

Hinweise und Vorgaben zu den Programmieraufgaben

Vorbereitung

Je nachdem, ob Sie eclipse (oder eine andere IDE) verwenden **oder** keine IDE, sondern einen Editor Ihrer Wahl verwenden und Java von der Konsole starten:

- Legen Sie (in einem Verzeichnis Ihrer Wahl) einen Ordner für die Prog2-Übungen an.
 - Falls Sie eclipse verwenden:
Definieren Sie diesen Ordner als eclipse-„workspace“ und legen Sie **für jede Programmieraufgabe** ein neues Java-Projekt (**kein Paket!**) an.
 - Falls Sie die Konsole benutzen:
Legen Sie in diesem Ordner für jede Programmieraufgabe einen neuen Unterordner an.
- Das Projekt bzw der Unterordner kann zB den Titel UxxAy haben, wobei xx für die Nummer des Übungsblattes und y für die Nummer der Aufgabe stehen.
- Legen Sie in diesem Projekt bzw Unterordner alle Java-Quellcode-Dateien an, die für die Aufgabe benötigt werden.
- Falls eine Datei (bzw eine Klasse) für mehrere Aufgaben benötigt wird oder in einer späteren Aufgabe modifiziert werden soll, so kopieren Sie die Datei (per copy & paste) in das neue Projekt bzw den neuen Unterordner.
- Verwenden Sie **keine Java-Pakete!**

Bearbeitung der Aufgaben

- Trennen Sie zwischen Problemlösung und Test! Das bedeutet, dass Sie zu jeder Aufgabe i.d.R. **mindestens zwei** Java-Klassen anlegen:
 - *Nutzklasse(n)* : alle Klassen (und Interfaces), die durch die Aufgabenstellung definiert werden - die eigentliche „Applikation“.
 - *JUnit-Test-Klasse*: Spezielle Klasse, die ausschließlich Test-Methoden (JUnit-Tests) enthält.
 - Die Methoden der Nutzklassen produzieren fast nie eine Bildschirm-Ausgabe. (Ausnahmen werden explizit angegeben!) Sie sollten allerdings - auch wenn es nicht ausdrücklich verlangt ist - immer die Methode `toString()` überschreiben (ebenso: `equals()`).
 - Eine *Main-Klasse* mit `main`-Methode ist nur in wenigen Fällen nötig. Falls Sie unbedingt Ergebnisse *sehen* möchten, dann definieren Sie spezielle Hilfsmethoden, die ausschließlich der Bildschirm-Ausgabe dienen.
- Verwenden Sie nur solche programmiersprachlichen Mittel, die bis zum jeweiligen Stand in der Vorlesung vorgestellt wurden. Das sind zB für das erste Übungsblatt:
 - Definition von Klassen, ggf Unterklassen, Abstrakten Klassen, Interfaces
 - Erzeugen von Objekten
 - Definition und Aufruf (statischer und nicht-statischer) Methoden
 - Schleifen (aber keine for-each-Konstruktion), Verzweigungen, Arrays
- Verwenden Sie *keine* Klassen der Java-API, die eines der Interfaces `Set`, `List`, `Queue` o.ä. implementieren, also *keine* `ArrayList`, `HashSet`, `Vector` usw.
- Genau das wollen wir lernen *selber zu implementieren!*

JUnit-Tests

Testen Sie Ihre Lösungen ausführlich! Nutzen Sie dazu unbedingt das framework JUnit. Zu einigen Aufgaben werden von uns Tests bereitgestellt. Arbeiten Sie erst dann weiter, wenn diese Tests erfolgreich durchlaufen wurden. Erstellen Sie selber weitere Tests, tauschen Sie Tests mit Ihren Kommilitonen aus, dh: testen Sie Ihre Applikation mit „fremden“ Tests!

UML-Diagramme

Wir werden häufig UML-(ähnliche) Beschreibungen nutzen, um die Klassenstruktur der Projekte zu beschreiben. Modifier, Attribute und Konstruktoren sowie ererbte Methoden werden dort der Übersichtlichkeit halber aber nicht dargestellt. Sie beschreiben also nur die „Mindest-Anforderungen“ der Klassen. Wenn nichts andere ausdrücklich genannt ist, sind die in den Diagrammen angegebenen Komponenten alle als `public` zu verstehen.

Alles, was Sie zur Implementierung an weiteren Attributen, Hilfsmethoden, Hilfsklassen, ... benötigen, sollten Sie wenn möglich als `private` deklarieren.