Übung 7

# Aufgabe Modularer Entwurf

## 1. Schritt: Welche Module gibt es?

Entscheidung gemäss Entwurfsentscheidungen:

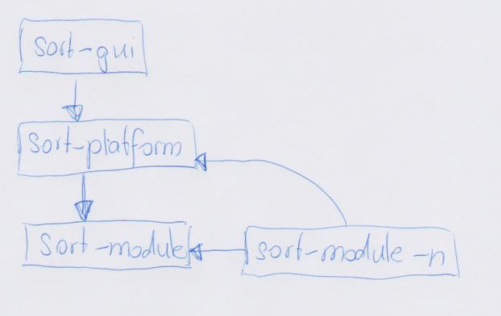
1. Es kann sein, dass später ein zweites GUI oder eine webbasierte Oberfläche hinzukommt. Für uns heisst das, das GUI muss entkoppelt werden 🡪 Modul GUI und Modul mit der eigentlichen Logik
2. Es kann sein, dass später neue Sortieralgorithmen hinzukommen. Für uns heisst das, dass es ein Basismodul Algorithmus und n weitere Module Algorithmen geben muss, respektive geben kann 🡪 Modul Algorithmus und für jede Implementierung ein eigenes Modul

Daraus kann man folgende Module erstellen:

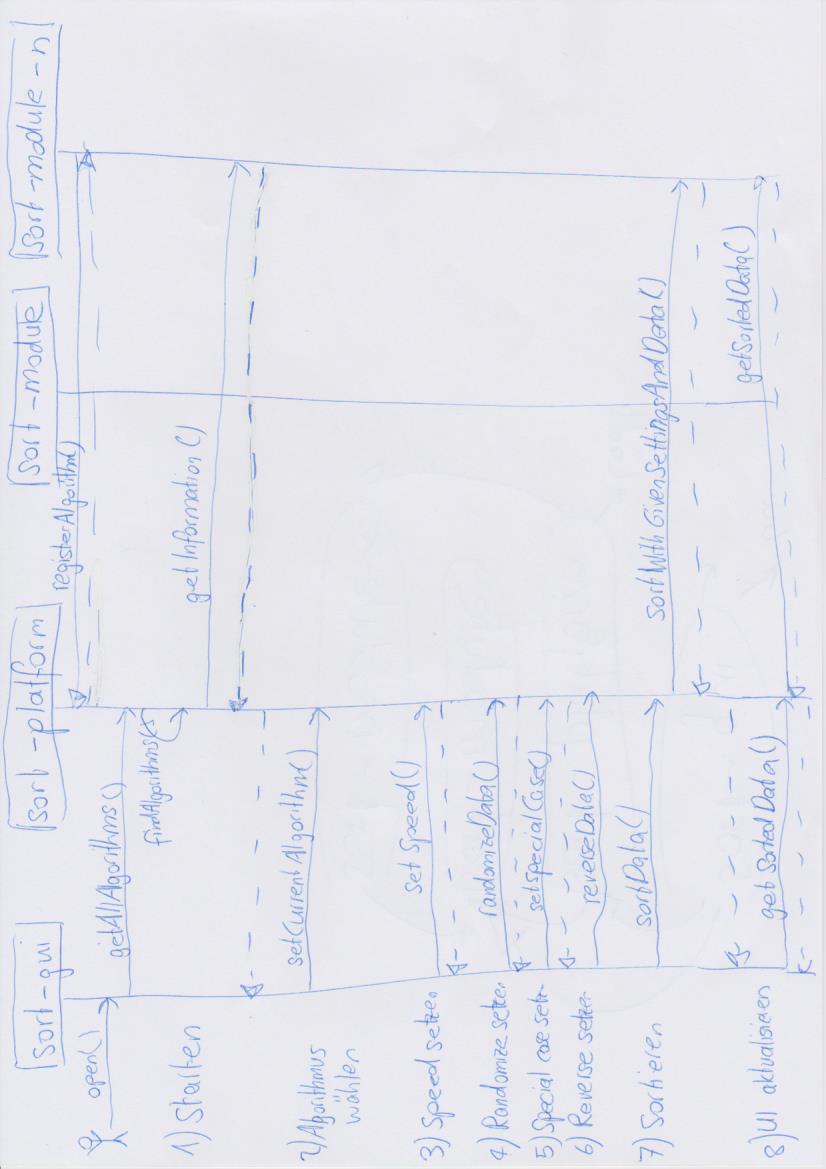
* sort-gui: Enthält das GUI zum Verwalten der Sortiersimulation
* sort-platform: Lädt alle Sortieralgorithmen und stellt einem Anwender (Hier sort-gui) eine definierte Funktionalität zur Verfügung. Zudem ermöglicht es Sortieralgorithmen aus den Modulen sort-module-n, sich in einer Registry anzumelden
* sort-module: Enthält die Schnittstelle zum Implementieren eines Sortieralgorithmus, welcher dann von sort-platform geladen werden kann
* sort-module-n: Enthält eine konkrete Implementierung eines Sortieralgorithmus, z.B. könnte es sort-module-bubble geben

## 2. Schritt: Welche Module sind von welchen anderen abhängig?

Die Abhängigkeiten sehen wie nachfolgend aus. Zu erwähnen ist, dass die zusätzliche Abhängigkeit zwischen sort-platform und den jeweiligen Modulen im Logical View noch eingezeichnet ist (Im Development View fehlt diese natürlich!)



## 3. Schritt: Wie genau arbeiten die Module zusammen?



## 4. Schritt: Wie können beim Systemstart die nötigen Laufzeitenbeziehungen etabliert werden?

Ohne OSGi genauer zu kennen fallt mir folgende Möglichkeit ein:

* Da die Sortiermodule sort-module-n von sort-platform abhängig sind, wird sort-platform zuerst geladen
* Nachdem sort-platform geladen ist, werden alle sort-module-n geladen, welche sich im Activator in einer Registry in sort-platform registrieren können. Das Modul sort-platform verfügt dann über eine Liste mit verfügbaren Sortieralgorithmen
* Das Modul sort-gui kann problemlos alle Sortieralgorithmen von sort-platform abfragen

## 5. Schritt: Wie sehen die Schnittstellen der einzelnen Module aus?

Die Schnittstellen könnte analog zum Flussdiagramm aussehen:

* sort-gui
  + Startet die grafische Applikation
  + Verwaltet alle Benutzereingaben und leitet sie an sort-platform weiter
* sort-platform
  + registerAlgorithm(): Ermöglicht es, einen Sortieralgorithmus in der Registry zu registrieren
  + getAllAlgorithms(): Liefert alle Sortieralgorithmen zurück
  + setCurrentAlgorithm(): Setzt den aktuellen Sortieralgorithmus
  + setSpeed(): Setzt die Geschwindigkeit beim Sortieren
  + setSpecialCase(): Setzt einen Spezialfall
  + randomizeData(): Verwürfelt alle Daten zufällig
  + reverseData(): Kehrt die Daten um
  + sortData(): Sortiert die Daten mit dem aktuell gesetzten Sortieralgorithmus und Einstellungen
  + getSortedData(): Liefert die sortierten Daten zurück
* sort-module
  + getInformation(): Gibt Information zum Namen des Sortieralgorithmus, seiner Komplexität und Laufzeitverhalten zurück
  + sortWithSettingsAndData(): Sortiert Daten mit bestimmten Einstellungen (z.B. Sortiergeschwindigkeit)
  + getSortedData(): Gibt die sortierten Daten zurück, kann analog auch in der obigen Methode passieren
* sort-module-n
  + Implementiert die Schnittstelle aus sort-module und implementiert ein konkretes Verhalten