



Java – Eine pragmatische Einführung

Sommersemester 2024

3. Übungsblatt

Hinweise: Die Abgabeuhrzeit ist verschoben auf 23:59 Uhr. Das Hochladen von Abgaben in Moodle ist ab diesem Blatt nur noch möglich, wenn Sie für eine Übungsgruppe angemeldet sind.

Aufgabe 8 (Generics, Erweiterung der Klasse Pair, 6 Punkte)

Machen Sie zunächst eine Kopie der Klasse `Generics/2-wildcard/Pair.java` aus dem Quellcode-Archiv zur Vorlesung.

- a) (3 P.) Modifizieren Sie die Klasse nun so, dass die Attribute `elem0` und `elem1` zwei verschiedene Typen `T0` und `T1` haben dürfen.
- b) (1 P.) Ergänzen Sie die Methode `toString()`.
- c) (1 P.) Ergänzen Sie die Methode `equals`.

Hinweis: Testen Sie zunächst, ob der zum Vergleich übergebene Parameter ein `Pair`-Objekt ist. Falls ja, können Sie diesen in ein Wildcard-`Pair` casten. Vergleichen Sie dann die Typen der Elemente und erst danach im Fall passender Typen deren Inhalte.

- d) (1 P.) Ergänzen Sie die Methode `clone()`.

Hinweis: Erzeugen Sie hier eine flache Kopie; die Elemente werden also nicht geklont, sondern Original und Klon enthalten danach Referenzen auf identische Elemente.

Testen Sie alle Änderungen jeweils im Hauptprogramm `main`.

Aufgabe 9 (Generics, Exceptions, Maximierer, 4 Punkte)

Schreiben Sie eine Klasse `Maximierer<T>` `extends Comparable<T>`, die das Maximum mehrerer Elemente vom Typ `T` bestimmt, die nacheinander an den Maximierer übergeben werden. Dabei soll nur das jeweilige bisherige Maximum gespeichert werden, nicht alle übergebenen Werte. Die folgenden Methoden sind gefordert:

- a) `public Maximierer<T> neuesElement(T t)`: prüft, ob `t` größer ist als das bisherige Maximum und ersetzt letzteres in diesem Fall; geben Sie den Maximierer selbst zurück, damit Sie in `main` mehrere Aufrufe hintereinanderhängen können.
- b) `public T getMaximum()`: gibt das bisherige Maximum zurück, falls schon Werte übergeben wurden; ansonsten wird eine Exception geworfen

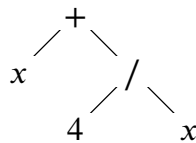
Tipp: Verwenden Sie `null` um anzuzeigen, dass noch keine Werte übergeben wurden.

Testen Sie im Hauptprogramm mindestens mit einem Wrappertyp und mit `String`. Fangen Sie außerdem mindestens eine Exception.

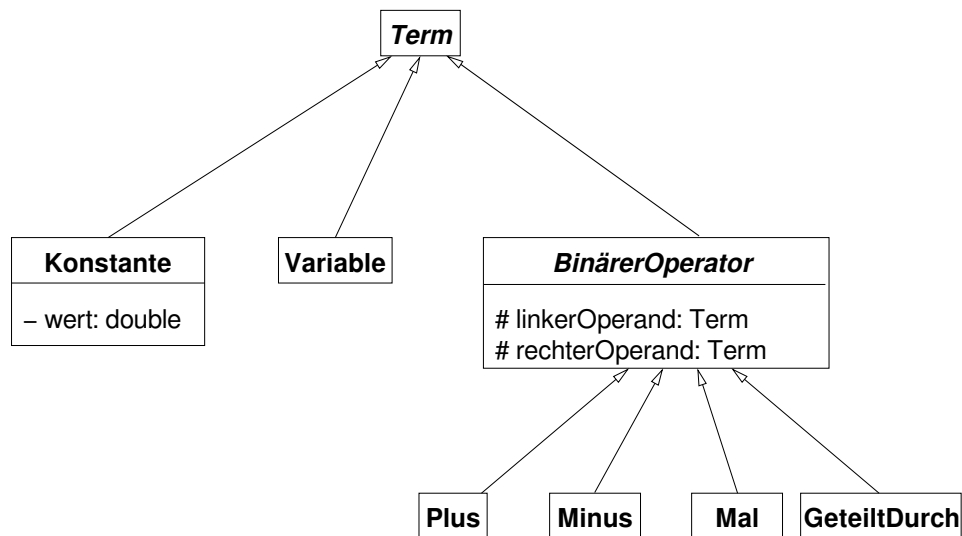
Aufgabe 10 (abstrakte Klassen, Exceptions, Terme, symbolisches Differenzieren, Nullstellen, 12 Punkte)

Implementieren Sie Java-Klassen für Terme.

Terme werden häufig als Bäume gespeichert, z. B. der Term $t(x) = x + 4 / x = +(x, /(4, x))$ als:



- a) (2 P.) Fangen Sie damit an, die Klassen aus dem folgenden UML-Diagramm zu implementieren (Klassen mit schäggestellten Namen sind abstrakt, # steht für protected).



- b) (2 P.) Ergänzen Sie Konstruktoren in allen Klassen, die eigene oder geerbte Attribute besitzen.
- c) (2 P.) Ergänzen Sie eine Methode `public double auswerten(double x)` in der Klasse `Term` und in allen nicht-abstrakten Unterklassen. Ergebniswert:
- wert für Konstanten
 - x für Variable
 - Beide Operanden werden ausgewertet und dann addiert/subtrahiert/multipliziert/dividiert für Plus/Minus/Mal/GeteiltDurch.
- d) (2 P.) Fügen Sie die Methode `public String toString()` in allen nicht-abstrakten Klassen hinzu. Setzen Sie runde Klammern um die Ausgaben aller vier binären Operatoren, z. B. $((2 \cdot x) - 3)$.
- e) (2 P.) Ergänzen Sie die Methode `public Term ableiten()` in `Term` und in allen nicht-abstrakten Klassen.

Erinnerung an die wichtigsten Ableitungsregeln: $(f \pm g)(x) = f'(x) \pm g'(x)$,

$$(f \cdot g)(x) = f'(x) \cdot g(x) + f(x) \cdot g'(x), \left(\frac{f}{g}\right)(x) = \frac{f'(x) \cdot g(x) - f(x) \cdot g'(x)}{g(x) \cdot g'(x)}$$

- f) (2 P.) Implementieren Sie schließlich in der Klasse `Term` die Methode `public static double newton(Term t, double x0)`, die das Newtonverfahren zur Nullstellenbestimmung mit Startwert x_0 auf den Term t anwendet.

Algorithmus Newtonverfahren

Eingabe Funktion f , Startwert x_0

für $i = 0, 1, \dots$

$$x_{i+1} := x_i - \frac{f(x_i)}{f'(x_i)}$$

Beenden Sie die Iteration, wenn $|x_{i+1} - x_i| < 10^{-14}$. Werfen Sie eine Exception, falls nach 100 Iterationen keine Konvergenz eingetreten ist.

Auf der Moodle-Seite finden Sie die Datei `TermTest.java`, die ein vollständiges Hauptprogramm mit einigen Tests enthält.

Abgabe: Di., 28.05.2024, 23:59 Uhr