**TreeSet**

TreeSet w Javie jest implementacją interfejsu Set, która przechowuje unikalne elementy w porządku posortowanym. Oto zalety i wady korzystania z TreeSet:

Zalety:

1. **Automatyczne sortowanie:** Elementy przechowywane w TreeSet są automatycznie sortowane w porządku rosnącym lub malejącym w zależności od zastosowanego komparatora lub naturalnego porządku elementów.
2. **Optymalna implementacja zapytań:** TreeSet używa drzewa czerwono-czarnego do przechowywania elementów, co zapewnia efektywne dodawanie, usuwanie i wyszukiwanie elementów. Operacje takie jak add, remove i contains mają złożoność czasową logarytmiczną (O(log n)) - co sprawia, że są one efektywne dla dużych zbiorów danych.
3. **Unikalność elementów:** TreeSet gwarantuje, że przechowywane elementy są unikalne. Jeśli próba dodania duplikatu zostanie wykonana, operacja ta zostanie zignorowana.

Wady:

1. **Złożoność pamięciowa**: Implementacja drzewa czerwono-czarnego używana przez TreeSet zajmuje więcej pamięci niż inne struktury danych, takie jak HashSet. Każdy element musi przechowywać dodatkowe informacje o strukturze drzewa, co może prowadzić do większego zużycia pamięci.
2. **Wolniejsze operacje wstawiania**: Wstawianie elementów do TreeSet może być nieco wolniejsze niż wstawianie ich do HashSet, szczególnie w przypadku dużych zbiorów danych. Jest to spowodowane koniecznością utrzymywania porządku elementów w drzewie.
3. **Brak równoległej obsługi**: TreeSet nie jest w pełni wspierane przez mechanizmy równoległego przetwarzania w Javie, co może ograniczać wydajność w niektórych przypadkach, zwłaszcza w aplikacjach, które wymagają intensywnego przetwarzania równoległego.

Podsumowując, TreeSet jest przydatnym narzędziem do przechowywania unikalnych, posortowanych elementów, ale należy pamiętać o jego ograniczeniach, takich jak większe zużycie pamięci i nieco wolniejsze operacje wstawiania w porównaniu z innymi implementacjami Set.