Software Engineering – Creational design patterns

Adam Krechowicz

1 Creational design patterns

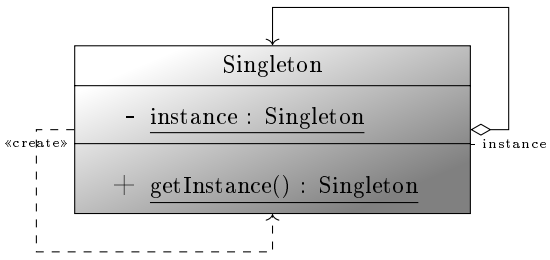
Creational design patterns allow to solve problems during creation of objects.
This group contains 5 patterns:

- Singleton
- Factory method
- Abstract factory
- Builder
- Prototype

1.1 Singleton

Is used when we want to:

- ensure existence of only one instance of class
- allow uniform access to this object

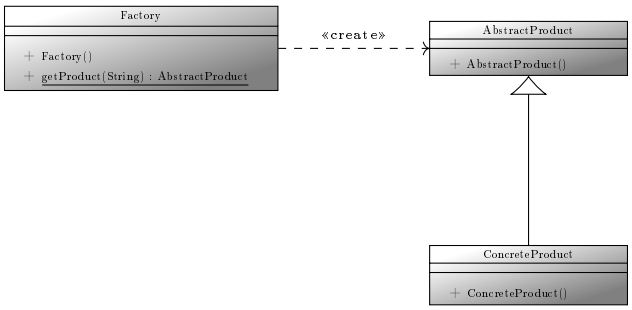


1.2 Factory method

Is used when we want to:

- define interface for object creating, but

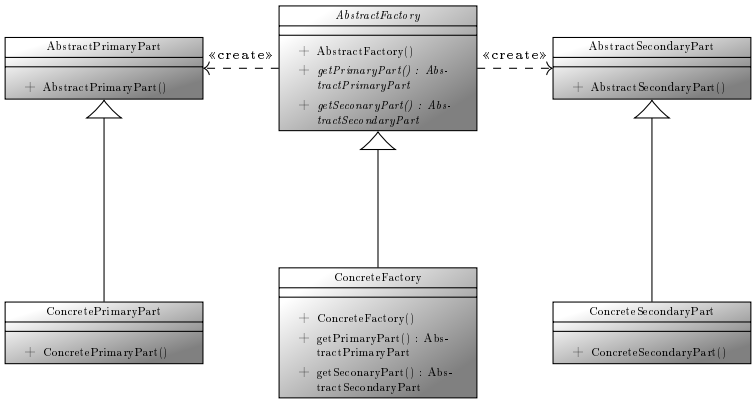
- we do not know the exact class of object



1.3 Abstract factory

Is used when we want to:

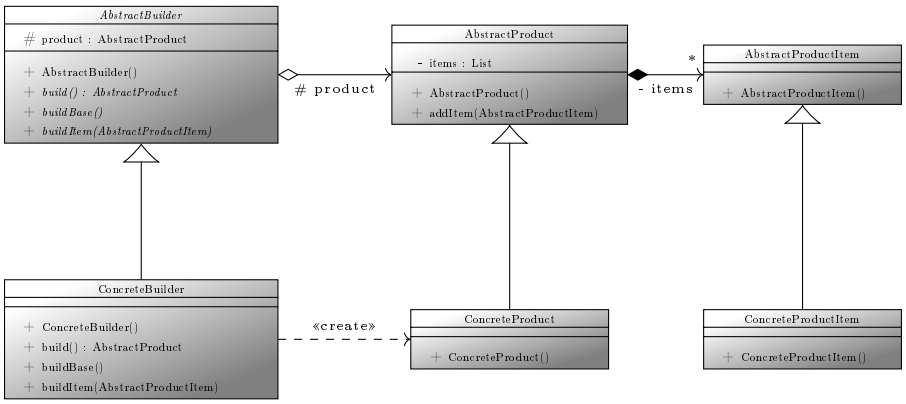
- define interface for object group creating, but
- we do not know the exact class of object



1.4 Builder

Is used when we want to:

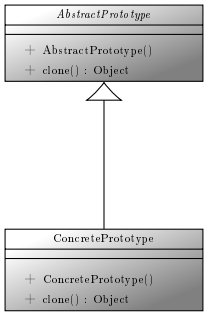
- create objects build from parts
- separate the complicated process of creating from object representation



1.5 Prototype

Is used when we want to:

- create many instances of class
- that are identical or only vary slightly
- are create base on some template



2 Tasks to complete

1. Identify the place for creational design patter in the system
2. Describe the problem that justifies the need of design pattern
3. Describe the pattern its theory and its place in the system
4. Create class diagram for pattern
5. Create source code that implements pattern
6. Create test source code for pattern

Each member of the team should pick other pattern.

Results should be placed in appropriate article each patter for section (<section class="pattern">). The structure of the section is as follows:

- h5 – pattern name
- <p class="author"> – author of pattern
- – pattern problem
- – pattern description
- <p class="uml pattern-diagram"> – pattern class diagram
- <pre class="pattern-code"><code class="lang-java"> – source code
- <pre class="pattern-test-code"><code class="lang-java"> – test source code