Factory ID

Student:

Izabela Kuźniar

Teacher:

Andrea Corradini

Course:

Software Design Patterns

# Name and category

Factory Method is a creational pattern.

# Intent:

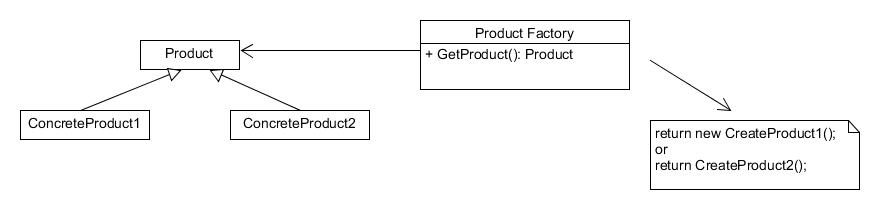
It handles object creation without specifying the exact class of the object that will be created.

# Motivation:

Factory Method Pattern separates the responsibility: clients need to know only how to use an object, factory needs to know what object to create exactly.

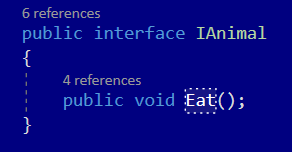
Used for example to keep track of databases connections.

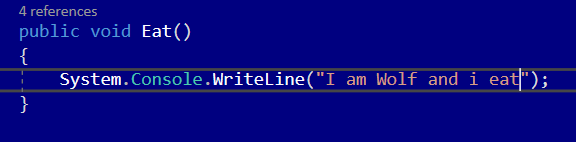
# Structure as a UML class diagram

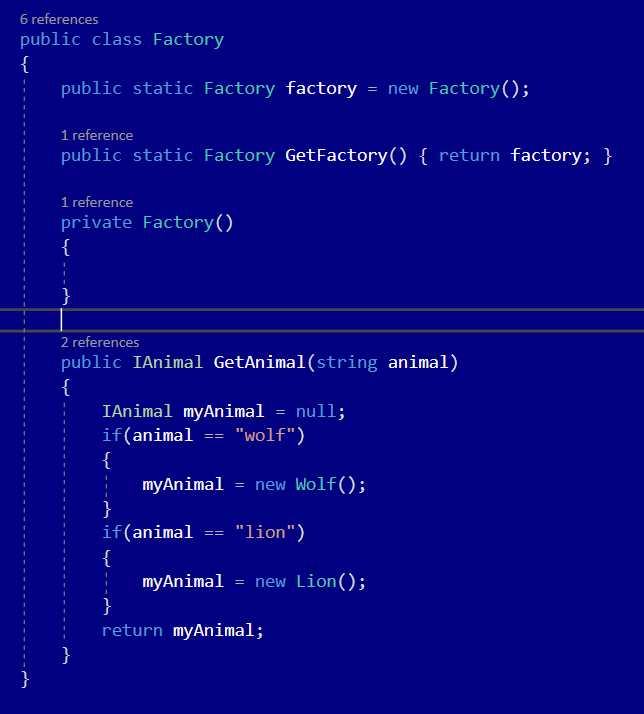


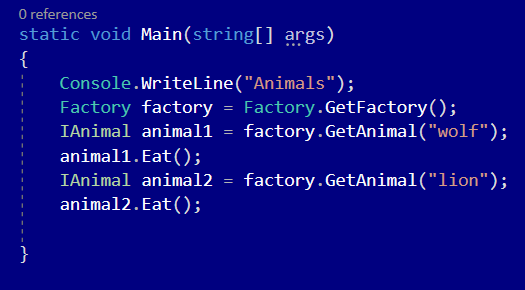
# Implementation:

Note: The Singleton implementation is a special case of static factory method









# Consequences:

Benefits:

* Decoupling
* Separation of responsibilities
* Easier to test

Drawbacks:

* More difficult to read (more code)
* Singleton violates the single responsibility principle: they control their own creation and lifecycle.
* Cause the code to be tightly coupled: makes them difficult to fake during the tests

# Known uses

* Keeping track of connections: socket connections, databases connections
* Separation

# Related patterns

1. Abstract Factory, Prototype and Builder are evolved version of Factory Pattern
2. Factory can be implemented as Singleton.