The Factory Pattern

Gerald Britton
IT Specialist

@GeraldBritton www.linkedin.com/in/geraldbritton

Overview

Classification: Creational

Defines interface for creating an object

Lets subclasses decide which object

Defers instantiation to subclasses

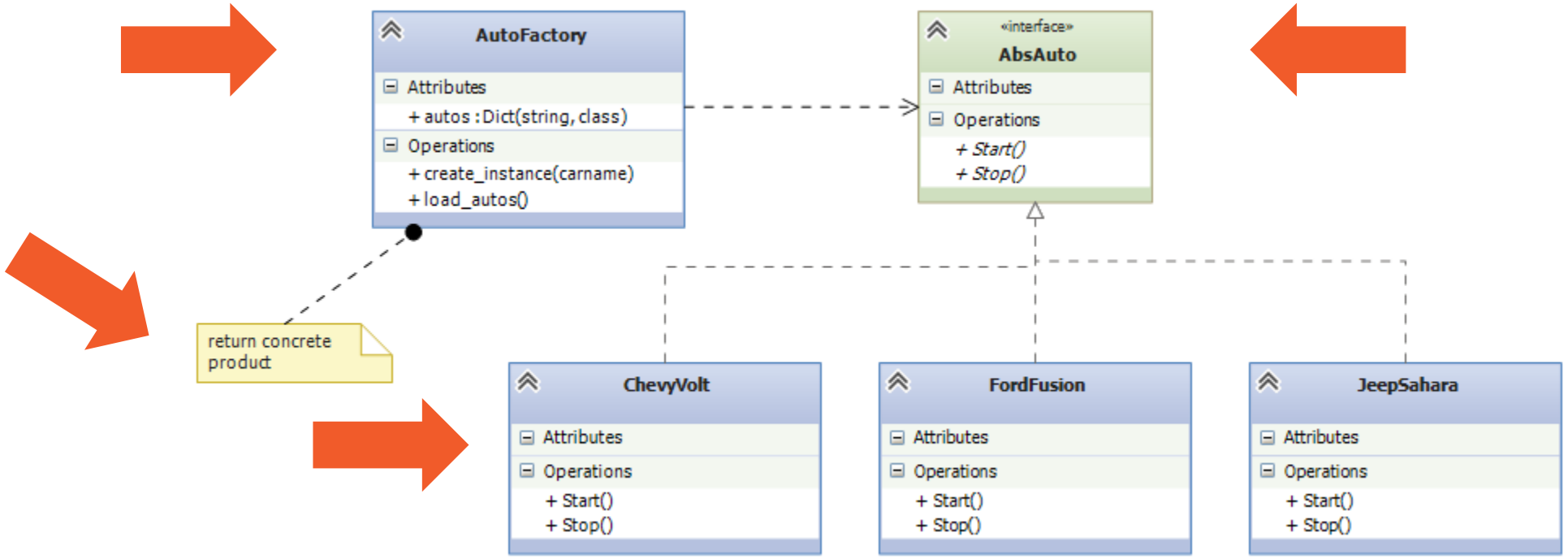
Also known as Virtual Constructor

Demo

Motivating Example:

- Create an object for a model of car
- Support several car models
- Defer the actual model until runtime

Simple Factory Pattern Structure



Demo

Using the Factory pattern

Problems solved so far:

- Eliminated the open/closed violation
- Eliminated the dependencies
- Separated concerns

Limited to one factory

What if we want more factories?

Classic Factory Pattern!

Factory Pattern Structure

Demo

Full Factory pattern

Added an abstract factory base class

Many factories can be implemented

The implementations can vary

A complex factory might use other patterns

- Builder pattern

Encapsulated the factory loader

Summary

Encapsulates object instantiation

Supports dependency inversion

Clients don't depend on implementation

Depend on abstraction instead

Simple Factory Pattern – one factory

- Often just what you need

Classic Factory Pattern – one or more

- Must get an instance of the factory
- Then make the object
- Most flexible