Prototype

1. Motivation
   1. Complicated objects (e.g., cars) aren't designed from scratch
      1. They reiterate existing designs
   2. An existing (partially or fully constructed) design is a Prototype
   3. We make a copy (clone) the prototype and customize it
      1. Requires deep copy support
   4. We make cloning convenient (e.g., via a Factory)
   5. Prototype: A partially or fully initialized object that you copy (clone) and make use of.
2. Don’t use cloneable
   1. Shallow copy: Person jane = john; (copy references)
   2. Deep copy: Replicated every single element.
   3. Typically cloneable implementation is not recommended. It doesn't specify whether it uses deep copy or shallow copy.
3. Copy constructor
   1. Take The object in the constructor. () eg: Address(Address a){...}
4. Copy Through Serialization
   1. Implement Seriazliable and use SerializationUtils.roundTrip();
   2. Can be used if we have a lot of fields and hierarchies.
5. Summary
   1. To implement a prototype, partially construct an object and store it somewhere.
   2. Clone the prototype
      1. Implement your own deep copy functionality; or
      2. Serialize and deserialize (They traverse the class graph)
   3. Customize the resulting instance.