Kurs programowania

Wykład 9

Wojciech Macyna

Java Collections Framework (w C++ Standard Template Library)

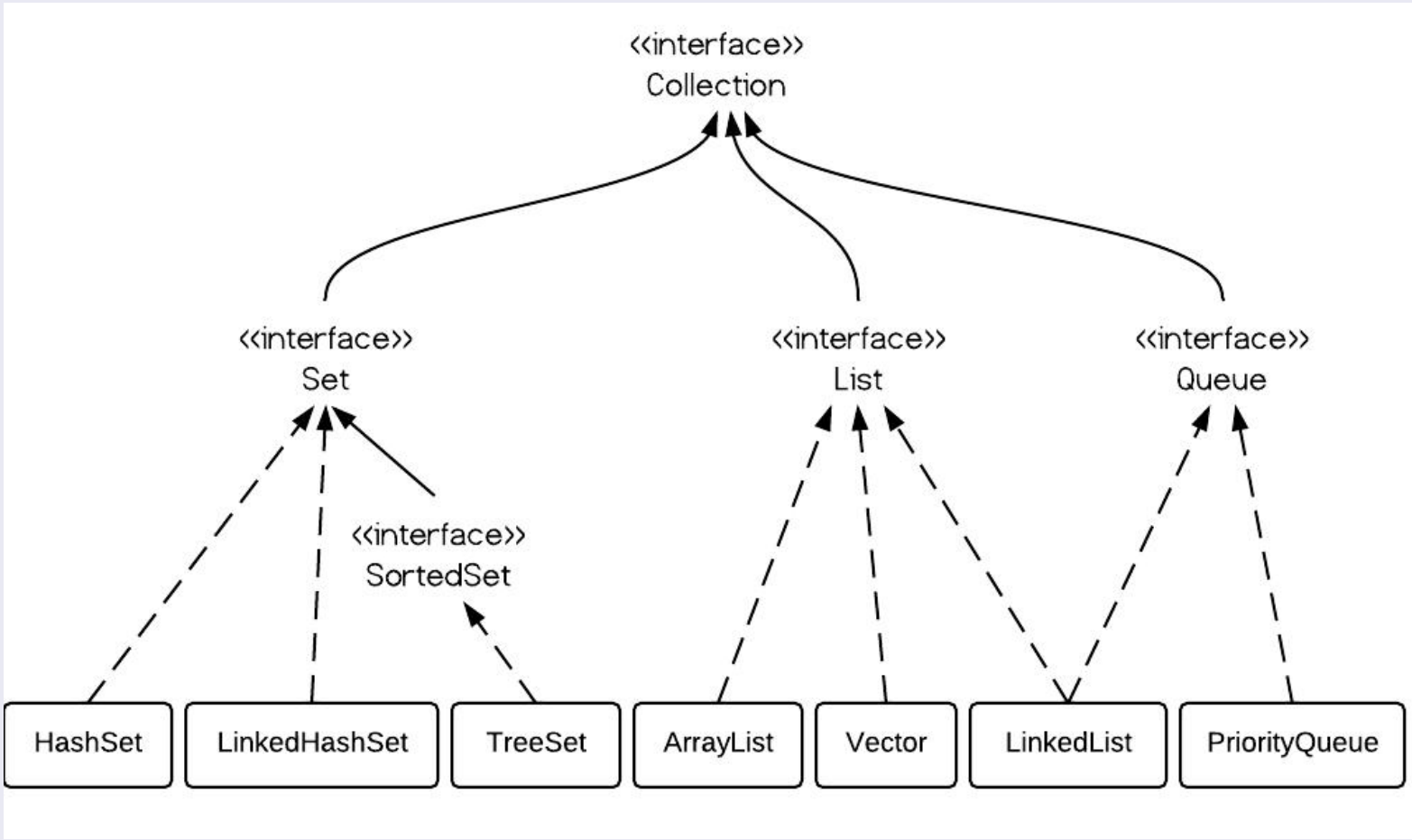
Kolekcja (kontener)

Obiekt grupujący/przechowujący jakieś elementy (obiekty lub wartości). Przykładami kolekcji są zbiór, lista czy wektor.

Składniki

- Interfejsy
- Implementacje (klasy)
- Algorytmy
- Iteratory

Hierarchia klas



Zbiory w Javie

Interfejs `java.util.Set` (`java.util.SortedSet`)

Zbiór elementów bez powtórzeń.

Przykładowe Implementacje

- `java.util.HashSet` (tablica haszująca, elementy nie są posortowane)
- `java.util.LinkedHashSet` (tablica haszująca, elementy połaczone ze sobą, elementy ułożone według kolejności wstawiania)
- `java.util.TreeSet` (automatycznie posortowane, implementacja algorytmu drzewa czerwono-czarneg0, logarytmiczny czas podstawowych operacji)

Klasa `java.util.Collections`

Zawiera pomocnicze metody statyczne do operowania na kolekcjach (zobacz także klasę `java.util.Arrays` dla tablic). Szczegółowy spis metod w dokumentacji.

Zbiory w Javie

Przykłady

`SetEx.java` – przykład użycia `HashSet i LinkedHashSet`

`SortedSetEx.java` – przykład użycia `TreeSet`

Listy w Javie i cpp

Przykłady

`ListEx.java` – przykład użycia `ArrayList`

`SortedListEx.java` – przykład użycia `LinkedList`

`PriorityQueueTest.java` – kolejka priorytetowa (`PriorityQueue`)

`lista.cpp` – lista w c++ (`list`)

`kolejka.cpp` – kolejka w c++ (`queue`)

`kolpr.cpp, kolpr2.cpp` – kolejka priorytetowa w c++ (`priority_queue`)

Przykład - Mapy w Javie

Interfejs `java.util.Map` (`java.util.SortedMap`)

Kolekcja przechowująca pary *klucz-wartość*, inaczej tablica asocjacyjna.

Przykładowe Implementacje

- `java.util.HashMap`
- `java.util.SortedMap` (klucze uporządkowane)
- `java.util.LinkedHashMap`
- `java.util.TreeMap` (automatycznie posortowane, implementacja oparta na drzwach binarnych)

Przykłady

`HashMapEx.java` – przykład użycia `HashMap`

`MapEx.java` – przykład użycia `TreeMap`

Standard template library (STL)

Struktury STL

- `stack` stos
- `queue` kolejka
- `priority queue` kolejka priorytetowa
- `list` lista
- `vector` tablica
- `deque` tablica dwukierunkowa

Typy uogólnione w Javie

W Javie typem uogólnionym może być tylko klasa (stąd typ `Integer` zamiast `int`). Jednak dzięki automatycznemu konwertowaniu nie jest to uciążliwe.

Pola typów uogólnionych przechowują tylko referencje do tych typów a nie kopie obiektów (w Javie nie ma niejawnego kopiowania).

Typem nie jest klasa parametryzowana ale dopiero jej ukonkretnienie z podanymi konkretnymi parametrami.

Klasy parametryzowane typami

W językach obiektowych można definiować klasy uogólnione zawierające pola, których typy są parametrami, tzw. szablony klas (np. klasy z JCF czy STL).

Parametry typów podaje się w nawiasach (<>) po nazwie klasy, odzielając je od siebie przecinkami.

Dodanie specjalnych własności typu podanego jako parametr

Jeśli potrzebujemy aby nasza klasa podawana jako parametr posiadała dodatkowe metody możemy to w Javie uzyskać dodając jaką klasę (interfejs) nasza klasa ma dziedziczyć.

Na przykład jeśli klasa użyta jako parametr powinna mieć porządek liniowy to powinna dziedziczyć interfejs `Comparable` z metodą `compareTo`. Deklaracja takiego interfejsu (jest już w języku Java) wygląda nastąpująco:

```
1 public interface Comparable<T>{
2   public int compareTo(T o);
3 }
```

Typy uogólnione java i cpp

Przykłady

`Stack.java, StackTest.java` – przykład implementacji stosu w Javie

`stos.cpp` – przykład implementacji stosu w cpp

`Tree.java, TreeTest.java, TreePairTest.java` – przykład implementacji drzewa binarnego w Javie

Podsumowanie

Realizacja mechanizmu typów uogólnionych w Javie wystarcza do większości zastosowań – pewne niedogodności wynikają z braku tego mechanizmu w pierwszych wersjach języka.

Rozwiązania przyjęte w C++ są bardziej elastyczne (umożliwiają posługiwanie się niezdefiniowanymi metodami przy pisaniu schematów czego poprawność jest sprawdzana dopiero podczas kompilacji konkretnej instancji szablonu). W Javie trzeba jawnie podać wymagania danej klasy będącej parametrem.