模板方法模式
Template Method

波波老师~研发总监/资深架构师

定义

- 模板方法模式在超类中定义算法的总体结构，它将某些步骤的实现推给子类，让子类细化算法的某些步骤，但是子类不能改变算法的总体结构。

案例代码~数据处理DataParser

```
abstract class DataParser {

    // generic data processing flow
    public final void process() {
        readData();
        processData();
        writeData();
    }

    // implemented by subclass
    abstract void readData();
    abstract void processData();

    // same for all subclass
    public void writeData() {
        System.out.println("Ouput generated, writing to CSV");
    }
}
```

模板方法，final子类不能重载

由子类实现的抽象方法

所有子类都相同的实现

代码~子类实现CSVDataParser

```
public class CSVDataParser extends DataParser {

    @Override
    void readData() {
        System.out.println("Reading data from csv file");
    }

    @Override
    void processData() {
        System.out.println("Looping through loaded csv file");
    }

}
```

代码~子类实现DatabaseDataParser

```
public class DatabaseDataParser extends DataParser {

    @Override
    void readData() {
        System.out.println("Reading data from database");
    }

    @Override
    void processData() {
        System.out.println("Looping through records in DB");
    }

}
```

代码~客户端程序

```
Reading data from csv file
Looping through loaded csv file
Ouput generated, writing to CSV
**********
Reading data from database
Looping through records in DB
Ouput generated, writing to CSV
```

```java
public class TemplateMethodMain {

    public static void main(String[] args) {
        CSVDataParser csvDataParser = new CSVDataParser();
        csvDataParser.process();
        System.out.println("**********");
        DatabaseDataParser databaseDataParser = new DatabaseDataParser();
        databaseDataParser.process();
    }

}
```

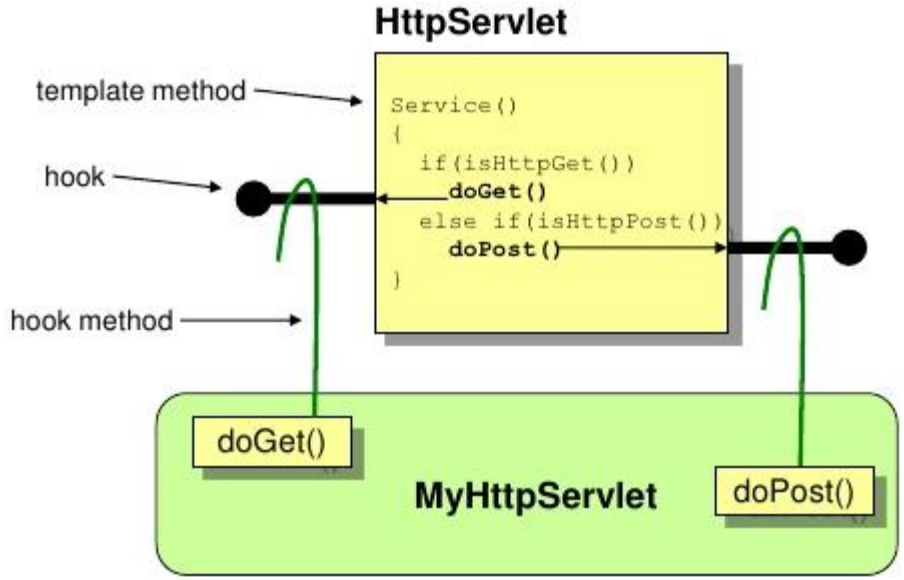
优势

- 扩展灵活~子类可以扩展定制行为
- 避免代码重复~总流程只实现一次
- 规范~规范流程和不能改变的步骤，只有某些步骤可以改

原理

- 开放封闭原则
  - Open-Closed Principle
  - 类应该对修改封闭，对扩展开放
- 好莱坞原理
  - Don’t call me, I will call you
  - IoC核心原理
  - 框架和库的根本区别
    - 框架调你的代码
    - 你调库的代码

应用~HttpServlet



- Life Cycle
  - Init()
  - service()
  - destroy()
- service()方法是GenericServlet类的一个抽象方法
- HttpServlet实现service()方法，定义了模板处理HTTP请求
  - doGet, doPost…
- 开发人员扩展HttpServlet并根据需求重写doGet, doPost等

其它应用

- Java IO中InputStream/OutputStream/Reader/Writer的非抽象方法
- 抽象集合类AbstractList/AbstractSet/AbstractMap中的非抽象方法
- Spring JDBC/REST Template
- **模板方法是框架之母**

课后思考

- 模板方法模式 vs 策略模式

参考

- Template method design pattern in java
  - https://java2blog.com/template-method-design-pattern-in-java/
- Template Method Design Pattern
  - https://howtodoinjava.com/design-patterns/behavioral/template-method-pattern/

代码+ppt

- https://github.com/spring2go/core-spring-patterns

波波微课

spring2go.com