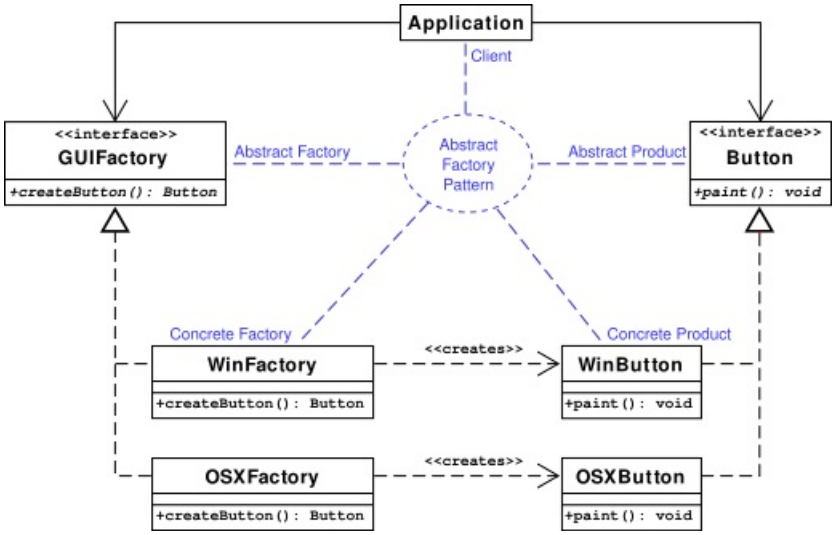
Creation

General:

- deal with object creation mechanisms, trying to create objects in a manner suitable to the situation
- These design patterns are all about class instantiation
- This pattern can be further divided into class
- creation patterns and object
- creational patterns
- While class
- creation patterns use inheritance effectively in the instantiation process, object
- creation patterns use delegation effectively to get the job done
- The creational patterns aim to separate a system from how its objects are created, composed, and represented
- They increase the system's flexibility in terms of the what, who, how, and when of object creation

Abstract Factory:

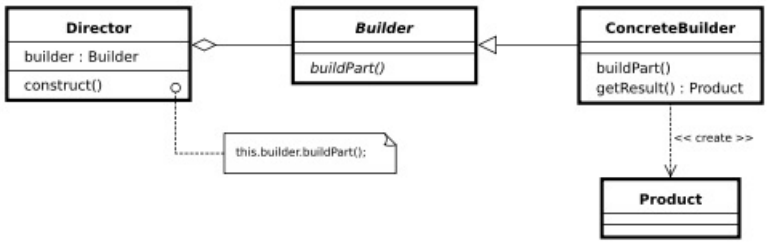
- Definition/Use
  - provides an interface for creating related or dependent objects without specifying the objects' concrete classes
  - provide an interface for creating families of related or dependent objects without specifying their concrete classes
  - encapsulates "new" ex: new product()
  - determines concrete type but returns abstract pointer
    - client code has no knowledge and isn't burdened by concrete type
    - adding new concrete types done by modifying client code to use different factory (1 line)
  - can determine concrete type from config file for example
- Structure



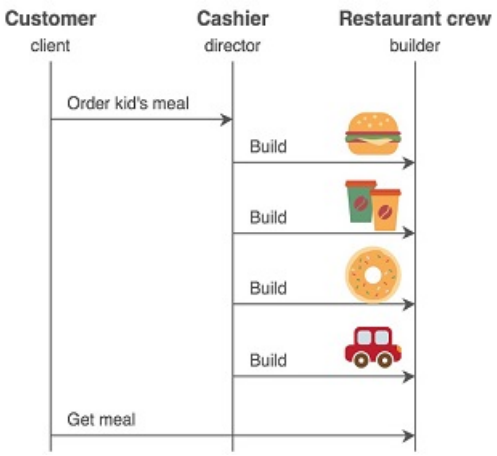
–

Builder:

- Definition/Use
  - separates the construction of a complex object from its representation so that the same construction process can create different representations
  - builder pattern is useful to avoid a huge list of constructors for a class
  - an application needs to create the elements of a complex aggregate
  - use builder to store parameters and then use that builder in constructor
  - separate the construction of a complex object from its representation so same construction can create different representations
- Structure



–

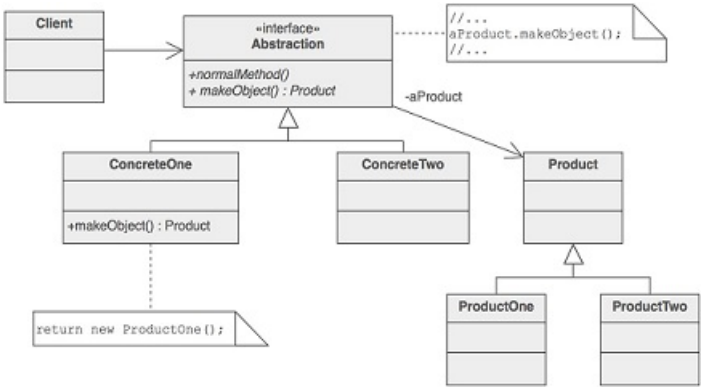


- Example
  - StringBuffer and StringBuilder
- Notes
  - put the builder term in the name of the builder class to indicate the use of the pattern to the other developers
  - if the target class contains flat data, your builder class can be constructed as a Composite that implements the Interpreter pattern

---

Factory Method:

- Definition/Use
  - allows a class to defer instantiation to subclasses
  - new operator considered harmful
  - define an interface for creating an object, but let subclasses decide which class to instantiate
  - provide a way for users to retrieve an instance with a known compile-time type, but whose runtime type may actually be different
  - an increasingly popular definition of factory method is: a static method of a class that returns an object of that class' type
    - unlike a constructor, the actual object it returns might be an instance of a subclass
- Structure

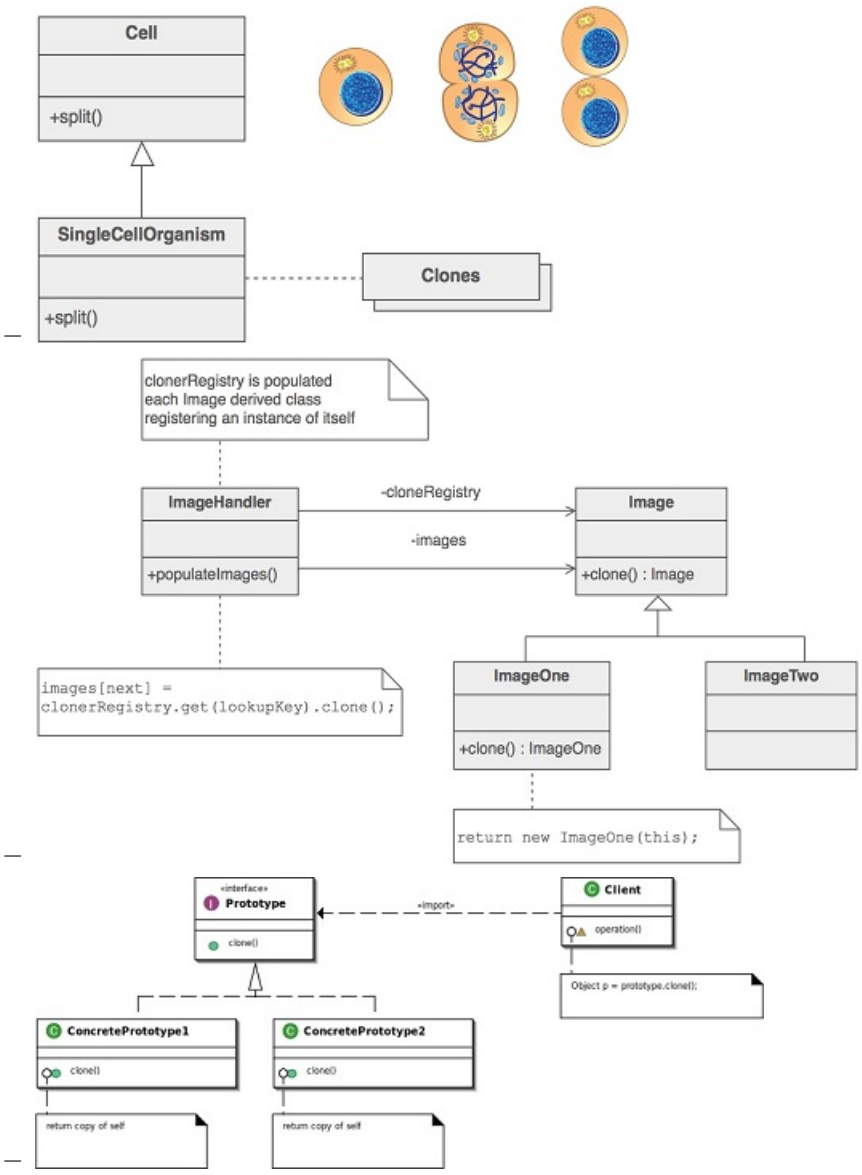


–

- Example
  - a factory method that is supposed to return an instance of the class Foo may return an instance of the class Foo, or an instance of the class Bar, so long as Bar inherits from Foo
  –
  ```
  Color.make_RGB_color(float red, float green, float blue)
  Color.make_HSB_color(float hue, float saturation, float brightness)
  ```
  –
  ```
  Letter.getLetter(char) if vowel return Vowel(char) else Consonant(char)
      (vowel, consonant extend letter)
  ```
- Notes
  - consider making all constructors private or protected

Prototype:

- Definition/Use
  - specifies the kind of object to create using a prototypical instance, and creates new objects by cloning this prototype
  - when the type of objects to create is determined by a prototypical instance, which is cloned to produce new objects
  - application ”hard wires” the class of object to create in each ”new” expression
- Structure



- Notes
  - add clone() method
  - add registry
  - put the prototype term in the name of the prototype classes to indicate the use of the pattern to the other developers

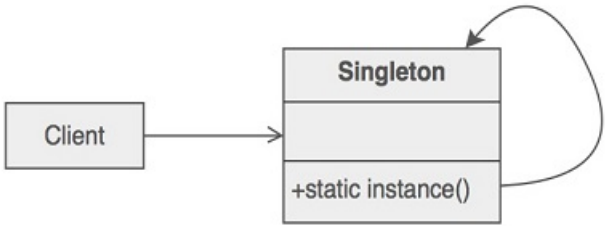
---

Singleton:

- Definition/Use
  - ensures that a class only has one instance, and provides a global point of access to it

  - ensure a class has only one instance, and provide a global point of access to it

- Structure

  - 

- Notes

  - make instantiation(private ThisSingleton()) private aka define all constructors to be protected or private
  - name the method getInstance() to indicate the use of the pattern to the other developers
  - define a private static attribute in the "single instance" class

---

Comparison:

- abstract Factory classes are often implemented with Factory Methods, but they can be implemented using Prototype
- factory method: creation through inheritance, prototype: creation through delegation