

## Exercise 3

1. What are the three core concepts of Object-Oriented programming?
2. What is the “Substitution Principle“?
3. What is the difference between “Subtyping”, “Inheritance” and “Subclassing“?
4. Which rules must be applied to conform to “Behavioral Subtyping“?
5. Which types of reuse in the small are there? Describe them.
6. Describe the properties of a `Set` and a `BoundedSet` using Java assert-statements. No implementation is needed, just the interface and contracts should be developed. Include the methods `add`, `remove`, `isElement` and `size`.
7. In the second exercise lesson we have learned that an `Integer[]` is a subtype of `Object[]`. The next step is implementing the two classes `IntegerList` and `ObjectList`. Both of them managed a list of objects of the corresponding type. Should `IntegerList` be a subtype of `ObjectList`? Or vice versa? What is the contract between both `insert` methods? If you see any problems, provide an example in Java.
8. Extending question 7; assume there is a generic class `List<T>` which manages a list of an arbitrary type `T`. Should `List<Integer>` be a subtype of `List<Object>`? What is the answer in Java? What is the answer in Eiffel? Describe the potential issues.