6.1 Introduction to Patterns

The recurring aspects of designs are called design patterns.

- A pattern is the outline of a reusable solution to a general problem encountered in a particular context.
- Many of them have been systematically documented for all software developers to use.
- A good pattern should
  - Be as general as possible
  - Contain a solution that has been proven to effectively solve the problem in the indicated context.
- Studying patterns is an effective way to learn from the experiences of others.

Pattern description

Context
- The general situation in which the pattern applies

Problem
- A short sentence or two raising the main difficulty.

 Forces:
- The issues or concerns to consider when solving the problem

Solution
- The recommended way to solve the problem in the given context.
  - Additional factors

 Antipatterns (Optional)
- Patterns that are inferior or do not work in this context.
- Related patterns (Optional)
- Patterns that are similar to this pattern

References
- Who developed or inspired the pattern

6.2 The Abstraction-Occurrence Pattern

- Context:
  - Often in a domain model you find a set of related objects (occurrences).
  - The members of such a set share common information but differ from each other in important ways.

- Problem:
  - What is the best way to represent such sets of occurrences in a class diagram?

- Forces:
  - You want to represent the members of each set of occurrences without duplicating the common information

Antipatterns:

Abstraction-Occurrence
Abstraction-Occurrence

Square variant