Juniorprogrammierer.de

Java 12: factory pattern

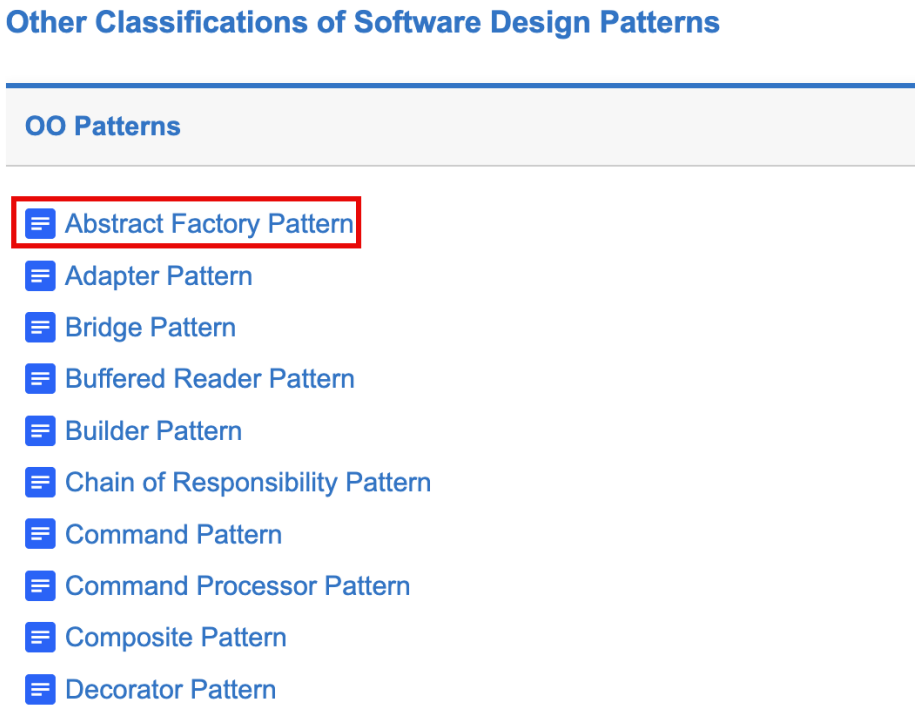
2024/25 – Sascha Stojanovic

Agenda

- What is it?
- Logic and advantage
- Code example
- Exercises

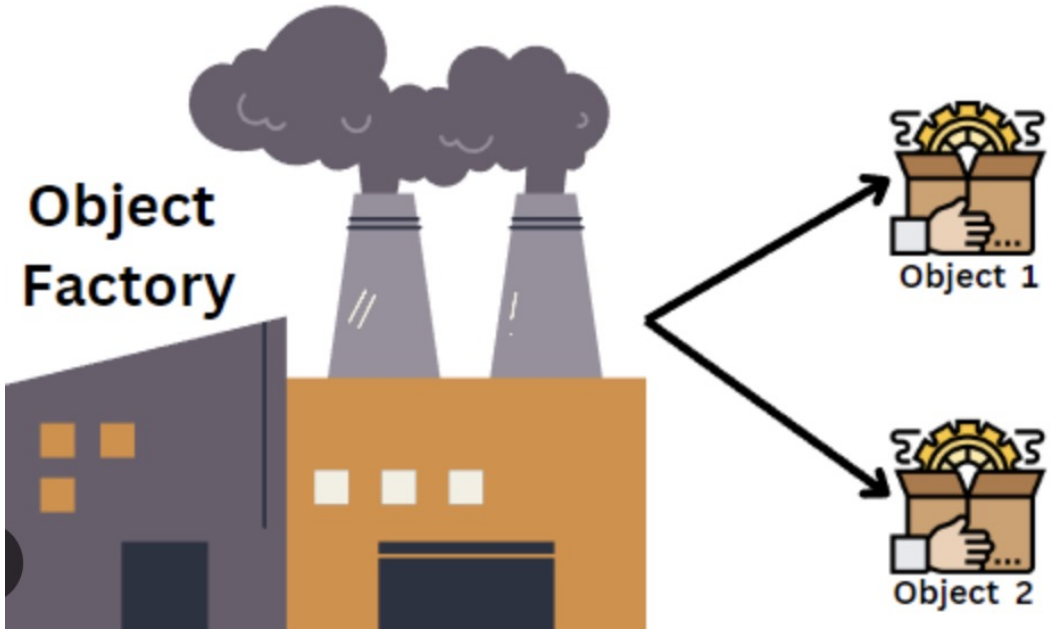
What is it?

- The factory pattern is one of many Software Design Patterns
- These Patterns define how to call for an object
- There are many different Factory pattern but we focus on a simple factory pattern



Logic and advantages

- Instead to allow all classed to create objects of a certain class only the Factory-Class can do so
- Advantage:
  - Since all object creation is in one place, this pattern makes it easier to replace these instances with stubs or mock objects for testing.
  - It is possible to hide the information about the concrete class of the object instances created by the factory by using interfaces.



Code example

Factory methods are in object class

```java
package Codeexamples;

public class Lehrer {


    private String name;
    private String subject;

    // c. Factory methods within the class
    public static Lehrer createMathTeacher(String name) {
        return new Lehrer(name, subject:"Mathematics");
    }

    public static Lehrer createScienceTeacher(String name) {
        return new Lehrer(name, subject:"Science");
    }

    // Private constructor used by factory methods
    private Lehrer(String name, String subject) {
        this.name = name;
        this.subject = subject;
    }
```

Main class is calling the factory -methods

```java
package Codeexamples;

public class LehrerMain {

    Run | Debug
    public static void main(String[] args) {
        Lehrer scienceLehrer = Lehrer.createScienceTeacher( name:"Sascha");
    }
}

```

Exercises

- Create the class “Schueler"
  - Constructor is private and has importing parameter “name” , “age” and ”character”
  - Schueler have a different ” character” of Type String, some are “Classclowns”, some are “Nerds”, some are “Athletes”
  - Create 3 factory methods to create a “Schueler” and give the methods fitting names. They differ depending on the ”character”
- Create a class “SchuelerMain”
  - Try to create a object normally
  - Create a Classclown, Athlete and nerd using your factory methods