

## Aufgabe 2: Suchbaum

- (a) Implementieren Sie in einer objektorientierten Sprache einen binären Suchbaum für ganze Zahlen! Dazu gehören Methoden zum Setzen und Ausgeben der Attribute `zahl`, `linker_teilbaum` und `rechter_teilbaum`. Design: eine Klasse `Knoten` und eine Klasse `BinBaum`. Ein `Knoten` hat einen linken und einen rechten Nachfolger. Ein `Baum` verwaltet die Wurzel. Er hängt neue `Knoten` an und löscht `Knoten`.
- (b) Schreiben Sie eine Methode `fuegeEin(...)`, die eine Zahl in den Baum einfügt!
- (c) Schreiben Sie eine Methode `postOrder(...)`, die die Zahlen in der Reihenfolge postorder ausgibt!
- (d) Ergänzen Sie Ihr Programm um die rekursiv implementierte Methode `summe(...)`, die die Summe der Zahlen des Unterbaums, dessen Wurzel der `Knoten` ist, zurückgibt! Falls der Unterbaum leer ist, ist der Rückgabewert 0!

```
1  int summe (Knoten x)...
```

- (e) Schreiben Sie eine Folge von Anweisungen, die einen Baum mit Namen `BinBaum` erzeugt und nacheinander die Zahlen 5 und 7 einfügt! In den binären Suchbaum werden noch die Zahlen 4, 11, 6 und 2 eingefügt. Zeichnen Sie den Baum, den Sie danach erhalten haben, und schreiben Sie die eingefügten Zahlen in der Reihenfolge der Traversierungsmöglichkeit postorder auf!
- (f) Implementieren Sie eine Operation `isSorted(...)`, die für einen (Teil-)baum feststellt, ob er sortiert ist.