

ISP Workshop

Das Interface Segregation Principle

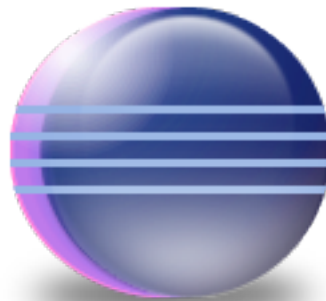
- Warum ist die Trennung von Schnittstellen sinnvoll?
- Ein Beispiels mit `java.util.Queue`
- Analyse des Beispiels mit `java.util.Queue`
- Entwurf mit getrennten Schnittstellen
- Refactoring des Beispiels ohne `java.util.Queue`
- Bewertung der Lösung
- Ausblick und Fragen

Warum ist die Trennung von Schnittstellen sinnvoll?

- Die Schnittstelle ist unübersichtlich (fat)
- Die Methoden haben keinen Zusammenhalt (cohesion)
- Die Schnittstelle erfüllt mehr als einen Zweck (single responsibility)
- Gruppen von Methoden für Klassen von Nutzern sind nicht erkennbar
- Es entstehen unnötige Abhängigkeiten bei Nutzern (dependency)
- Wrapper und Adapter werden auch unübersichtlich (fat)
- Instabilität ist wahrscheinlicher (instability)

Ein Beispiel mit `java.util.Queue`

- Ein Producer erzeugt Meldungen
- Ein Consumer verarbeitet Meldungen
- Das Programm verbindet die beiden mittels einer Queue
- Consumer und Producer arbeiten parallel
- Das Programm gibt nach 1 Sekunde den Stand aus

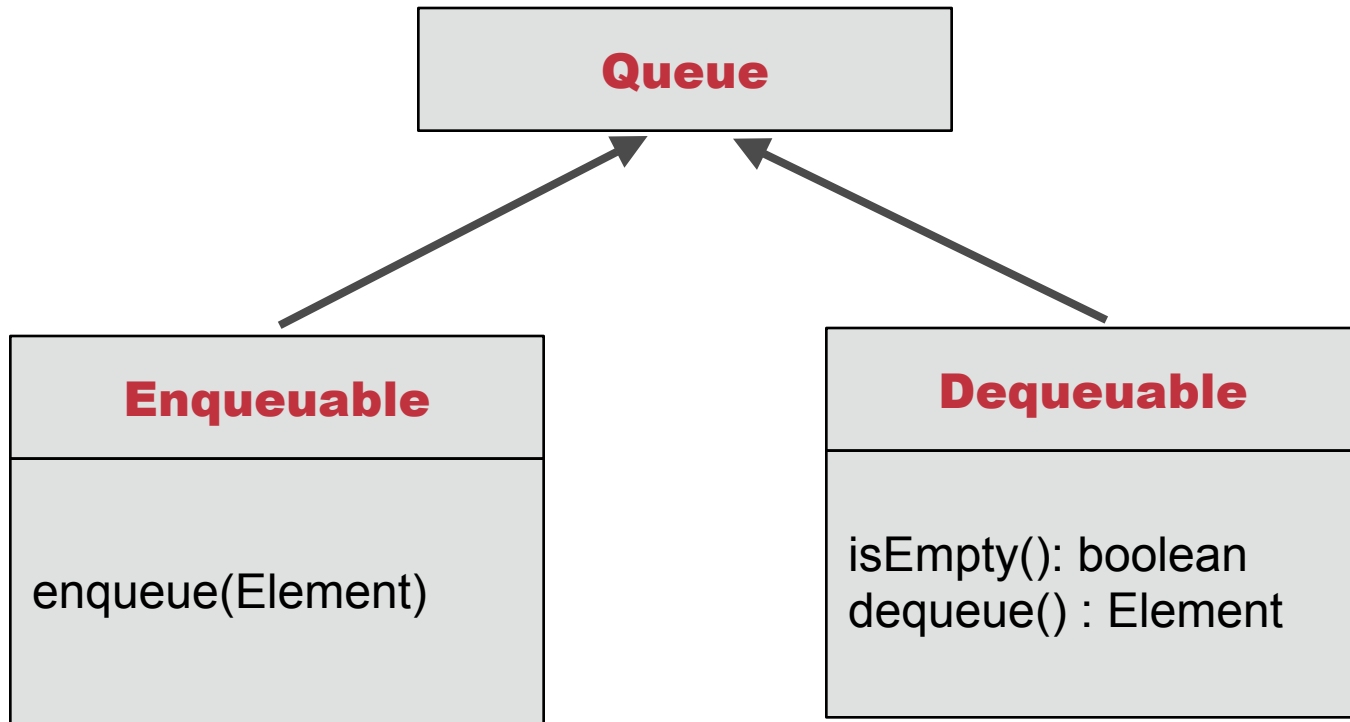


3 TODOs lösen

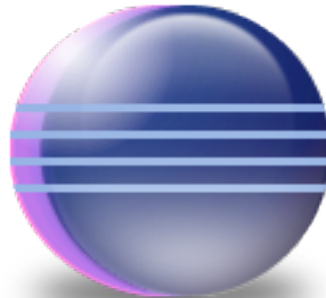
Analyse des Beispiels mit `java.util.Queue`

- Queue erbt von Collection und hat ca. 20 Methoden
 - `add(Element)`, `addAll()` oder `offer(Element)`?
 - `element()`, `remove()`, `poll()`, `peek()` oder gar `remove(peek())`?
- Keine Kontrolle über die Funktionalität durch die Schnittstelle
 - Der Producer könnte auch wieder Elemente entfernen
 - Der Consumer könnte auch `clear()` aufrufen
- Code Completion der IDE ist fast nutzlos
 - zu viele Möglichkeiten

Entwurf mit getrennten Schnittstellen



Refactoring des Beispiels ohne `java.util.Queue`



- Anlegen getrennter Schnittstellen
- neue Queue als Adapter auf `LinkedList` implementieren
- getrennte Schnittstellen in `Consumer` und `Producer` verwenden

Bewertung der Lösung

- Der neue Code sieht fast identisch aus
- Getrennte Schnittstellen verhindern Fehlbenutzung
- Die Entwicklung mit getrennten Schnittstellen ist einfacher
- Durch Adapter können auch Fat-Interfaces getrennt werden

Ausblick und Fragen

- mehr Adapter verwenden?
- Collection Framework ersetzen (deCollect)?
- ...