 1:

Schreibe ein Programm zur Bestimmung des *kleinsten gemeinsamen Vielfaches (kgV)* zweier natürlicher Zahlen.

Beispiel: Zahl1 = 5, Zahl2 = 7, kgV = 35

2. Designdokument und UML-Diagramme

-> Umwandlung in ein Designdokument

Gegeben sind zwei natürliche Zahlen (**a** und **b**).
Gesucht wird das kleinste gemeinsame Vielfache.

Dies bedeutet ein Vielfaches **A** der **Zahl a** muss dasselbe ergeben
wie das Vielfache **B** der **Zahl b**!

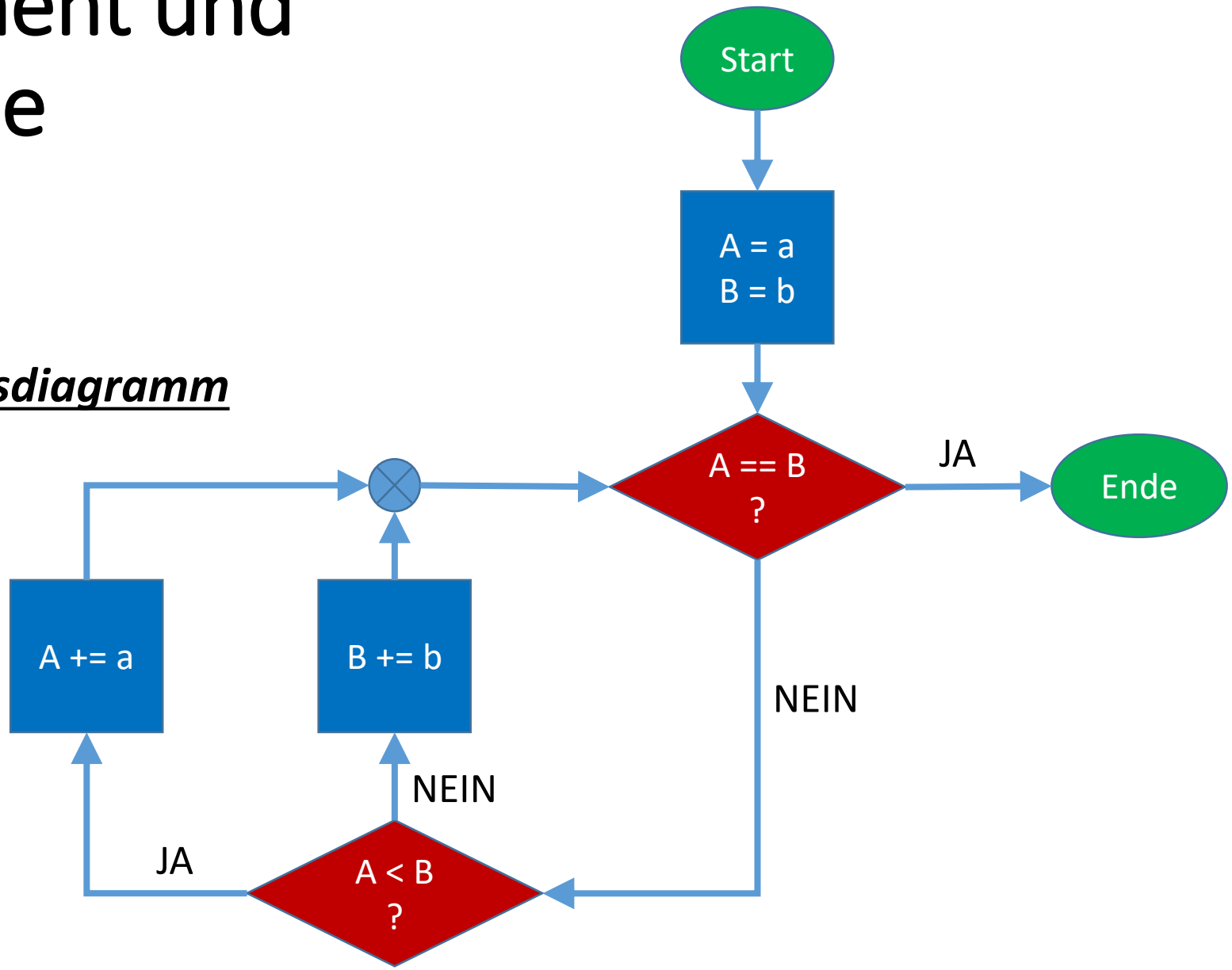
Lösungsansatz ausformulieren:
Wir starten mit dem Wert der jew. Zahl (**A = Wert a** und **B = Wert b**).
Anschließend ***erhöhen*** wir ***den kleineren Wert*** von **A** bzw. **B**
um den Wert a bei A bzw. b bei B **solange bis A und B gleich sind**.

2. Designdokument und UML-Diagramme

-> UML-Diagramme

Für dieses Beispiel:

das ***Ablaufdiagramm / Flussdiagramm***



3. Fehlerbehandlung

Wenn ein Fehler passiert, wird das JAVA-Programm normalerweise sofort beendet. Wenn man dies nicht will, kann man kritische Programmteile absichern und bei einem Fehler reagieren.

```
try {
        // in diesem Block koennte etwas schief gehen,
        // dann wird ein Fehler „geschmissen“
} catch (Throwable th) {
        // der Fehler wird „abgefangen“
        // und dieser Block wird alternativ ausgeführt
}
```

WICHTIG: Diese Prozedur soll auch helfen Fehler besser zu finden. Daher sollte man vermeiden zu viel zusammenhängenden Code innerhalb eines „try“ Blocks zu schreiben!