

و ابتغ فيما آتاك الله الدار الآخرة و لا تنس نصيبك
من الدنيا

CS251 – Software Engineering Lectures 18: Intro to DP

Slides by Rick Mercer, Christian Ratliff,
Oscar Nierstrasz and others



Outline

☛ **Introduction to Design Patterns**

☛ **Template Method Pattern**

☛ **Decorator Pattern**



Resources

• The Swing Tutorial

<http://docs.oracle.com/javase/tutorial/uiswing/index.html>

• List of patterns: <http://www.oodesign.com/>



1. Background₁

- ✦ Search for recurring successful designs – emergent designs from practice (via trial and error)
- ✦ Supporting higher levels of reuse (i.e., reuse of designs) is quite challenging
- ✦ Described in Gama, Helm, Johnson, Vlissides 1995 (i.e., “gang of 4 book”)
- ✦ Based on work by Christopher Alexander (an Architect) on building homes, buildings and towns.

Background₂

- ✦ Design patterns represent solutions to problems that arise when developing software within a particular context. E.g., problem/solution pairs within a given context
- ✦ Describes recurring design structures
- ✦ Describes the context of usage

Background₂

- ✦ Design patterns represent solutions to problems that arise when developing software within a particular context. E.g., problem/solution pairs within a given context
- ✦ Describes recurring design structures
- ✦ Describes the context of usage

Background₃

- Patterns capture the static and dynamic *structure* and *collaboration* among key participants in software designs
- Patterns facilitate *reuse* of successful software architectures and designs.

Elements of Design Patterns

✦ Design patterns have four essential elements:

- Pattern name
- Problem
- Solution
- Consequences

Pattern Name

- ✚ A handle used to describe:
 - a design problem
 - its solutions
 - its consequences
- ✚ Increases design vocabulary
- ✚ Makes it possible to design at a higher level of abstraction
- ✚ Enhances communication
- ✚ *“The Hardest part of programming is coming up with good variable [function, and type] names.”*



Problem

- Describes when to apply the pattern
- Explains the problem and its context
- May describe specific design problems and/or object structures
- May contain a list of preconditions that must be met before it makes sense to apply the pattern



Solution

- ✦ Describes the elements that make up the
 - design
 - relationships
 - responsibilities
 - collaborations
- ✦ Does not describe specific concrete implementation
- ✦ Abstract description of design problems and how the pattern solves it

GoF: Design Patterns

Creational:

Abstract Factory

Builder

Factory Method

Prototype

Singleton

Structural:

Adapter

Bridge

Composite

Decorator

Façade

Flyweight

Proxy

Behavioral:

Chain of Responsibility

Command

Interpreter

Iterator

Mediator

Memento

Observer

State / Player-role

Strategy

Template Method

Visitor





Outline

✦ **Introduction to Design Patterns**

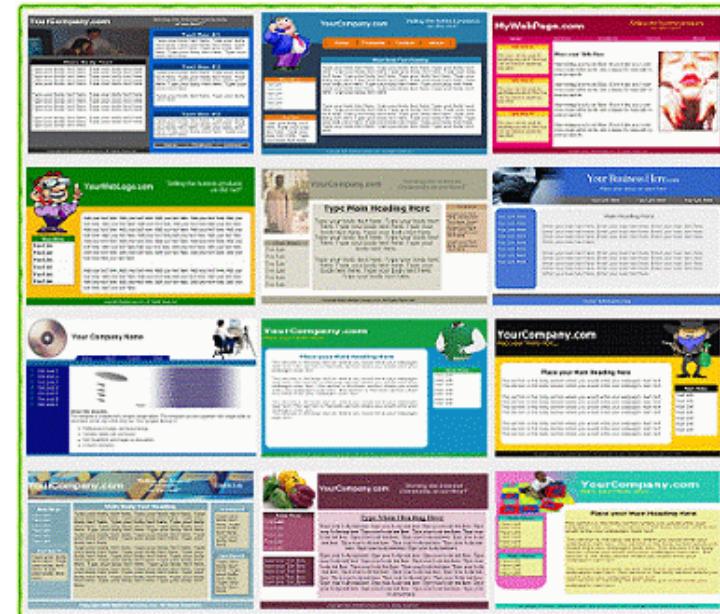
✦ **Template Method Pattern**

✦ **Decorator Pattern**

Template Method Pattern

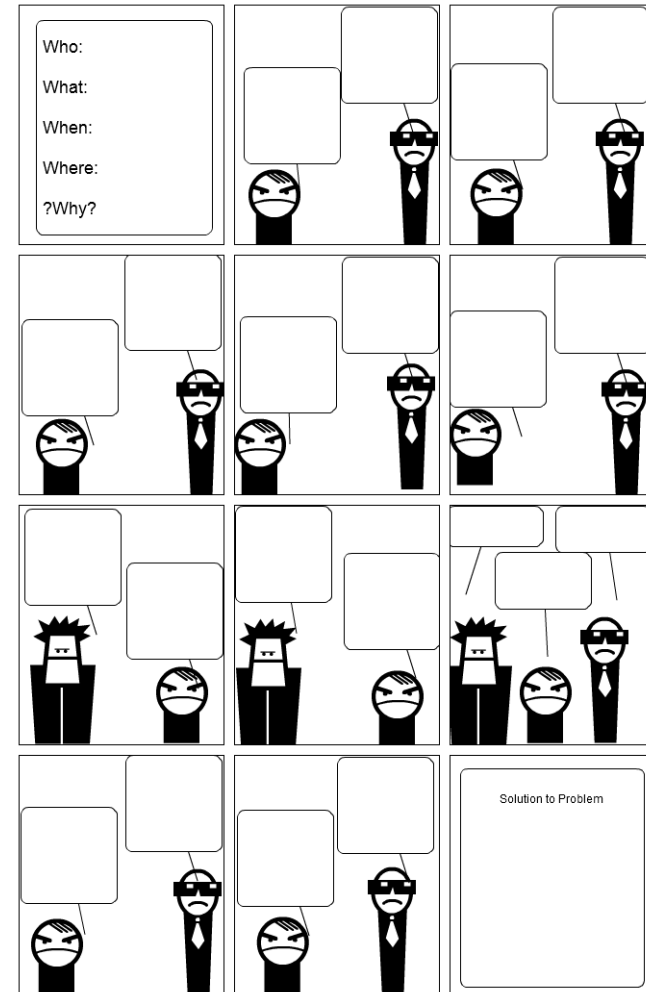
✿ A template is a **preset format**, used as a starting point for a particular application so that the format does not have to be recreated each time it is used.

- MS Word Templates
- Java Generics
- C++ Templates



Template Method Pattern

⚡ A template method defines **an algorithm in a base class** using abstract operations that subclasses override to provide concrete behavior.

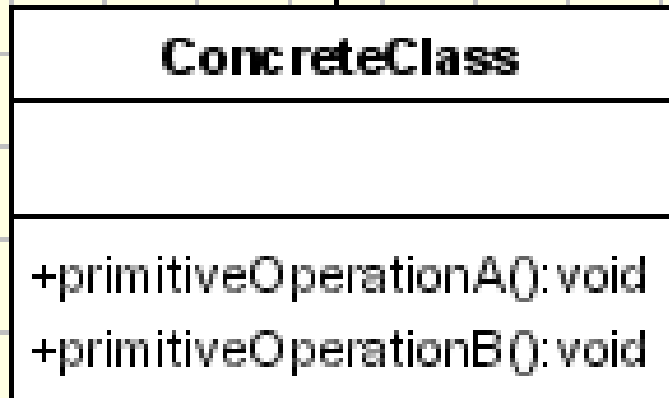
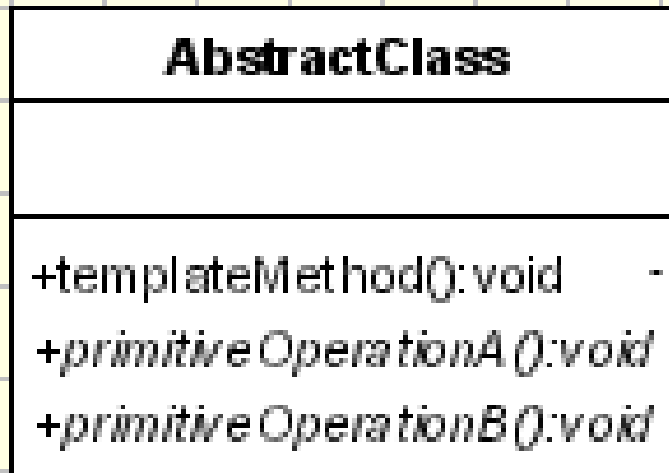




Intent

1. Define the **skeleton of an algorithm** in an operation, deferring some steps to subclasses.
2. Template Method lets **subclasses redefine certain steps** of an algorithm **without** letting them to change the algorithm's structure.

Implementation



```
public final void templateMethod(){
    ...
    primitiveOperationA();
    ...
    primitiveOperationA();
    ...
}
```

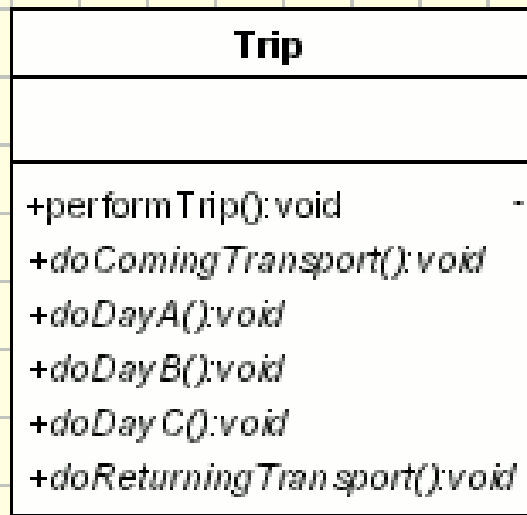
Usage

- ✦ **Implementing the invariant parts** of an algorithm once and leave it up to **subclasses** to implement the behavior that can vary.
- ✦ **Refactoring** is performed and **common behavior** is identified among classes. A abstract base class containing all the common code (in the template method) should be created to avoid code duplication.

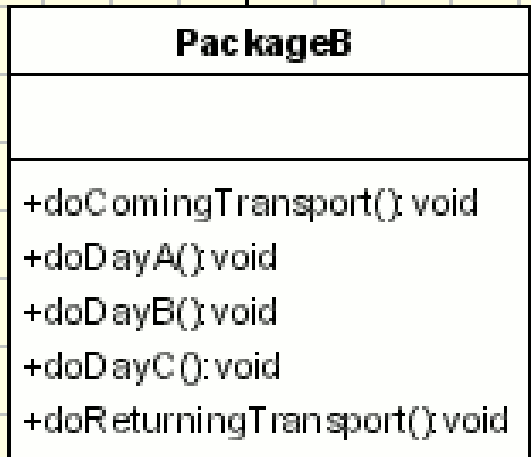
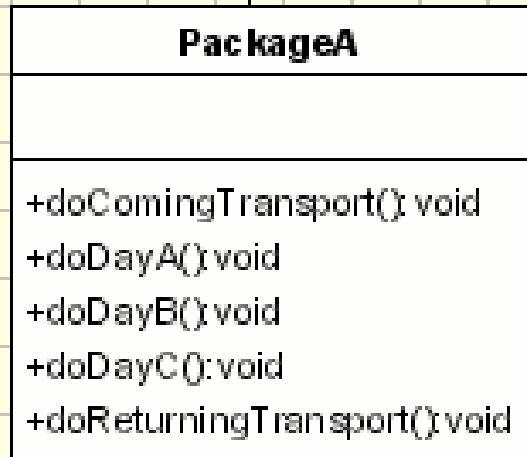
Example 1



- ✚ We have to develop an application for a travel agency.
- ✚ The travel agency is managing each trip.
- ✚ All the trips contain common behavior but there are several packages.
- ✚ Each trip contains the basic steps:
 - The tourists are transported to the holiday location by plane/train/ships,...
 - Each day they are visiting something
 - They are returning back home.



```
public final void performTrip(){
    doComingTransport();
    doDayA();
    doDayB();
    doDayC();
    doReturningTransport();
}
```



Example 1
See Code





Outline



Introduction to Design Patterns



Template Method Pattern



Decorator Pattern



Decorator Pattern

✦ Read it in readings R13

Readings

- Readings 13 and 14
- Read code sets

Top 10 Books Programmers Read

- ✦ *Code Complete* by Steve McConnell (2E, 2004).
- ✦ *The Pragmatic Programmer* by Andrew Hunt and David Thomas (1999).
- ✦ *Structure and Interpretation of Computer Programs* by Abelson, Sussman, and Sussman (2E, 1996).
- ✦ *The C Programming Language* by Brian Kernighan and Dennis Richie.

Top 10 Books Programmers Read

- ✦ *Refactoring: Improving the Design of Existing Code* by Martin Fowler, Kent Beck, John Brant, and William Opdyke (1999).
- ✦ *Design Patterns: Elements of Reusable Object-Oriented Software* by Gamma, Helm, Johnson, and Vlissides (1994). The "Gang of 4" book.
- ✦ *The Mythical Man-Month* by Frederick Brooks (1995).

Top 10 Books Programmers Read

- 1. *The Art of Computer Programming, Volume 1: Fundamental Algorithms* (3rd Edition, 1997) by Donald Knuth.
- 2. *Compilers: Principles, Techniques and Tools 2E* by Aho, Lam, Sethi, and Ullman (2006).

Top Magazines Recommended to Read

☛ *Dr Dobbs* الله يرحمها

☛ *IEEE Software*