

# WebQuest

## Aula Semana 09

### Mais Sobre Padrões de Projeto Básicos:

#### Static Factory Method, Null Object,

#### Hook Methods e Hook Classes

#### Introdução

O objetivo deste WebQuest é consolidar o entendimento e implementação dos seguintes padrões básicos: Static Factory Method, Null Object, Hook Methods e Hook Classes.

Um padrão é básico se ele é usado isoladamente ou como parte de outros padrões de projeto do livro GoF [Recurso Secundário 1].

Recomendo comprar o livro do Prof. Guerra [Recurso Secundário 2].

#### Tarefa

Conhecer, ver exemplos e exercitar o uso dos padrões de projeto básicos Static Factory Method, Null Object, Hook Methods e Hook Classes.

#### Processo

##### 1. [Com seu colega do lado/da frente/de trás]

- a. [05min] [Recurso Primário 1] Definir o que é e para que serve o padrão básico Static Factory Method, nomes alternativos e estrutura.

Resposta: É um design pattern utilizado para encapsular a criação de objetos. Também conhecido como "Construtor Virtual", é construído primeiramente ao tornar privado o construtor da classe e, em seguida, definindo-se métodos estáticos que realizam a instanciação de objetos.

- b. [10min] Dada a classe `RandonIntGenerator`, que gera números aleatórios entre um mínimo e um máximo, implemente-a passo-a-passo:

```
public class RandonIntGenerator {  
    public int next() {...}  
    private final int min;  
    private final int max;  
}
```

Como os valores `min` e `max` são final, eles devem ser inicializados na declaração ou via construtor. Vamos inicializar por meio de um construtor!

```
public RandonIntGenerator(int min, int max) {  
    this.min = min;  
    this.max = max;  
}
```

Crie um novo construtor, supondo que o valor `min` é fornecido e o valor `max` é o maior valor inteiro do Java (`Integer.MAX_VALUE`)!

```
public RandonIntGenerator(int min) {  
    this.min = min;  
    this.max = Integer.MAX_VALUE;  
}
```

Crie um novo construtor, supondo que o valor `max` é fornecido e o valor `min` é o menor valor inteiro do Java (`Integer.MIN_VALUE`)!

```
public RandonIntGenerator(int max) {  
    this.min = Integer.MIN_VALUE;  
    this.max = max;  
}
```

Como resolver este problema? Ok

- c. [05min] Melhore a legibilidade do código abaixo:

```
public class Foo{
    public Foo(boolean withBar){
        //...
    }
}
```

**Foo foo = new Foo(true);**

```
// -----

public class Foo{
    private Foo(boolean withBar){
        //...
    }
    public createWithBar(){
        return new Foo(true);
    }
    public createWithoutBar(){
        return new Foo(false);
    }
}
```

- d. [Exercício para Casa] Em [Recurso Primário 1], estende-se o gerador de inteiro do item b) para suportar inteiro, Double, Long e String. Mostrar uma implementação com static factory methods que resolva essa situação
2. [Com outro colega do lado/da frente/de trás][Mudar de local, se for preciso]
- a. [05min] Definir o que é e para que serve o padrão básico Null Object, nomes alternativos e estrutura.

Resposta: No Design Pattern Null Object, um objeto nulo substitui a verificação da instância do objeto Null. Em vez de utilizar uma verificação com "if" para verificar o valor nulo, Null Object define uma solução pra conseguir lidar com referencias nulas na aplicação. Esse objeto nulo também pode ser usado para fornecer um comportamento padrão caso os dados não estejam disponíveis.

- b. [10min] Dada a classe RealCustomer abaixo, projetar e implementar um exemplo de aplicação simples, mostrando o antes (sem o padrão) e o depois (com o padrão) quando alguns clientes reais existem no repositório de clientes e outros ainda não fazem parte dele! Simular tudo o que for

necessário para exemplificar a necessidade do uso do Null Object, inclusive o repositório de clientes!

```
public class RealCustomer {  
    public RealCustomer(String name) {  
        this.name = name;  
    }  
    @Override  
    public String getName() {  
        return name;  
    }  
    @Override  
    public boolean isNil() {  
        return false;  
    }  
}
```

3. [Com outro colega do lado/da frente/de trás][Mudar de local, se for preciso]

- a. [05min] Definir o que é e para que serve o padrão básico Hook Method, nomes alternativos e estrutura. [Recursos Primários 3 e 4]

Resposta: O Design Pattern Hook Method é utilizado para valorizar a especialização de subclasses. Os métodos da superclasse invocam métodos que são implementados apenas nas subclasses, cada uma da forma mais conveniente.

- b. [10min] Pesquisar no [Recursos Primários 3 e 4] ou em qualquer outra fonte e projetar e implementar um exemplo de aplicação simples, mostrando o antes (sem o padrão) e o depois (com o padrão)! Ok

4. [Com outro colega do lado/da frente/de trás][Mudar de local, se for preciso]

- a. [07min] Diferencie hook method de hook class, começando com um exemplo não operacional em Java que implementa um hook method e transforme-o em hook class.

### **Recursos Primários**

1. [Static Factory Method] <http://jlordiales.me/2012/12/26/static-factory-methods-vs-traditional-constructors/>

- (former link: <http://jlordiales.wordpress.com>)
2. [Null Object]  
[https://sourcemaking.com/design\\_patterns/null\\_object](https://sourcemaking.com/design_patterns/null_object)
  3. PDF com arquivo do link desativado  
<https://www.cs.oberlin.edu/~jwalker/nullObjPattern/> [TIDIA - Semana 09]
  4. [Hook Methods 1] Hook Methods—Livro Guerra [TIDIA - Semana 09]
  5. [Hook Methods 2] <http://c2.com/cgi/wiki?HookMethod>
  6. [Hook Classes] Hook Classes—Livro Guerra [TIDIA - Semana 09]

### **Recursos Secundários**

1. Gamma, Erich; Richard Helm, Ralph Johnson, and John Vlissides (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ["Gang of Four" or GoF]
2. Eduardo Guerra. Design Patterns com Java: Projeto Orientado a Objetos Guiado por Padrões. São Paulo: Casa do Código, 2013. [ISBN 978-85-66250-11-4][e-Book R\$ 29,90]
3. Null Object apresentado como refatoração:  
<http://www.refactoring.com/catalog/introduceNullObject.html>
4. Null Object é chamado de "Special Case" no catálogo "EAA" do Fowler: <http://martinfowler.com/eaCatalog/specialCase.html>