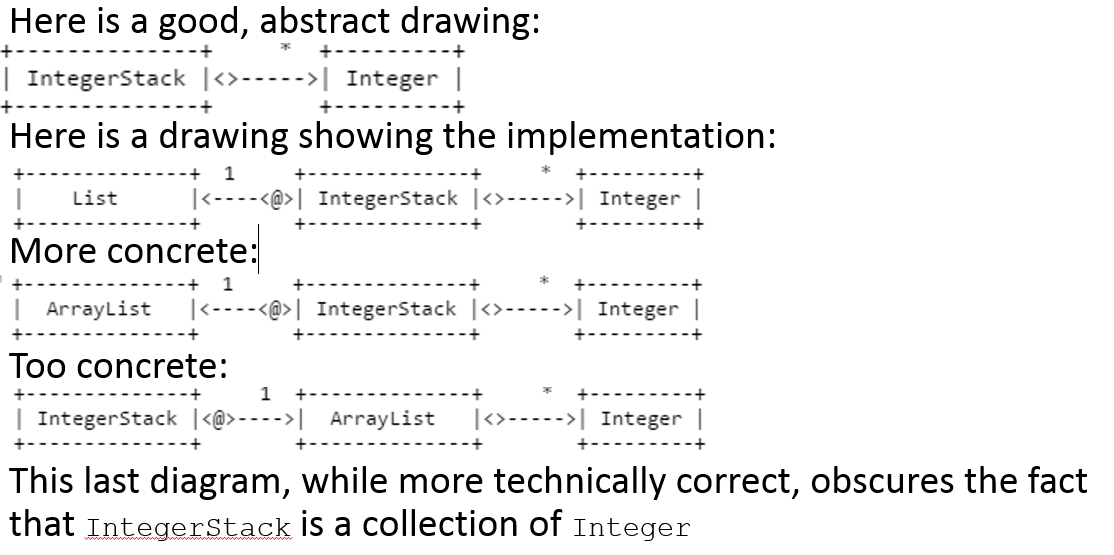
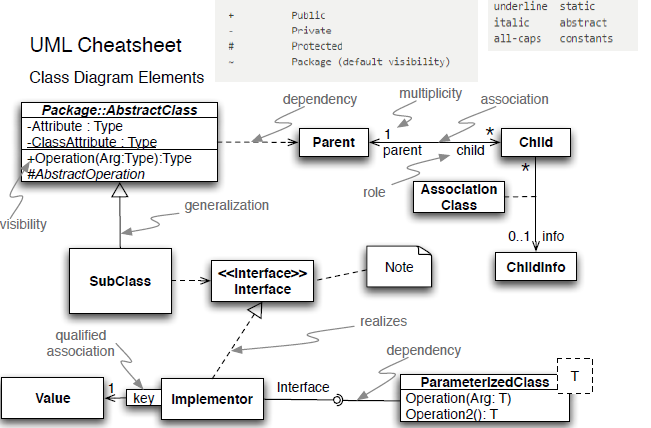


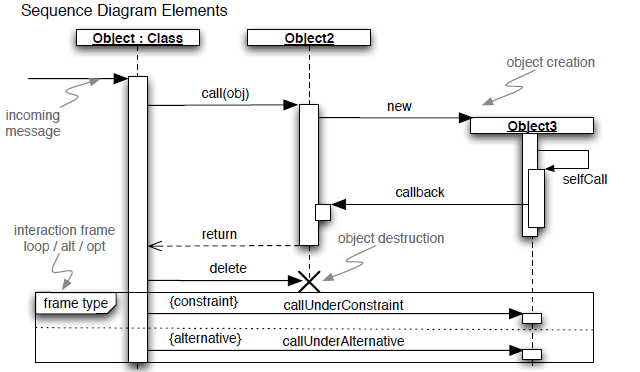
Collection classes

A *collection* may hold references to many other objects.

Usually the objects are of some uniform type, eg, a list of dogs, a set of colors







Interface

* Type of an object. Defines behaviors the object implements

Class

* Definition of an object, including behaviors and state (fields)

Constructor

* Code that runs when an object is initialized to setup the object’s initial state

Delegation

* Allowing some other object to provide the behavior (e.g. lambdas)

Subclassing

* Allowing some other class to provide behavior (e.g. abstract classes, interface implementations)

Immutable Data Class

* Object whose state does not change (e.g. String)

Mutable Data Class

* Object whose state can change

Collection Class

* Encapsulates a collection of objects. Can be either immutable or mutable

MVC

* IO logic (View) is separate from business logic (Model).

Static Class

* Class that does not have an instantiable state, defines only global fields and methods

Static Factory

* Static class whose purpose is to generate instances of some concrete interface implementations. The interface and factory are public, but the interface implementations are not

Builder

* A mutable object used to build an immutable object. Add pieces to the builder bit by bit, then convert into an immutable object

Command

* Encapsulate a message in an object.

Strategy

* Encapsulate behavior into it’s own object. Receiving objects delegate behavior to a method parameter or field