FOOSE SoSe 2010

Übung 1

Christian Kroiß

LMU München

05.05.2010

Aufgabe 1

Betrachtet wird die Datenstruktur der Vorrangschlangen (priority queues) mit natürlichzahligen Elementen. Zu einer leeren Vorrangschlange empty können mit Hilfe der Operation add nacheinander neue Elemente hinzugefügt werden. Für nicht-leere Vorrangschlangen liefert die Operation max das maximale Element, das sich am längsten in der Vorrangschlange befindet. Dieses kann mit der Operation remove entfernt werden.

a) Geben Sie eine sorten-geordnete Signatur Σ_{PQ} für Vorrangschlangen an.

b) Beschreiben Sie die Eigenschaften der oben genannten Operationen mit Hilfe von Termen des Typs *Boolean*.

1.a) Signatur

Grundidee:

- Natürliche Zahlen als Sorte erforderlich
- Unterscheidung zwischen leerer und nicht-leerer Warteschlange

$$
\begin{aligned}
\Sigma_{PQ} &= (S, \leq, F) \\
S &= \{Nat, NePQueue, PQueue\} \\
\leq &= \{NePQueue \leq PQueue\} \\
F &= \{empty :\rightarrow PQueue, \\
&\quad add : Nat \times PQueue \rightarrow NePQueue, \\
&\quad remove : NePQueue \rightarrow PQueue, \\
&\quad max : NePQueue \rightarrow Nat\}
\end{aligned}
$$

1.a) Signatur (2)

Grundidee:

- Natürliche Zahlen als Sorte erforderlich
- Unterscheidung zwischen leerer und nicht-leerer Warteschlange
- Für Terme: Einführung des Typs `Boolean` und $=, <, >, \leq, \geq$

$$
\begin{aligned}
F \quad = \quad & \{..., \\
& \mathit{true} :\rightarrow \mathit{Boolean}, \mathit{false} :\rightarrow \mathit{Boolean}, \\
& \_\mathit{implies}\_ : \mathit{Boolean} \times \mathit{Boolean} \rightarrow \mathit{Boolean}, \\
& \_ > \_ : \mathit{Nat} \times \mathit{Nat} \rightarrow \mathit{Boolean}, \\
& \geq, <, \leq \mathit{analog}, \\
& \_ = \_ : \mathit{Nat} \times \mathit{Nat} \rightarrow \mathit{Boolean}, \\
& \_ = \_ : \mathit{Boolean} \times \mathit{Boolean} \rightarrow \mathit{Boolean}, \\
& \_ = \_ : \mathit{PQueue} \times \mathit{PQueue} \rightarrow \mathit{Boolean}\}
\end{aligned}
$$

1.b) Terme

Vorgehen: Identifikation von Invarianten

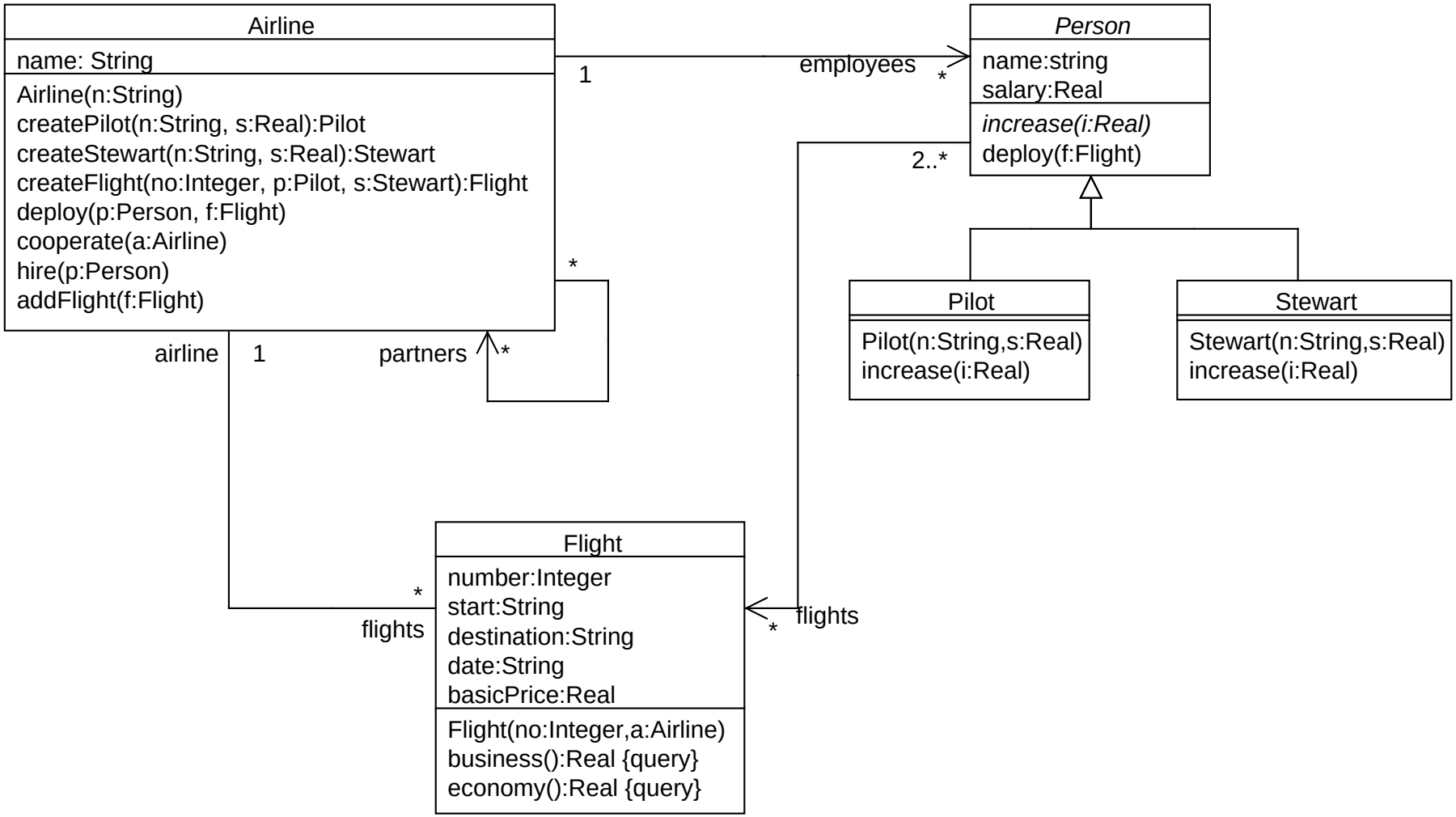
- Das Element mit dem höchsten Wert (Priorität) steht immer am Anfang der Warteschlange.
- Remove liefert immer das Element mit der höchsten Priorität.
- Falls mehrere Elemente die gleiche Priorität haben, liefert remove das älteste.

1.b) Terme

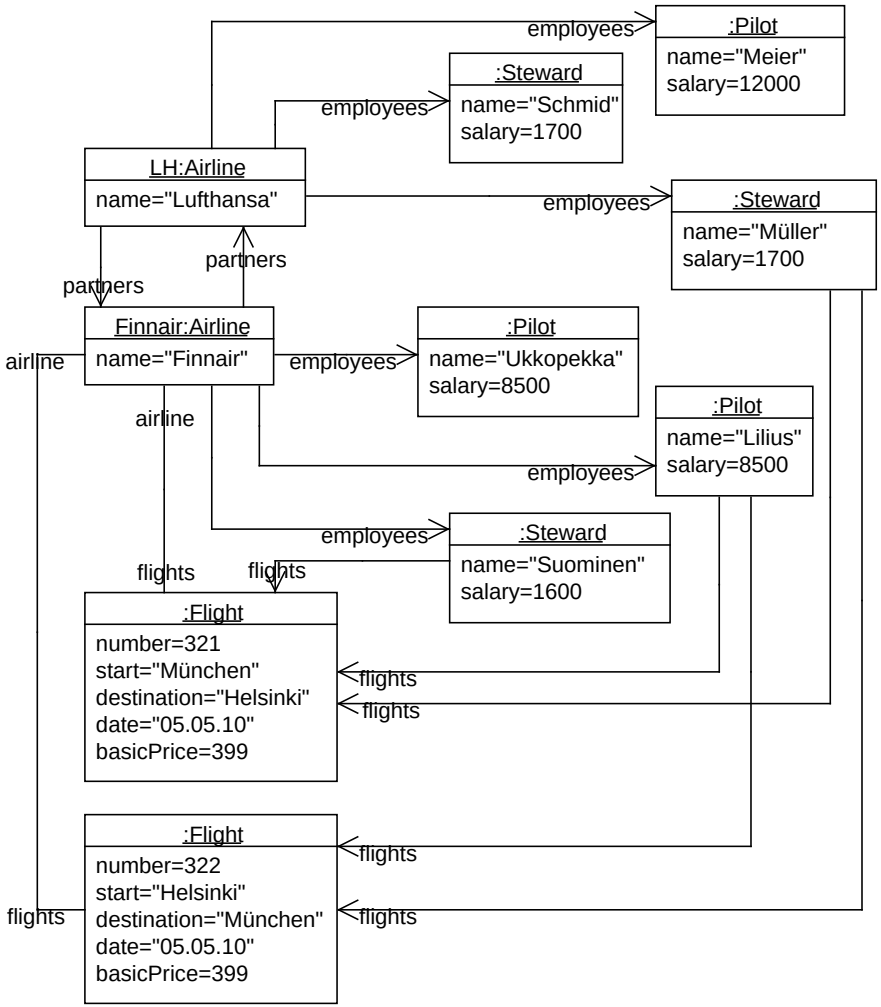
$$\textit{Seien} \qquad x : \textit{Nat}, p : \textit{NePQueue} \in X$$

$$\begin{aligned}
\textit{max}(\textit{add}(x, \textit{empty})) &= x \\
x > \textit{max}(p) \quad &\textit{implies} \quad \textit{max}(\textit{add}(x, p)) = x \\
x \leq \textit{max}(p) \quad &\textit{implies} \quad \textit{max}(\textit{add}(x, p)) = \textit{max}(p) \\
\textit{remove}(\textit{add}(x, \textit{empty})) &= \textit{empty} \\
x > \textit{max}(p) \quad &\textit{implies} \quad \textit{remove}(\textit{add}(x, p)) = p \\
x \leq \textit{max}(p) \quad &\textit{implies} \quad \textit{remove}(\textit{add}(x, p)) = \\
&\qquad\qquad \textit{add}(x, \textit{remove}(p))
\end{aligned}$$

Aufgabe 2 - Klassendiagramm



Aufgabe 2 - Objektdiagramm



3.a) OCL-Signatur

$$
\begin{aligned}
\Sigma_{\Delta}^{OCL} &= (S_{\Delta}^{OCL}, \leq, F_{\Delta}^{OCL}) \\
S_{\Delta}^{OCL} &= Base^{OCL} \cup Class_{\Delta} \cup Coll_{\Delta}^{OCL} \\
F_{\Delta}^{OCL} &= Std_{\Delta}^{OCL} \cup Inst_{\Delta}^{OCL} \cup A_{\Delta} \\
Base^{OCL} &= \{OclAny, OclVoid, Real, Integer, Boolean, String, Null\} \\
Class_{\Delta} &= \{Airline, Flight, Person, Pilot, Steward\}
\end{aligned}
$$

3.a) OCL-Signatur (2)

$$
\begin{aligned}
Coll_{\Delta}^{OCL} &= \{Collection(OclAny), Set(OclAny), Sequence(OclAny), \\
&\qquad Bag(OclAny), \\
&\quad \ldots, \\
&\quad Collection(Null), \ldots, Bag(Null), \\
&\quad Collection(Airline), \ldots, Bag(Airline), \\
&\quad \ldots, \\
&\quad Collection(Steward), \ldots, Bag(Steward)\} \\
\leq &= \{Pilot \leq Person, Steward \leq Person, \ldots\} \\
Std_{\Delta}^{OCL} &= sieheFolien
\end{aligned}
$$

3.a) OCL-Signatur (3)

$$
\begin{aligned}
Inst_{\Delta}^{OCL} &= \{Airline.allInstances() :\rightarrow Set(Airline), \\
&\quad \ldots, \\
&\quad Steward.allInstances() :\rightarrow Set(Steward)\} \\
A_{\Delta} &= \{\_.name : Airline \rightarrow String, \\
&\quad \_.employees : Airline \rightarrow Set(Person), \\
&\quad \_.partners : Airline \rightarrow Set(Airline), \\
&\quad \_.flights : Airline \rightarrow Set(Flight), \\
&\quad \_.airline : Flight \rightarrow Airline, \\
&\quad \_number : Flight \rightarrow Integer, \\
&\quad \ldots\}
\end{aligned}
$$

3.b) Terme (i)

(i) den Grundpreis eines Flugs f

Sei $f : Flight \in X$, dann

$f.basicPrice \in T(\Sigma_{\Delta}^{OCL}, X)_{Real}$

Baumdarstellung	Präfix
	$.basicPrice(f)$

3.b) Terme (ii)

(ii) die Menge aller Flüge, die gemeinsam von einem Piloten p und einem Flugbegleiter s betreut werden

Infix:

$$\text{Sei} \quad p : \mathit{Pilot}, s : \mathit{Steward} \in X \quad \text{, dann}$$

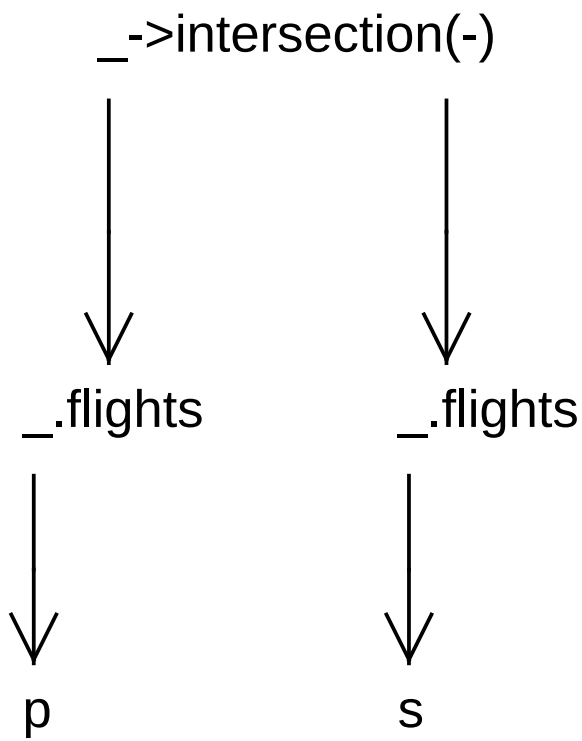
$$p.\mathit{flights} \rightarrow \mathit{intersection}(s.\mathit{flights}) \in T(\Sigma_{\Delta}^{OCL}, X)_{\mathit{Set}(\mathit{Flight})}$$

Präfix:

$$\rightarrow \mathit{intersection}(.\mathit{flights}(p), .\mathit{flights}(s))$$

3.b) Terme (ii) (2)

Baumdarstellung:



b) Terme (iii)

(iii) die Anzahl der Angestellten der Fluglinie, die einen Flug f durchführt
Infix:

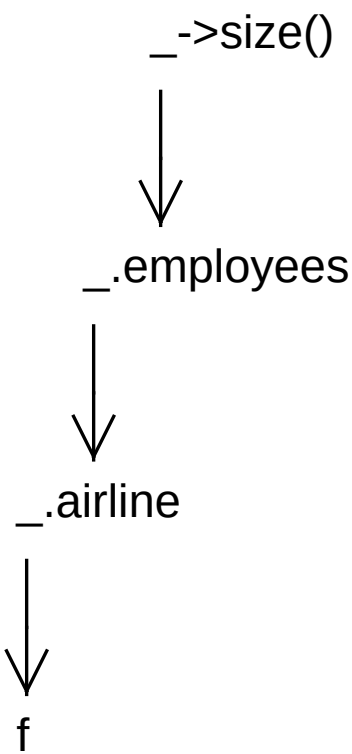
$$\text{Sei} \quad f : \mathit{Flight} \in X \quad \text{, dann}$$
$$f.\mathit{airline}.\mathit{employees} \rightarrow \mathit{size}() \ \in T(\Sigma_{\Delta}^{OCL}, X)_{\mathit{Integer}}$$

Präfix:

$$\rightarrow \mathit{size}(.\mathit{employees}(.\mathit{airline}(f)))$$

b) Terme (iii) (2)

Baumdarstellung:



b) Terme (iv)

(iv) ob eine Person p "‘Meier"’ heißt und Pilot ist
Infix:

$$\text{Sei} \quad p : Person \in X \quad \text{, dann}$$

$$p.name = \text{''}\, Meier\text{''}\ and\, p.oclIsTypeOf(Pilot)\ \in T(\Sigma_{\Delta}^{OCL}, X)_{Integer}$$

Präfix:

$$and(=(.name(p), \text{''}\, Meier\text{''}), .oclIsTypeOf(Pilot)(p))$$

b) Terme (iv) (2)

Baumdarstellung: