Deferring Algorithm Steps with the Template Method Pattern

Gerald Britton
IT SOLUTIONS DESIGNER

@GeraldBritton www.linkedin.com/in/geraldbritton

Motivation

Imagine you are a passenger

Bound for New York or Amsterdam

If close to NYC

- Take the bus to New York

If in Canada

- Fly to New York

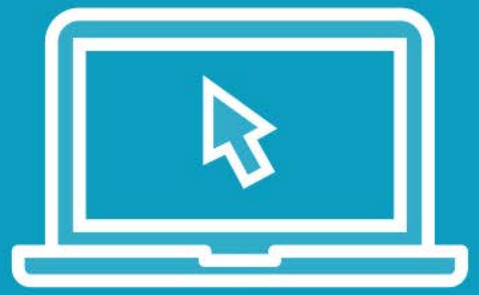
If you live in North America

- Fly to Amsterdam

Busses and planes have a similar purpose

Differ in how they work

Demo

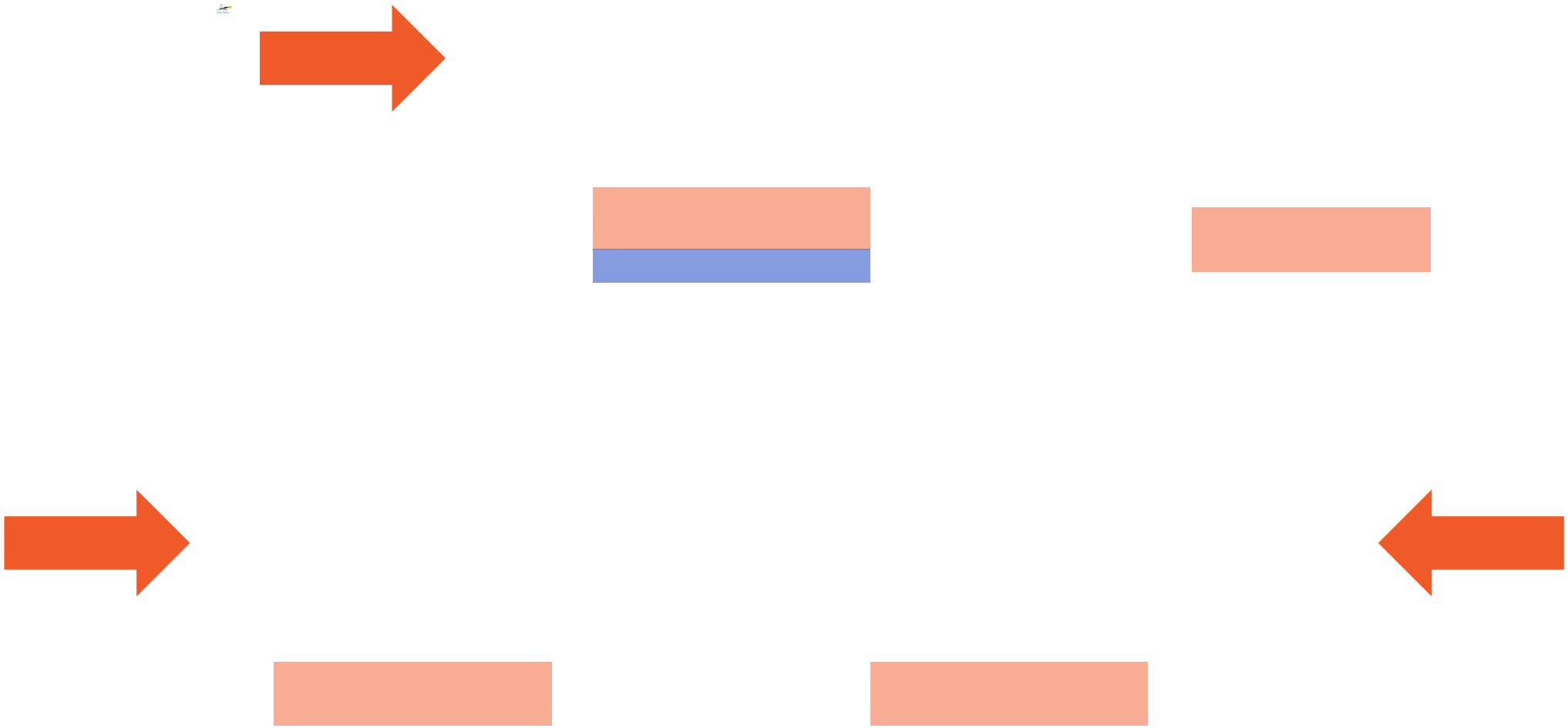


- Build two classes
  - One for a bus trip
  - One for a plane trip
- Test them

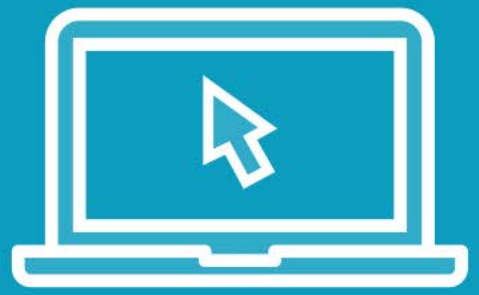
Template Method

- Classification: Behavioral
- Defines algorithm skeleton
- Start with an abstract base class
- Three types of methods
  - Abstract
  - Concrete
  - Hook
- Order of the methods is fixed
- Encapsulated in a template method

Template Method Structure



Demo



Use the Template Method Pattern

Simplify the bus and airplane example

Consequences

- Great platform for code reuse
- Ensures required steps are implemented
- Allows for overriding some steps
- Allows for “hooking” into the algorithm
- Enforces the algorithm’s structure
- Not useful if algorithm must vary
- Comparable to the Builder Pattern
- Action oriented

Summary

When to use Template Method Pattern?

- Recognize similarities in classes

Implement equivalent algorithms

Three method types:

- Abstract
- Concrete but overridable
- Hooks

Don’t repeat yourself!