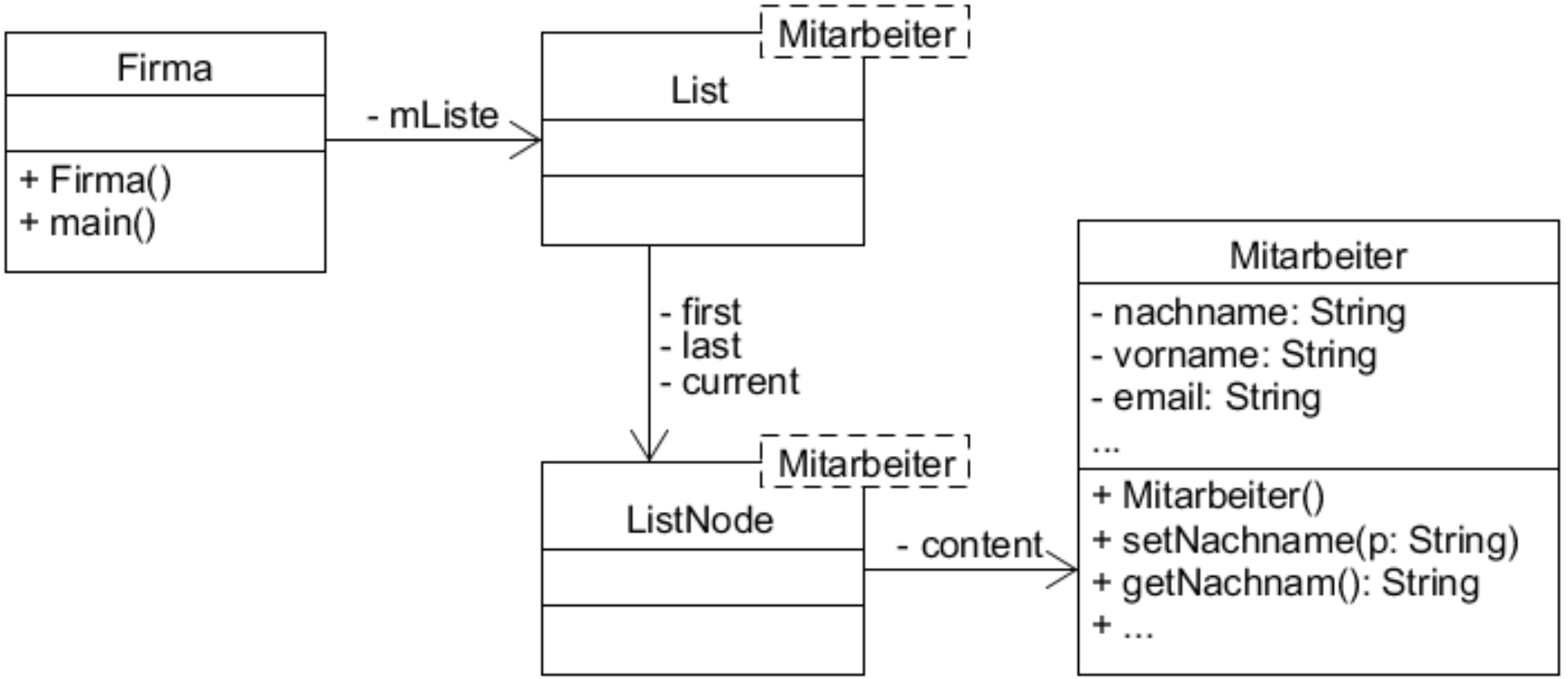
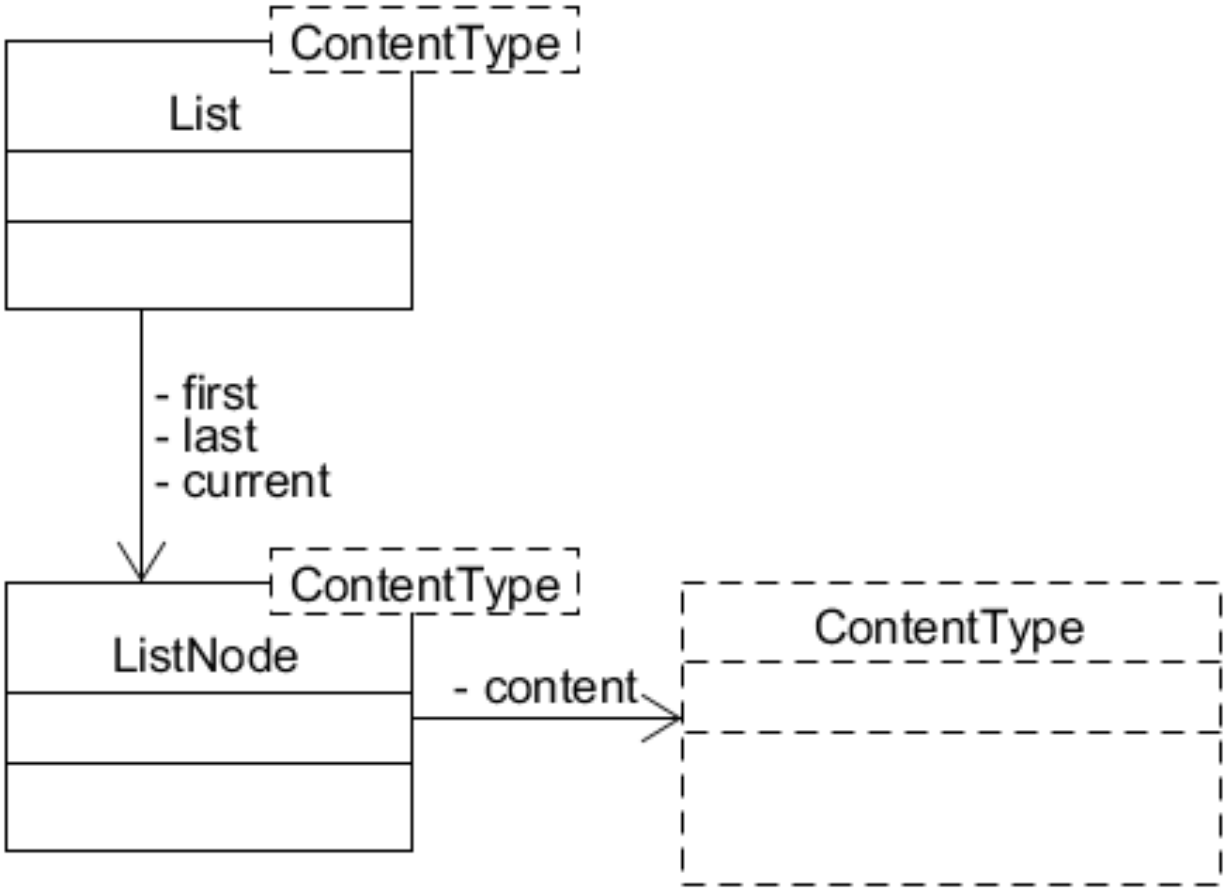
Interfaces

Beispiel: Liste

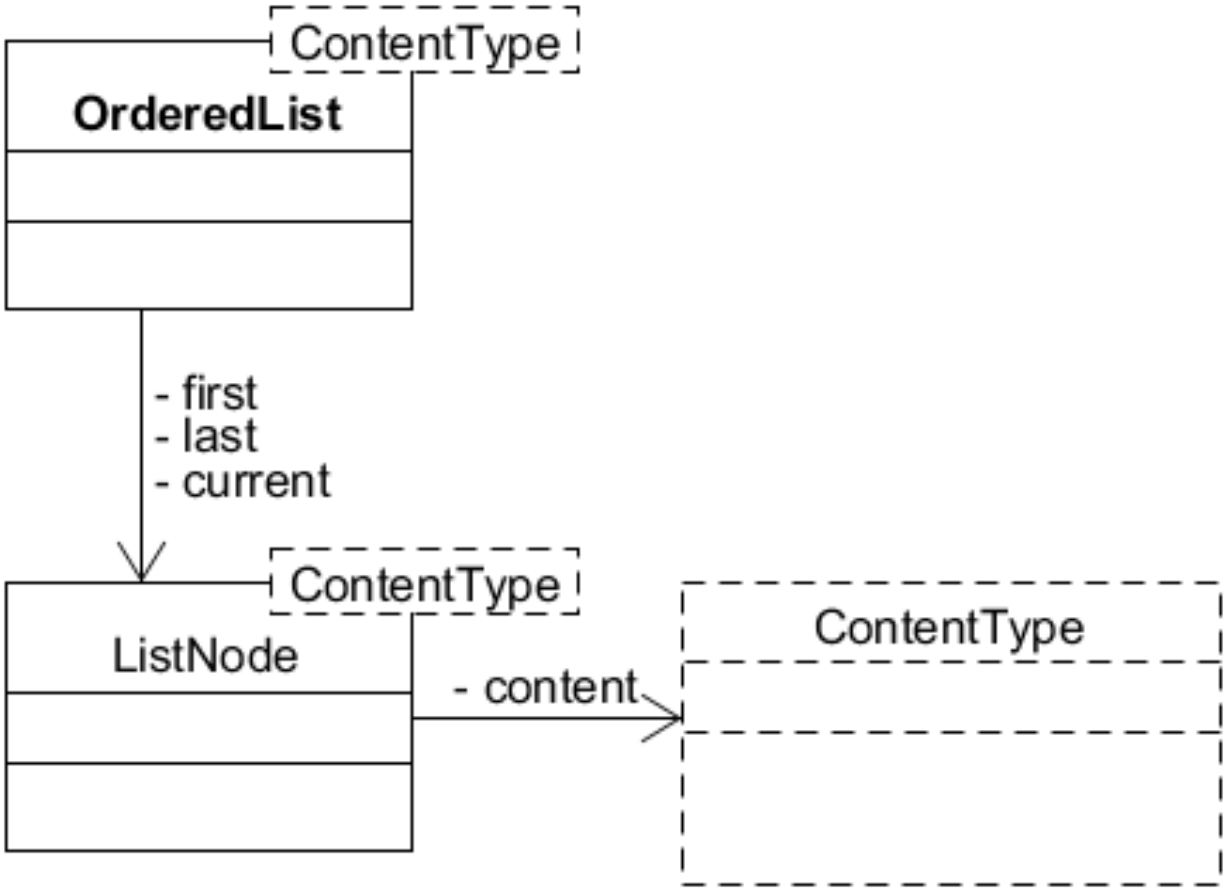


Wünschenswert: Liste **geordnet** nach Nachnamen

Klasse List allgemein

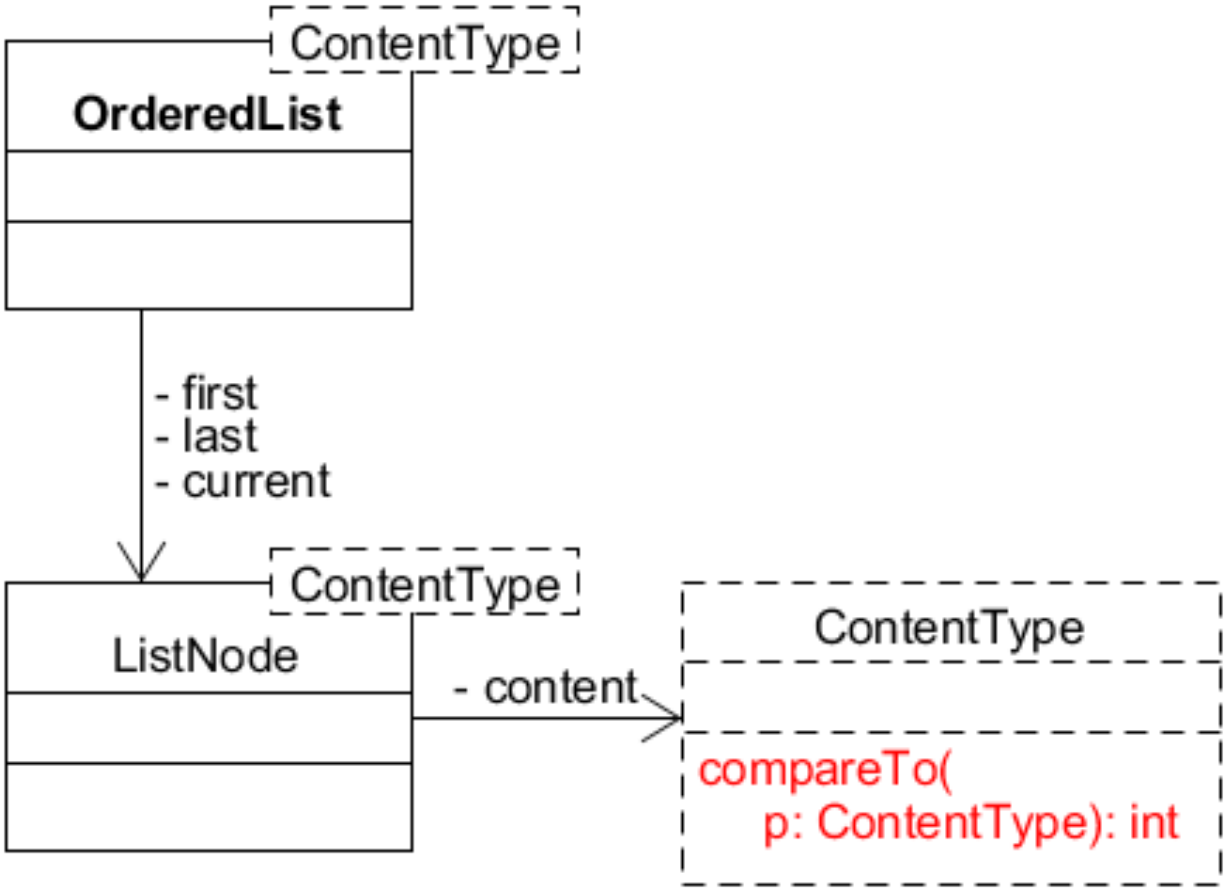


Idee: OrderedList



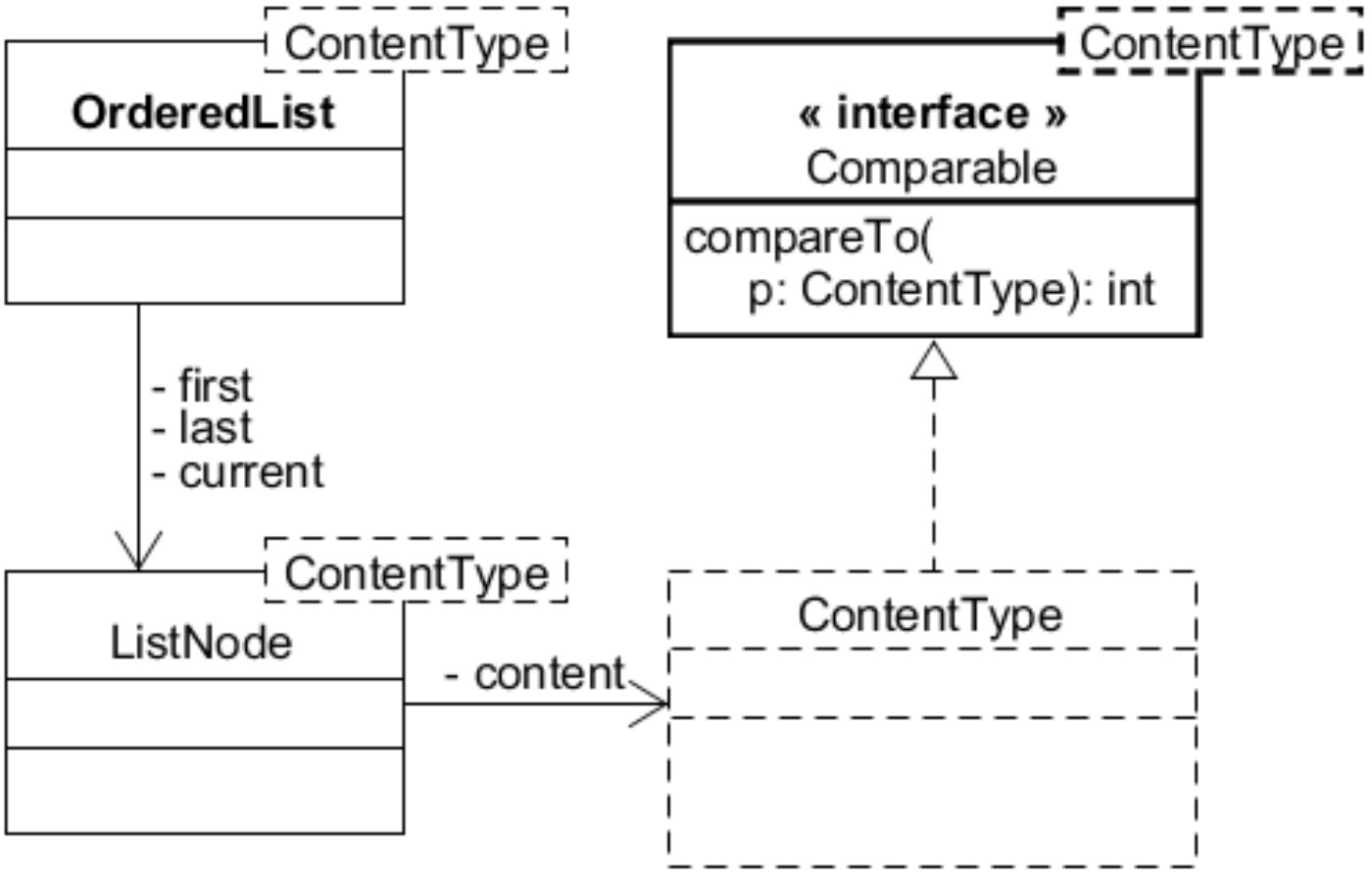
Statt List (ungeordnet) → OrderedList (geordnet)
Inhalte müssen aber **vergleichbar** sein!

Idee: OrderedList



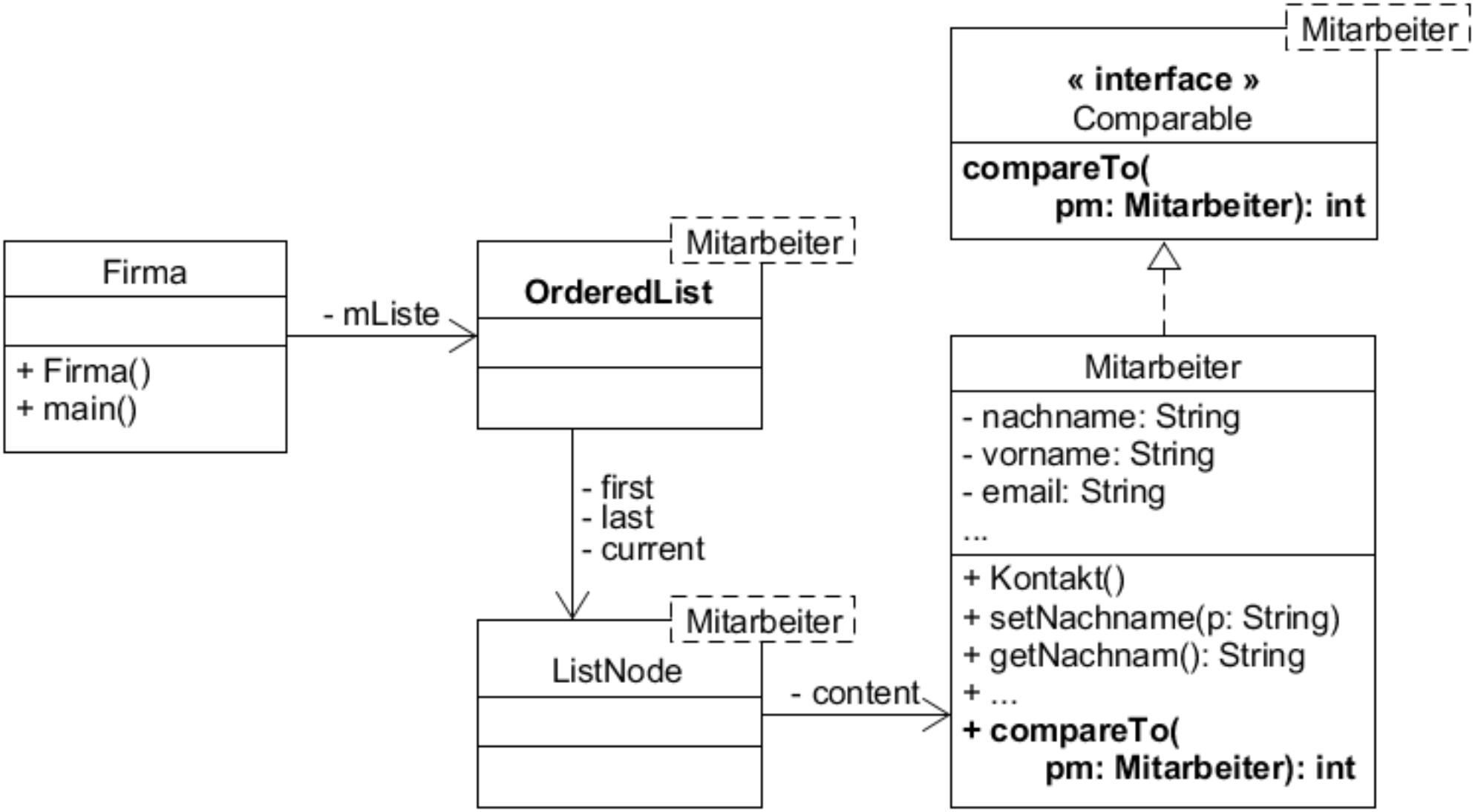
Leider gibt es die Klasse „ContentType“ nicht
→ kann keine Methoden definieren

Vorschrift durch Interface



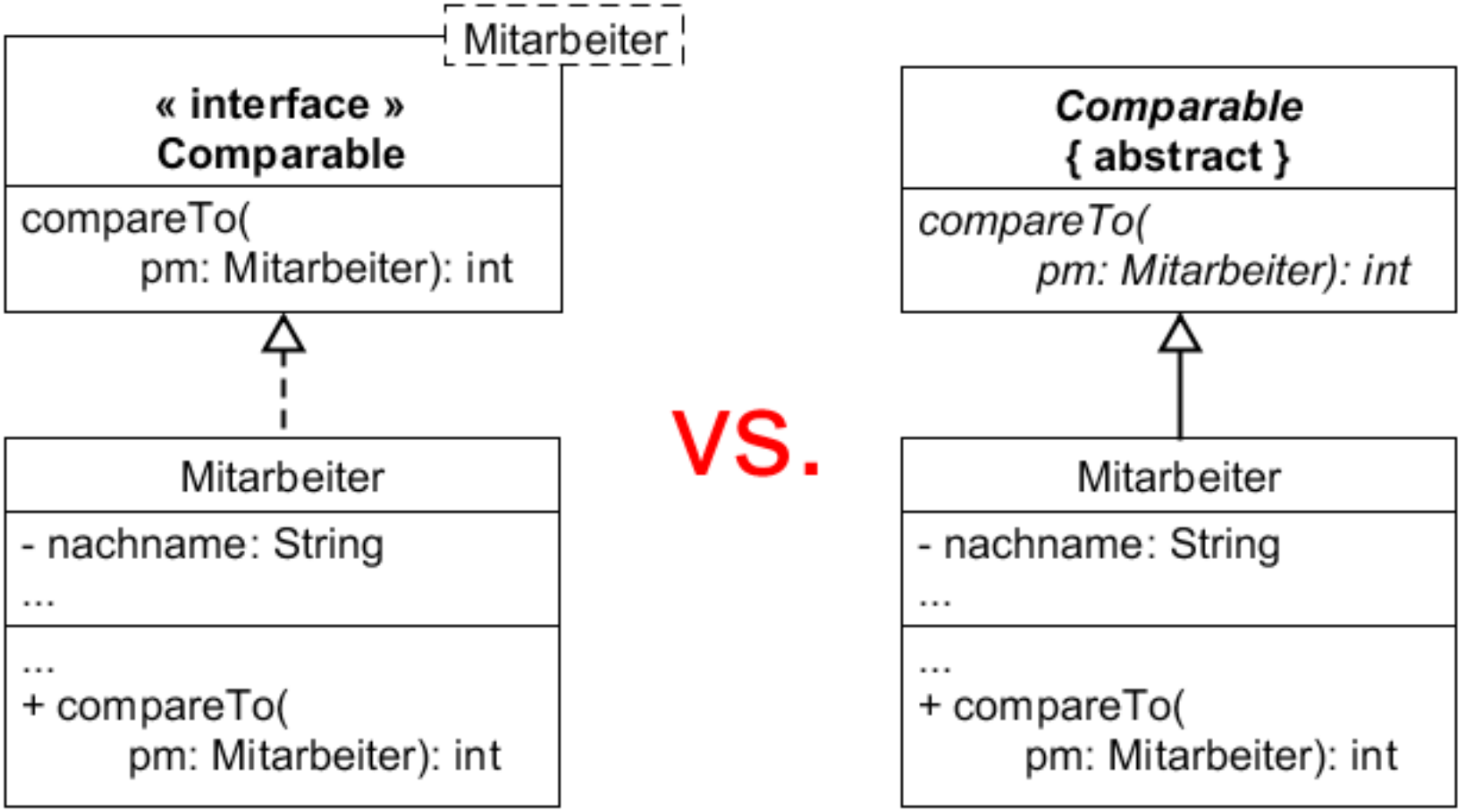
Das **Interface** „Comparable“ schreibt vor:
Die Klasse, die ContentType ersetzt, **muss** die Methode compareTo() implementieren.

Modell mit OrderedList



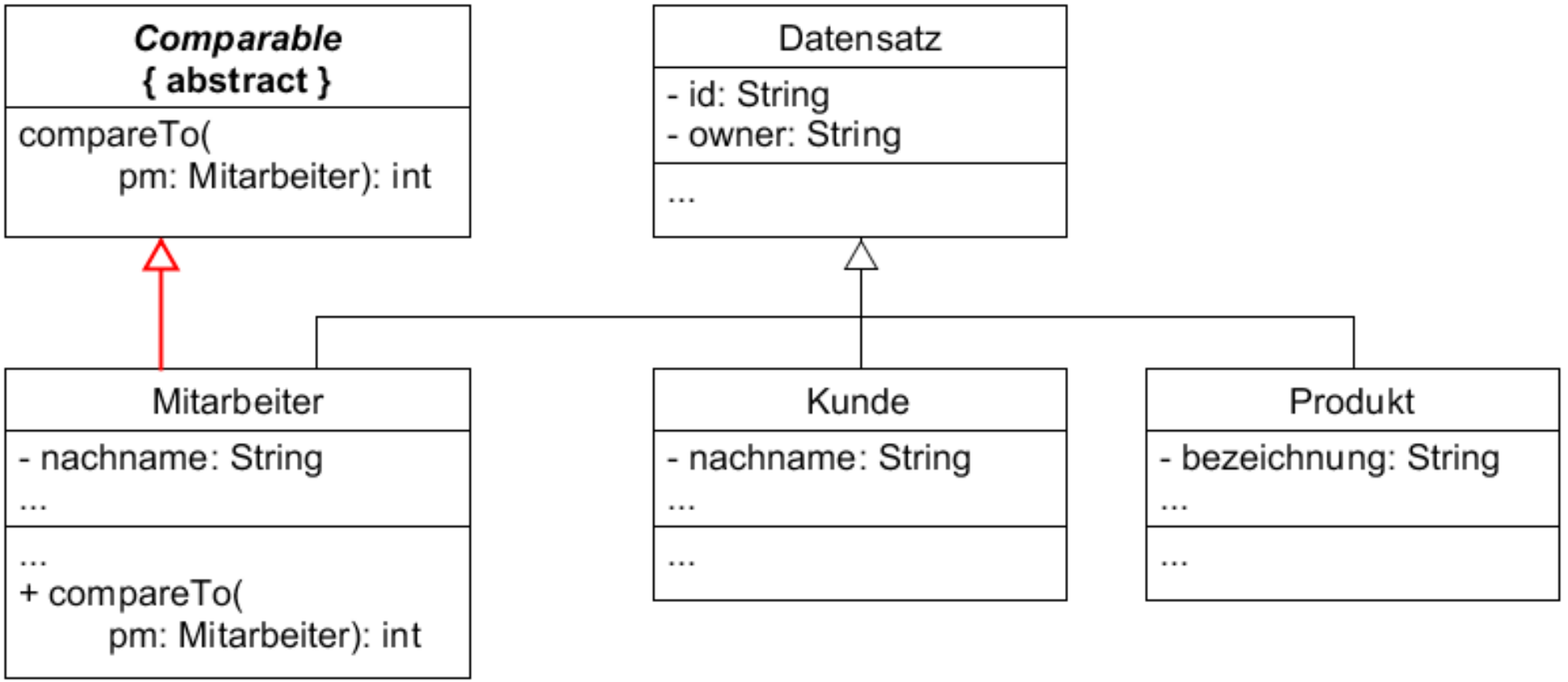
Klasse Mitarbeiter **implementiert** Interface Comparable
→ damit die Methode compareTo()

Interface vs. Vererbung



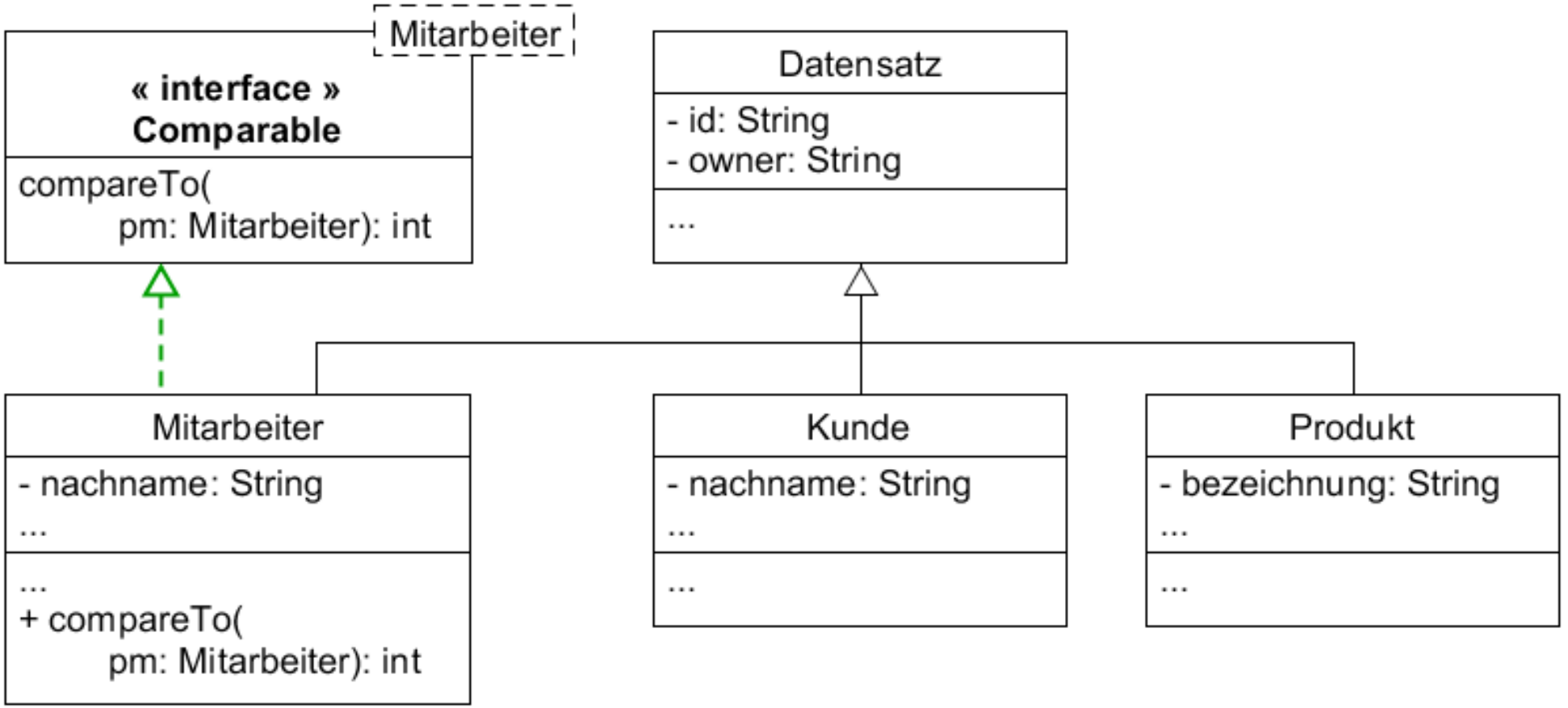
Wozu braucht es ein neues Konzept?
Hätte eine abstrakte Oberklasse nicht gereicht?

Interface vs. Vererbung



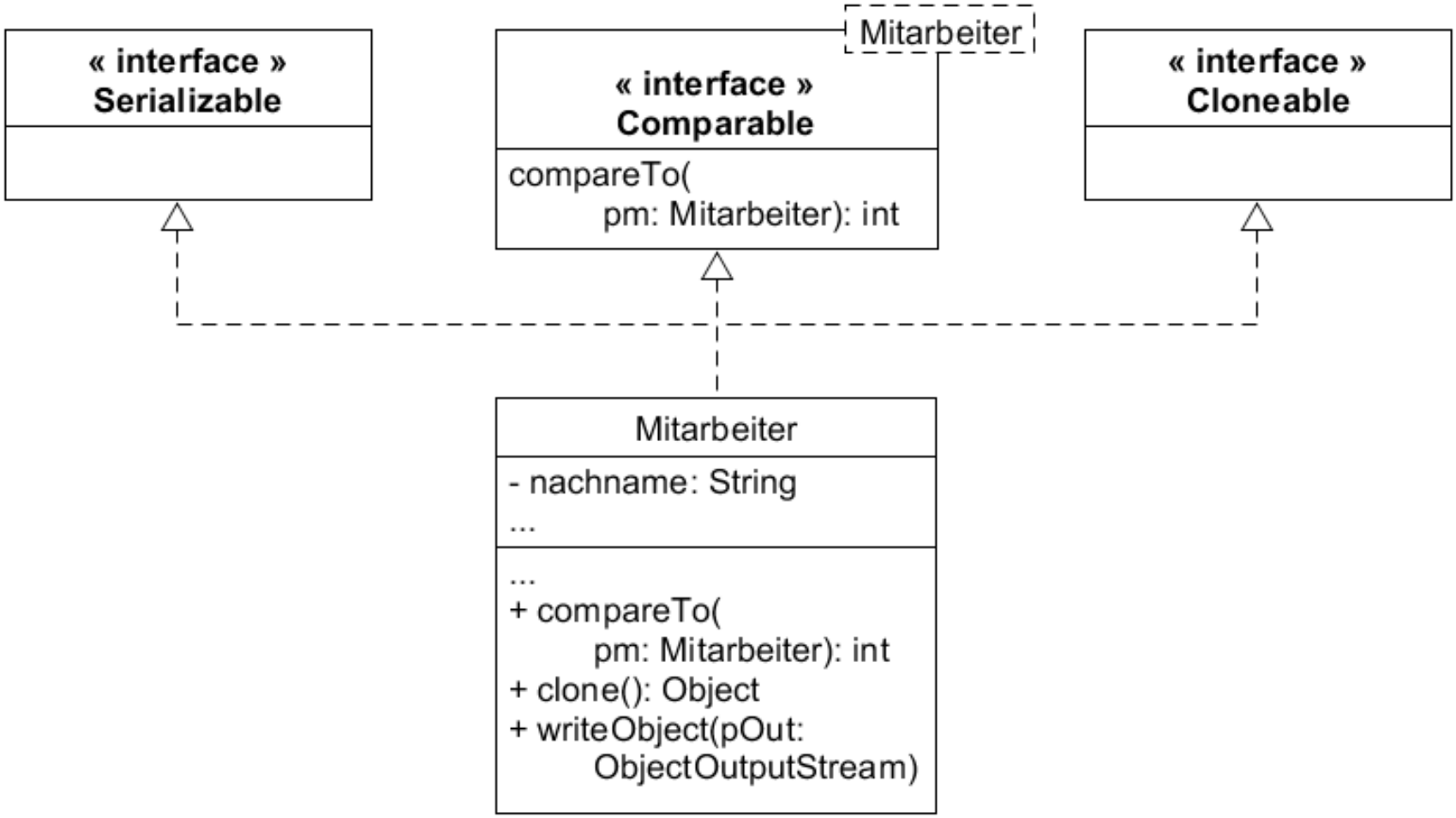
Mitarbeiter ist evtl. schon Teil einer Hierarchie
→ zwei Oberklassen sind nicht erlaubt!

Interface vs. Vererbung



Mit Interface kein Problem !

Interface vs. Vererbung



Evtl. sind Funktionen für weitere Kontexte nötig

→ **Mehrere Interfaces sind erlaubt**.

Vergleich

Interface	Abstrakte Klasse
kein Objekt erzeugbar	kein Objekt erzeugbar
hat nur Methoden ohne Implementierung	kann auch Methoden mit Implementierung haben
Klasse „implementiert“ Interface	Unterklasse „erbt von“ abstrakter Klasse
Mehrere Interfaces möglich	Mehrfachvererbung nicht zulässig
	

Autor / Quellen

Autor:

- Christian Pothmann (cpothmann.de)
  Freigegeben unter CC BY-NC-SA 4.0, Juni 2021