

Day 9: Creational Patterns Overview

Design pattern categories

Five design pattern categories:

1. Creational
2. Structural
3. Behavioral
4. Functional
5. Scala-specific design patterns

The first three categories cover the infamous **Gang of Four** design patterns.

Creational patterns:

- `abstract factory pattern` — one layer of abstraction above factories, i.e. put a common interface to your factories and initialize factories using that.
- `factory method pattern` — instead of calling a concrete constructor, just call a method on a factory that (depending on specific runtime info like types) will give you the object you need
- `lazy initialization pattern` — delay evaluation until needed. Much easier in Scala.
- `singleton pattern` — one object, globally available. Again, easier in Scala.
- `object pool pattern` — preallocate objects of a given type, using and deallocating as needed. Think database connections.
- `builder pattern` — if you've got many permutations of constructor parameters, the builder pattern helps.
- `prototype pattern` — use `clone()` to create an object from something else.