

# 3. Übung zur Vorlesung Software-Produktlinien



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Programmierparadigmen für Software-Produktlinien - Präprozessoren

## Aufgabe 1 C-Präprozessor

- Geben Sie für das Feature-Modell in Abbildung 1 eine Präprozessor-Kodierung (1) als einfaches Makro und (2) als Makro mit Argumenten an und geben Sie jeweils für beide Darstellungen die entsprechende Kodierung der Konfiguration  $p = \{\text{BEV}, \text{TEA}\}$  an.

- Geben Sie für den Term

```
[i:int,f:float] optimum ([i:int,f:float] x, [i:int,f:float] y) {  
    if (x [min:<,max:>] y)  
        return x;  
    else  
        return y;  
}
```

eine äquivalente C-Implementierung (1) mit Präprozessor-Kodierung und (2) mit Variability Encoding an.

- Geben Sie für die folgenden Direktiven die jeweilige Ausgabe des C-Präprozessors an.

a) `foo = X;`  
`#define X 4`  
`bar = X;`

b) `/*`  
`*/ # /*`  
`*/ defi\  
ne FO\  
0 10\  
20`  
`bar = F00;`

c) `#define min(X,Y) ((X) < (Y) ? (X) : (Y))`  
`min (1,2)`  
`min (min (1,2),2)`

d) `#define COMMAND(NAME) { #NAME, NAME ## _command}`  
`struct command commmands[] = {`  
 `COMMAND (quit),`  
 `COMMAND (help),`  
 `...`  
`};`

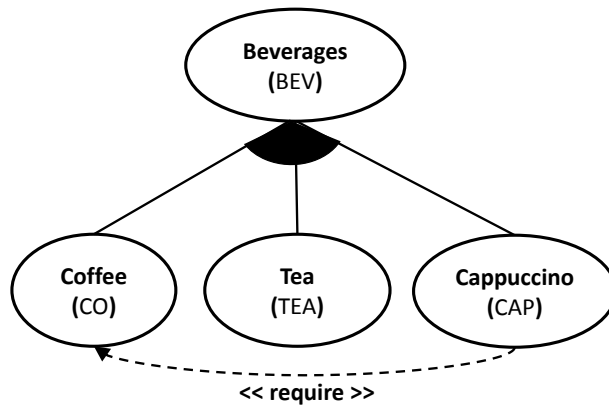


Abbildung 1: Ausschnitt aus dem Feature-Modell der Vending Machine SPL

- e) `#define double(x) (2*(x))`  
`#define call_with_1(x) x(1)`  
`call_with_1 (double)`
- f) `#define CALLFOO(X) foo(X,`  
`x = CALLFOO(1) 2);`
- g) `#define foo (4 + foo)`

Korrektur: Expansionsstelle "foo"  
fehlt