

Übungsblatt 3

1. Was sind die drei Kern-Konzepte der objektorientierten Programmierung?
2. Was ist das Substitutions-Prinzip?
3. Was ist der Unterschied zwischen “Subtyping”, “Inheritance” und “Subclassing“?
4. Welche Regeln gelten beim “Behavioral Subtyping“?
5. Welche Formen der Wiederverwendung im Kleinen gibt es? Beschreibe sie kurz.
6. Beschreibe die Eigenschaften einer `Set` und einer `BoundedSet` Klasse mittels Java-Code und Assert-Statements. Keine Implementierung ist notwendig, lediglich die Schnittstellen und Verträge sollen erarbeitet werden. Sehe mindestens die Methoden `add`, `remove`, `isElem` und `size` vor.
7. In der zweiten Übungs-Stunde haben wir gesehen, dass ein `Integer[]` ein Subtyp von `Object[]` ist. Wir wollen nun zwei Klassen `IntegerList` und `ObjectList` implementieren, die jeweils eine Liste von Objekten des entsprechenden Typs verwalten. Soll `IntegerList` ein Subtyp von `ObjectList` sein? Oder andersherum? Wie sehen die Verträge der beiden `insert` Methoden aus, wenn man die erlaubten Parameter-Typen im Vertrag beschreibt? Falls du ein Problem siehst, gib Beispielcode an.
8. Aufbauend auf Übung 7: Es gebe eine generische Klasse `List<T>`, die eine Liste eines beliebigen Typs verwaltet. Soll eine `List<Integer>` ein Subtyp von `List<Object>` sein? Wie sieht die Antwort für Java aus? Wie sieht die Antwort für Eiffel aus? Beschreibe Probleme die du dabei siehst.