Inversion of Control
Dependency Injection

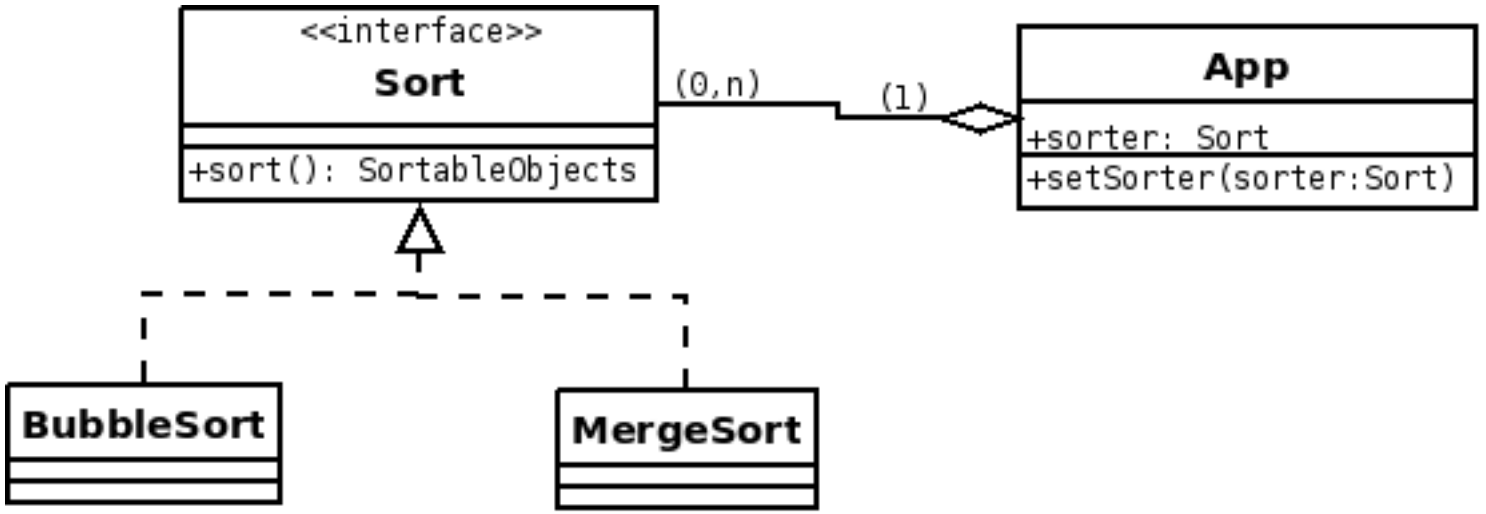
Florian Motlik 13.10.2008
JSUG - Treffen 1 WS 08/09

Inversion of Control

- Don't call us, we call you.
- Trennung Kontrolle und Ausführung
- Generische Kontrollklasse und spezifische Implementierung
- Iterator vs Code Block/Closure
  stooges = ['Larry', 'Curly', 'Moe']
  - foreach (stooge in stooges) {print stooge + "\n"}
    - Kontrolle der Iterierung in Hand des Programmierers
  - stooges.each {|stooge| print stooge + "\n"}
    - each Methode kontrolliert Iterierung. Interne Iterierung.
- Event-driven programming - z.B. Swing Gui Listener

Dependency Injection

- Anwendung von IOC
- Kontrolle über Dependencies



- Kopplung von App zu SortImpl verringern
- Kein direkter Aufruf eines SortImpl Konstrukturs
- Sorter wird "Injected" in App durch Framework
- Setter & Constructor based Injection

Vorteile/Nachteile

- Vorteile
  - Weniger Abhängigkeiten (Kapselung)
  - Kein lookup Code nötig
  - Zwang zu Interfaces -> bessere Struktur
  - Test/Run Konfiguration möglich und nur einmal zu machen.
- Nachteile
  - Möglicherweise viel Konfiguration (XML Hell)
  - Zusätzliches Framework
  - Abhängigkeiten zum Framework (vermeidbar)
  - Refaktorisierung umständlich/fehleranfällig (IDE)
  - Weniger Code Transparenz

Links

- http://en.wikipedia.org/wiki/Dependency_injection
- http://en.wikipedia.org/wiki/Inversion_of_control
- http://en.wikipedia.org/wiki/Strategy_pattern
- http://en.wikipedia.org/wiki/Closure_(computer_science)
- http://martinfowler.com/articles/injection.html

Licensed under CreativeCommons Attribution + ShareAlike (by-sa)
http://creativecommons.org/licenses/by-sa/3.0/at/