

Aufgabe 1: Symbolliste mit einfachen Typen

Gegeben sei folgendes Programm:

```

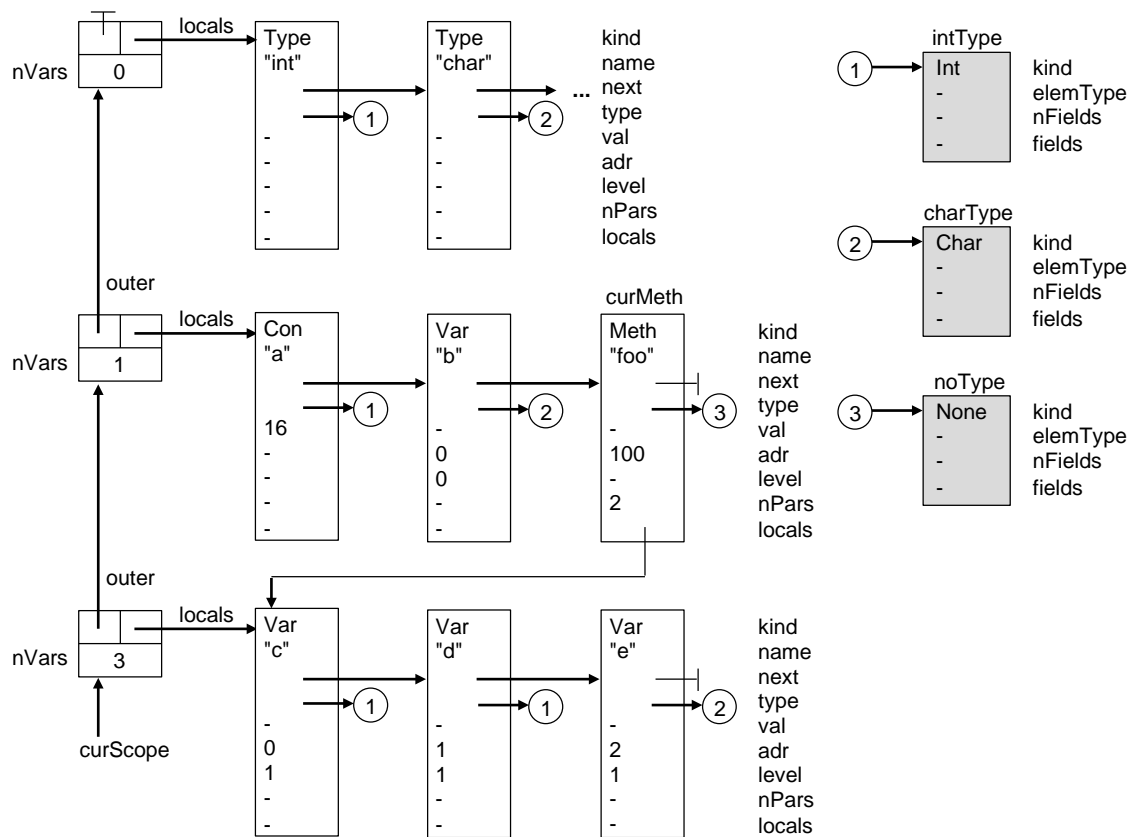
program P
  final int a = 16;
  char b;
  {
    void foo (int c, int d)
      char e;
      { ... // <== 1
      }

    void bar()
      int f, g;
      { ... // <== 2
      }
  }
  
```

- a) Zeichnen Sie die Symbolliste zum Zeitpunkt, an dem sich der Compiler an der Stelle 1 (in foo) befindet. Zeichnen Sie alle Scopes mit ihren Objekt- und Strukturknoten. Für das Universum reicht es, einige Beispielknoten zu zeichnen. Geben Sie auch die Werte der Felder in Objekt- und Strukturknoten an.
- b) Zeichnen Sie die Symbolliste zum Zeitpunkt, an dem sich der Compiler an der Stelle 2 (in bar) befindet.

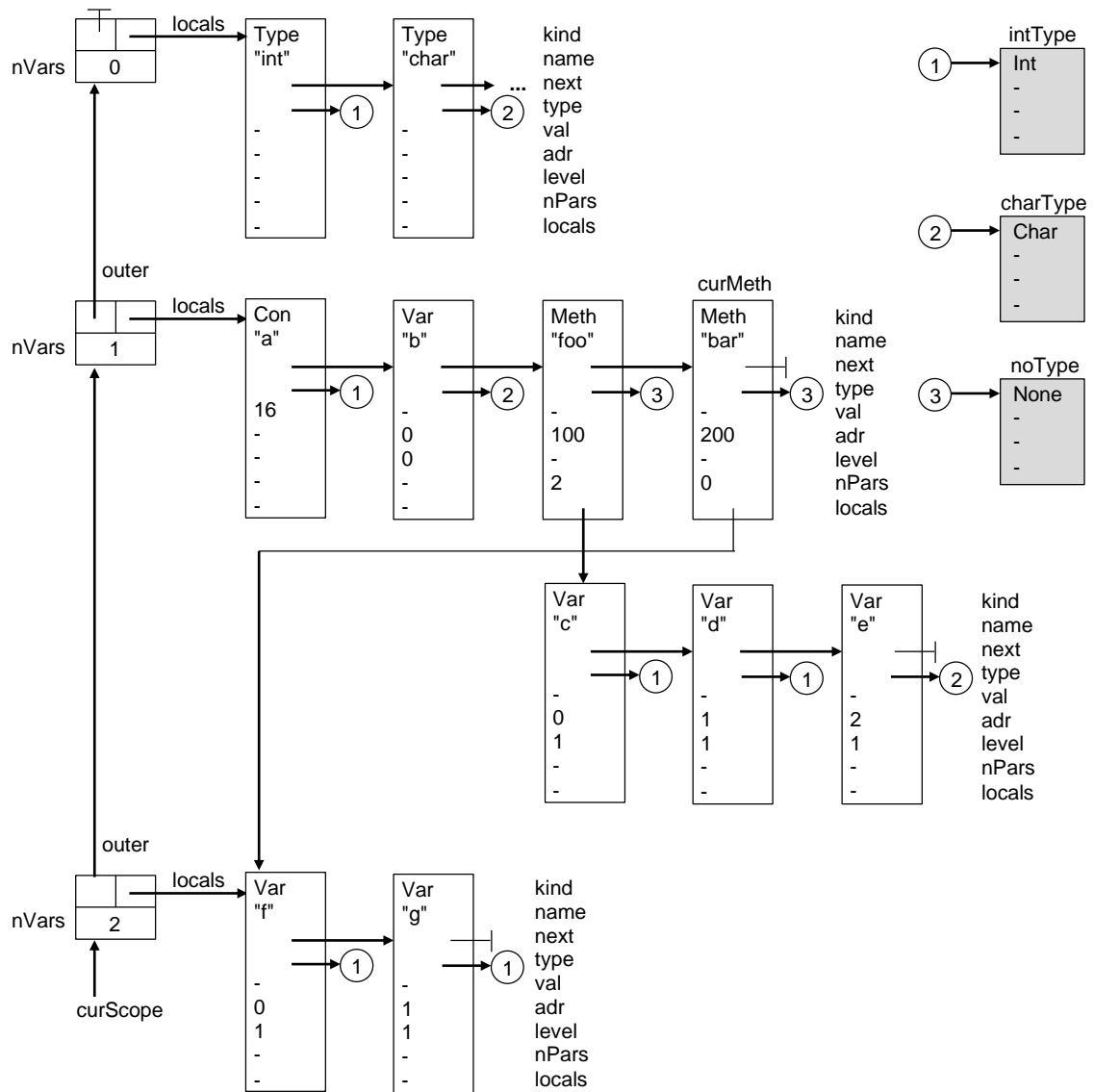
Lösung

a) Symbolliste an der Stelle 1 (Annahme: foo hat die Adresse 100 im Codebereich).



Es existiert je ein Scope für das Universum, für das Programm P und für die Methode foo (Objektknoten curMeth). Die Variablen des Programmscopes haben level = 0, die Variablen des Methodenscopes level = 1. Die Adressen der Variablen werden fortlaufend bei 0 beginnend vergeben.

b) Symbliste an der Stelle 2 (Annahme: bar hat die Adresse 200 im Codebereich).



Der Scope von foo wurde abgebaut und ein neuer Scope für bar wurde geöffnet. Es existiert wieder ein Scope für das Universum, einer für das Programm P und einer für die Methode bar (Objektknoten curMethod). Die lokalen Objekte der Methode foo hängen aber immer noch am Objektknoten von foo.