

4.1 Abstrakte Syntax für Java

Erstellen Sie eine UML-Klassendiagramm für die in Vorlesung vorgestellten abstrakten Syntax.

4.2 Programmieren in abstrakter Syntax

Schreiben folgendes Java-Programm in abstrakter Syntax als Haskell-Programm

```
class AbsSyn {
    int i;
    int j;

    void m(int a, int b) {
        while(b < 15) {
            i++;
            m(a, i);
        }

        return a;
    }
}
```

4.3 Prettyprint für die abstrakte Syntax (Prüfungsaufgabe)

In Haskell kann Prettyprint-Funktionen für beliebige Datentypen mit Instanzen der Typklasse `Show` deklarieren:

```
data Folge = Empty | Cons(Integer , Folge)

instance Show Folge where
    show Empty = "[]"
    show (Cons(e, Empty)) = (show e)
    show (Cons(e, es)) = (show e) ++ ", " ++ (show es)
```

Schreiben Sie für die Abstrakte Syntax von Java eine Instanz der Typklasse `Show` und testen Sie die Prettyprint-Funktion mit den Beispiel aus Aufgabe 4.2.