

1 Phasen bei der Durchführung eines Softwareprojekts.

4

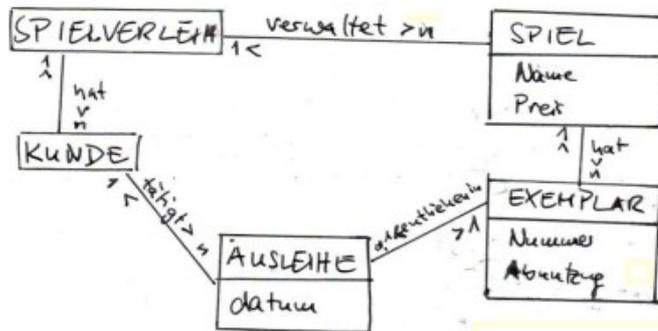
z.B.: Das Wasserfallmodell

- Anforderungsanalyse: Festlegung der Anforderungen an das System. Ergebnis ist eine Anforderungsbeschreibung → dient als "Vertrag" zwischen Anwender/Entwickler
- Entwurf: Modellierung des Systems
- Implementierung: Codierung des Entwurfs in einer Programmiersprache
- Test und Integration: Test der einzelnen Komponenten, Einbau, Systemtest
- Einsatz & Wartung: Fehlerbeseitigung nach Inbetriebnahme

Hinweis: Da die Wartung nicht mehr zur eigentlichen Erstellung gehört, muss sie daher nicht aufgeführt sein. Auch andere Modelle als das Wasserfallmodell können vorgestellt werden.

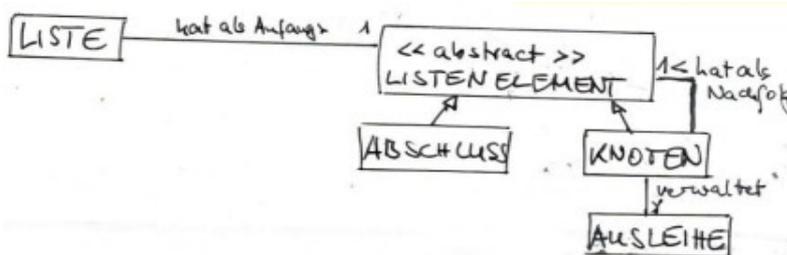
2a

6



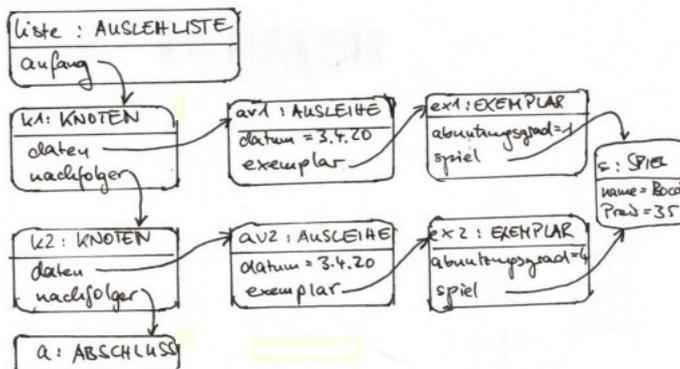
b

4

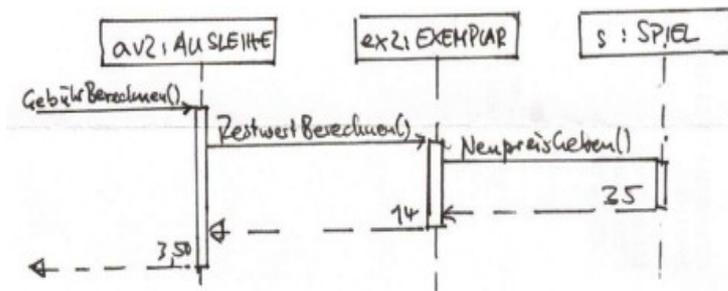


c

7



d



5

e Lösung in Python

7

Klasse Liste:

```
def entfernen(self, ex: Exemplar) -> None:
    self.anfang.entfernen(ex)
```

Klasse Listenelement:

@abc.abstractmethod

```
def entfernen(self, ex: Exemplar) -> Listenelement:
    pass
```

Klasse Abschluss:

```
def entfernen(self, ex: Exemplar) -> Listenelement:
    return self
```

Klasse Knoten:

```
def entfernen(self, ex: Exemplar) -> Listenelement:
    if self.daten.exemplarGeben() == ex:
        return self.nachfolger
    else:
        self.nachfolger.entfernen(ex)
    return self
```

Zur Erläuterung für die Java-NutzerInnen:

Der Rückgabebetyp steht hinter dem Pfeil „->“

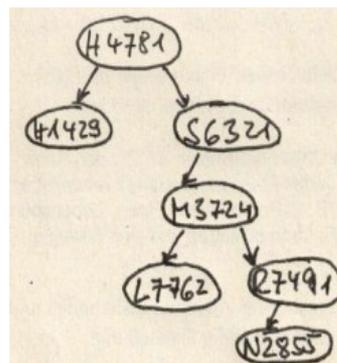
self entspricht this und ist in der Regel notwendig als erstes Element im Methodenkopf und bei jedem Aufruf an das Objekt selbst.

3a H4781 → H1429 → S6321 → M3724

5

H4781 → S6321 → H1429 → M3724

H4781 → S6321 → M3724 → H1429



b Beim Inorder-Durchlauf wird erst der linke Nachfolger dazu aufgefordert, die Prozedur zu starten, dann gibt der Knoten seinen Wert aus, danach wird der rechte Nachfolger dazu aufgefordert.

4

Die Kunden bekommen ihre Zahl nicht zwingend lexikographisch zugeordnet. Es ist z. B. nicht gesagt, dass Hr. Herbert eine kleinere Nummer als Frau Huber erhält.

c $2500 \leq 2^n - 1$ 2
 $\Leftrightarrow 2501 \leq 2^n$
 $\Leftrightarrow \lg 2501 \leq n$
 $\Leftrightarrow 11,2 \leq n$
 Es sind also 12 Stufen notwendig.

d Ein balancierter Baum stellt sicher, dass die Ebenenzahl minimal ist und somit der Baum schnell durchsucht werden kann. Ein Nachteil ist der hohe Verwaltungsaufwand beim balancierten Einfügen. 3

e Jeder Knoten besitzt nicht mehr nur einen Nachfolger, sondern einen linken und einen rechten. 2

f Lösung in Python 10

```
Klasse Kundenbaum:
def passwortPrüfen(self, kdnr:int, pw:str) -> bool:
    self.wurzel.passwortPrüfen(kdnr, pw)
```

```
Klasse Bauelement
@abc.abstractmethod
def passwortPrüfen(self, kdnr:int, pw:str) -> bool:
    pass
```

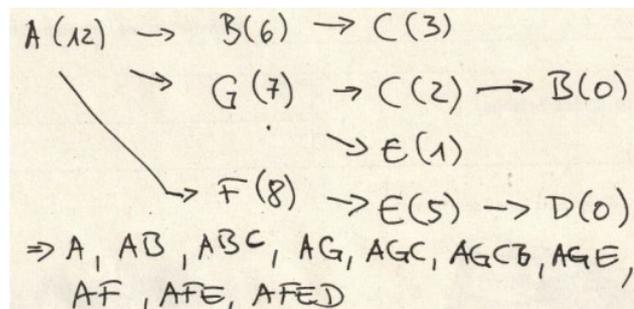
```
Klasse Abschluss:
def passwortPrüfen(self, kdnr:int, pw:str) -> bool:
    return False
```

```
Klasse Knoten:
def passwortPrüfen(self, kdnr:int, pw:str) -> bool:
    k:int = self.daten.kundennummerVergleichen(kdnr)
    if k==1:
        return self.liNf.passwortPrüfen(kdnr, pw)
    elif k==0:
        return self.daten.istPasswortGleich(pw)
    elif k==-1:
        return self.liNf.passwortPrüfen(kdnr, pw)
```

4a 6

	A	B	C	D	E	F	G
A		6				4	5
B	6		3				
C		3		4			4
D			4		5		
E				5		3	6
F	3				3		
G	5		4		6		

b



5

c Methode alleZüge(aktKnotenNr, Punkte)
Setze alle Knoten auf unbesucht
Besuchen(aktKnotenNr, Punkte, " ")
endeMethode

10

Methode Besuchen(knotenNr, Punkte, Pfad)
Setze knoten[knotenNr] auf besucht
Pfad = Pfad + knoten[knotenNr].BezeichnerGeben()
Ausgeben(Pfad, Punkte)
wiederhole für i von 0 bis knoten.länge-1
wenn $0 < \text{matrix}[\text{knotenNr}][i]$ und $\text{matrix}[\text{knotenNr}][i] \leq \text{Punkte}$ und knoten[i] unbesucht
Besuchen(i, Punkte-matrix[knotenNr][i], Pfad)
Setze knoten[knotenNr] auf unbesucht

80