Software Engineering

Lesson Design Pattern 05
Template Method
v1.0a

Uwe Gühl

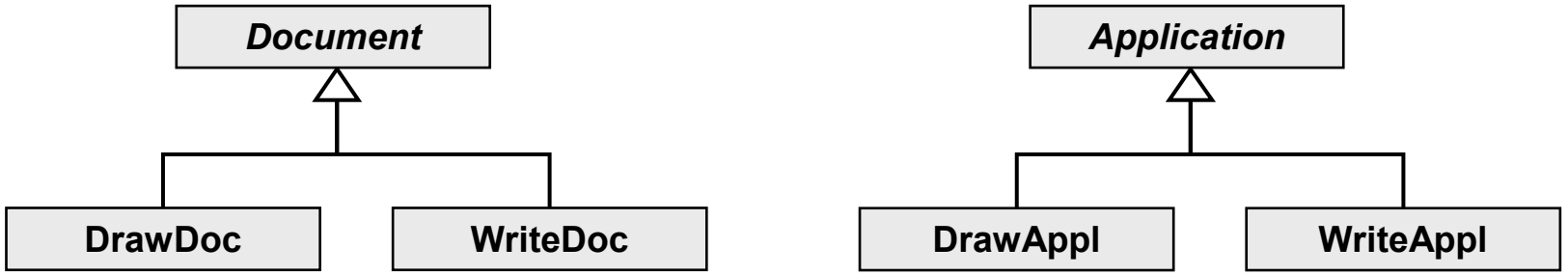
Fall 2007/ 2008

Template Method

- Intent:
  - Define the skeleton of an algorithm in an operation, deferring some steps to subclasses
  - Template Method lets subclasses redefine certain steps of an algorithm without changing the algorithm's structure
  - A parent class declares “placeholders” for algorithms, and child classes implement this placeholders.
  - ... is a Behavioral Pattern

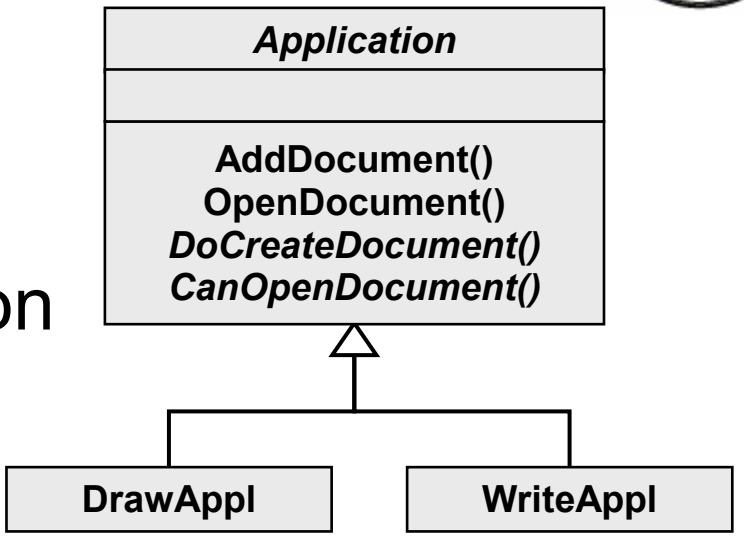
Template Method

- Motivation
  - Application framework with
    - Application classes
      - responsible for opening documents
    - Document classes
      - Represents the information in a document
  - Application framework subclasses Application and Document classes to suit specific needs



Template Method

- Motivation
  - Let's study a little bit Application




```
abstract class Application {

    public void OpenDocument (string docName) {
        if (!CanOpenDocument (docName)) {
            return;                              // can not handle
        }

        Document doc = DoCreateDocument();

        if (doc) {
            _docs.AddDocument(doc);              // ...
            doc.open();
            doc.DoRead();
        }
    }
}
```

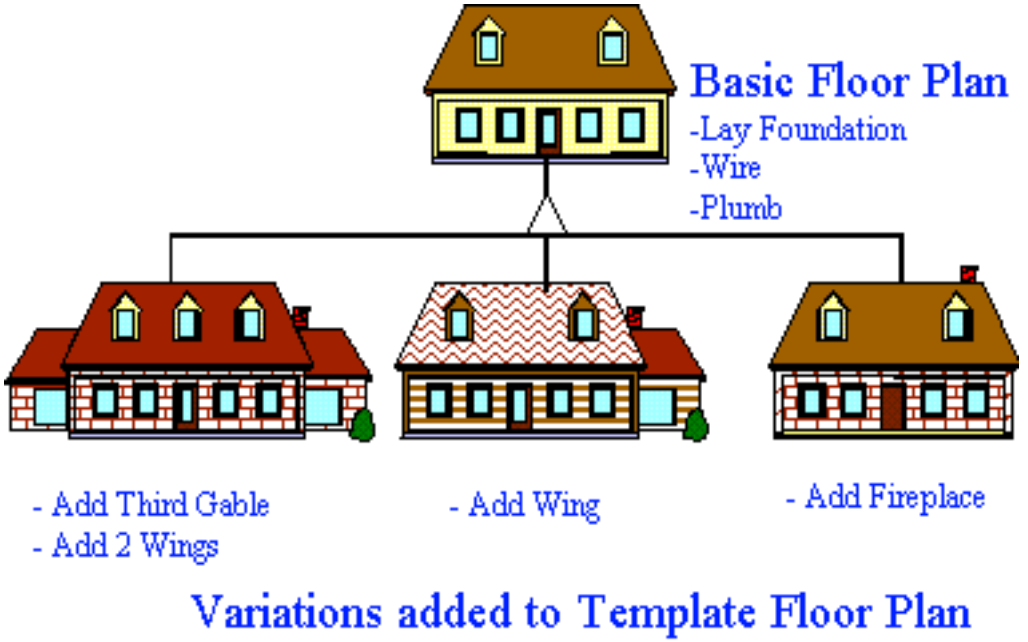
Template Method

Template Method

- Motivation
  - OpenDocument in Application defines all the necessary steps how to open a document – independent which kind of document gets open
  - OpenDocument is called a **Template Method** as it
    - defines abstract operations to be overwritten by subclasses for concrete behavior
    - fixes the order of the steps of the algorithm

Template Method

- Example – Home builders*
  - The basic algorithm for each Home builder is the same – described in a Template Method
  - Foundation, framing, plumbing, and wiring is same for each house
  - In concrete steps of the algorithm variation is possible to produce a wider variety of models.



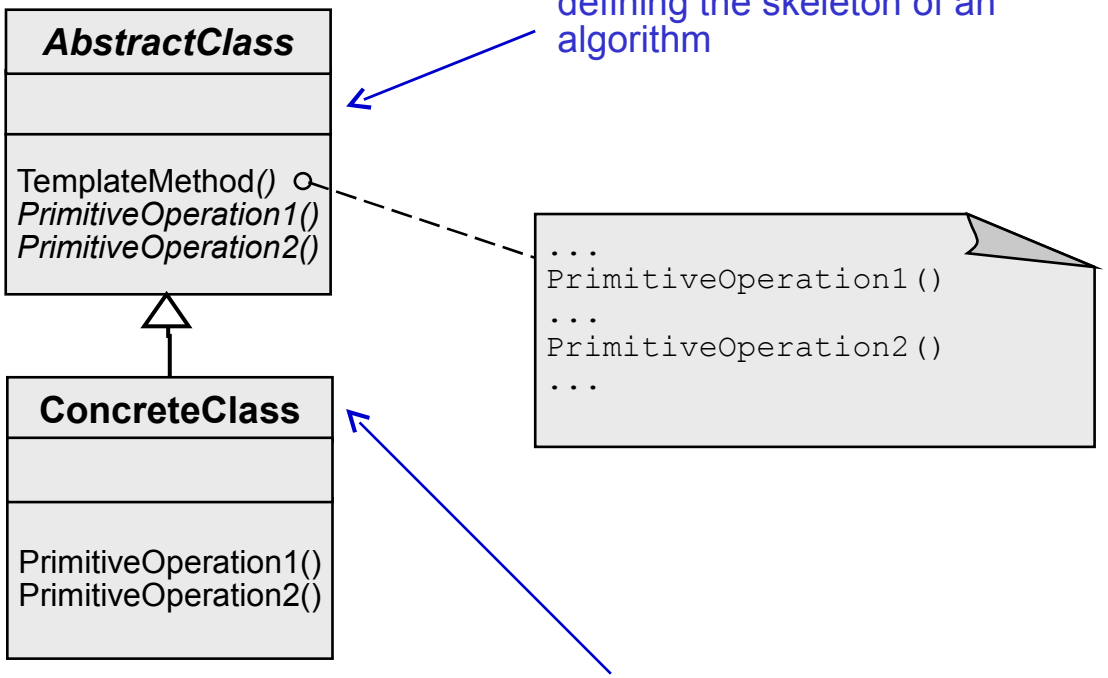
* Source: [Michael Duell, "Non-software examples of software design patterns", Object Magazine, Jul 97, p54]

Template Method

- Consideration
  - To offer the same accessibility the Template and the component need the same interface
  - So clients have transparent access
  - The Template sends requests to the component and executes additional activities (e. g. drawing of a border)
  - Recursive use of multiple Templates allows dynamical adding of functionality

Template Method

- Structure



- defines abstract primitive operations, where child classes implements steps of an algorithm
- implements a **template method** defining the skeleton of an algorithm

- Implements the primitive operations to conduct child class specific steps of the algorithm

Template Method

- Collaboration
  - **ConcreteClass** depends on **AbstractClass** to implement the invariant steps of the algorithm

Template Method

- Applicability
  Use the Template Method Pattern
  - to implement invariant parts of an algorithm once and delegate to the child classes the behavior that could be different
  - if you would like to localize common behavior to avoid code duplication
  - to control how and where child classes extend behavior – with so called “hook” operations

Template Method

- Consequences

+ helps propagating code reuse, that's why important in class libraries

  - leads to an inverted code structure known as “The Hollywood principle” - “Don't call us, we'll call you”, the parent class calls the operation of a child class

Template Method

- Consequences
  - Template Methods call following kind of operations
    - ConcreteClass operations
    - AbstractClass operations
    - primitive operations
    - factory methods (compare to Factory Method)
    - hook operations
      which provide default behavior (e. g. nothing),
      that could be extended by child classes

Template Method

- Consequences
  - Differ
    - hook operations – could be overwritten
    - abstract operations – must be overwritten

Template Method

- Implementation
  - Minimizing primitive operations
    - Important goal: Minimize the number of primitive operations a subclass has to override
  - Naming conventions
    - It's a good idea to point out the operations that have to be overridden, e. g. the MacApp framework for Macintosh applications prefixes Template Methods with “Do-” like “DoRead”

Template Method

- Known Uses (see [GHJ+95])
  - As a fundamental principle they are in almost every abstract class
  - Especially frameworks use template methods

Template Method

- Related Patterns
  - Factory Methods
    They are often called by template methods
  - Difference to Strategy
    - Template Methods use inheritance to change part of an algorithm
    - Strategy uses delegation to modify the entire algorithm