Gama Categorization

1. Design patterns are typically split intp three categories.
2. This is called Gama Categorization after Erich Gamma.
   1. Creational
      1. Deal with the creation(construction) of objects
      2. Explicit(constructor) vs implicit (DI, reflection, etc.)
      3. Wholesome (single statement) vs piecewise (step-by-step)
   2. Structural
      1. Concerned with the structure (eg. class members)
      2. Many patterns are wrappers that mimic the underlying class interface
      3. Stress the importance of good API design
   3. Behavioral
      1. They are all different; no central theme

Builder Pattern - When construction becomes a little too complicated

1. Some objects are simple and can be created in a single constructor call
2. Other objects require a lot of ceremony to create
3. Having an object with 10 constructor arguments is not productive
4. Instead, opt for piecewise construction
5. **Builder provides an API for constructing an object step-by-step**
   1. When piecewise object construction is complicated, provide an API for doing it succinctly.

Builders in java

1. StringBuilder
   1. StringBuilder sp = new StringBuilder();
   2. sp.append(“...”);
   3. This builds up your string

Builder

1. Example code written with HtmlElement and HtmlBuilder.

Fulient Builder

1. Method returns the same type so we can create like a chain
   1. Ex: sb.append(“hey”).append(“erhe)......

Fluent Builder Inheritance with java generics

1. Example written with personbuilder and employeebuilder
2. clasName<t extends className<T>>
3. Return cast self like (T)this; and override in classes that extends the main one.

Faceted Builder

1. Code example written With employee..

Summary

1. A builder is a separate component for building an object.
2. Can either give builder a constructor or return it via a static function
3. To make the builder fluent, return this.
4. Different facets of an object can be built with different builders working in tandem via a base class.