Padrão de projetos de software

Padrão de projeto Visitor

Curso de Engenharia de software
Disciplina: Padrão de projetos de software
Professor: Guilherme

O que é o padrão visitor?

Tradução do padrão: Padrão de visitante

Na programação orientada a objetos e na engenharia de software, o padrão de projeto do visitante é uma forma de separar um algoritmo de uma estrutura de objeto na qual ele opera. Um resultado prático dessa separação é a capacidade de adicionar novas operações a estruturas de objetos existentes sem modificar as estruturas.

O padrão Visitor sugere que você coloque o novo comportamento em uma classe separada chamada visitor , em vez de tentar integrá-lo às classes existentes.

Criação

DIGITAL INNOVATION ONE

Criação

onde:

V: "public", "protected" ou "private"
T: concreto ou abstrato
M: "static" ou "final"
R: tipo de dado ou "void" // não retorna nada só
N: nome que é fornecido ao método// padrão
P: parâmetros que pode receber // se o método for usado de ficar dentro do parâmetro ou parâmetro vazio
E: exceções que pode lançar
C: código que possui ou vazio

DIGITAL INNOVATION ONE

Particularidades

- Assinatura: é a forma de identificar unicamente o método

  Ass = nome + parâmetros

  Método:

```
public double calcularTotalVenda(double
precoItem1, double precoItem2, double precoItem3)
{...}
```

  Assinatura:

```
calcularTotalVenda(double precoItem1,
double precoItem2, double precoItem3)
```

DIGITAL INNOVATION ONE

Exemplos

public String getNome() { … } // retorna um Nome
public double calcularTotalNota() {…}
public int verificarDistancia(int cordenada1, int cordenada2) {…}
public abstract void executar() ; // corpo vazio do método
public void alterarFabricante(Fabricante fabricante) { … }
public Relatorio gerarDadosAnaliticos(Cliente cliente, List<Compra> compras) {…} // como passar mais de um paramentro

public static R N(P) {…}

DIGITAL INNOVATION ONE

Utilização

Passa-se uma mensagem através de uma classe ou objeto.

nome_da_classe.nome_do_metodo(); ou nome_da_classe.nome_do_metodo(...);

nome_do_objeto.nome_do_metodo(); ou nome_do_objeto.nome_do_metodo(...);

Math.random(); ou Math.sqrt(4);

usuario.getEmail(); ou usuario.alterarEndereco(
endereco);

DIGITAL INNOVATION ONE

Particularidades

- Construtor e Destrutor: são métodos especiais usados na Orientação a Objetos.

- Mensagem: é o ato de solicitar ao método que o mesmo execute. Esta pode ser direcionada a um objeto ou a uma classe.

DIGITAL INNOVATION ONE

Boas práticas

- Nomes devem ser descritivos, mas curtos
- Notação camelo

```
verificarSaldo(); executarTranferencia(...); existeDebito();
```

- Deve possuir entre 80 e 120 linhas
- Evite lista de parâmetros longas
- Visibilidades adequadas

DIGITAL INNOVATION ONE

Exercitando

Cria uma aplicação que resolva as seguintes situações:

- Calcule as 4 operações básicas: soma, subtração, multiplicação e divisão. Sempre 2 valores devem ser passados.
- A partir da hora do dia, informe a mensagem adequada: Bom dia, Boa tarde e Boa noite.
- Calcule o valor final de um empréstimo, a partir do valor solicitado. Taxas e parcelas influenciam. Defina arbitrariamente as faixas que influenciam nos valores.

DIGITAL INNOVATION ONE

Exercitando

Observações:

- Tente ao máximo criar métodos que trabalhem sozinhos ou em conjunto
- Pode chamar um método dentro de outro
- Pode passar como parâmetro, a chamada de um outro método

DIGITAL
INNOVATION
ONE

Aula 2: Sobrecarga

Métodos

DIGITAL INNOVATION ONE

Objetivos

1. Entender o que é sobrecarregar um método

2. Saber como criar sobrecargas

"É a capacidade de definir métodos para diferentes contextos, mas preservando seu nome. "

DIGITAL INNOVATION ONE

Criação

Alterar a assinatura do método:

Ass = nome + parâmetros

```
converterParaInteiro(float f);
converterParaInteiro(double d);
converterParaInteiro(String s);
converterParaInteiro(float f, RoundType rd);
converterParaInteiro(double d, RoundType rd);
converterParaInteiro(String s, RoundType rd);

converterParaInteiro(RoundType rd, String s);
converterParaInteiro();
```

DIGITAL INNOVATION ONE

Exemplos

https://docs.oracle.com/javase/7/docs/api/java/io/PrintStream.html

ghjj

void	**println()** Terminates the current line by writing the line separator string.
void	**println(boolean x)** Prints a boolean and then terminate the line.
void	**println(char x)** Prints a character and then terminate the line.
void	**println(char[] x)** Prints an array of characters and then terminate the line.
void	**println(double x)** Prints a double and then terminate the line.
void	**println(float x)** Prints a float and then terminate the line.
void	**println(int x)** Prints an integer and then terminate the line.
void	**println(long x)** Prints a long and then terminate the line.
void	**println(Object x)** Prints an Object and then terminate the line.
void	**println(String x)** Prints a String and then terminate the line.

DIGITAL INNOVATION ONE

Exemplos

https://docs.oracle.com/javase/7/docs/api/java/lang/String.html

`static String`	**`valueOf`**`(boolean b)` Returns the string representation of the `boolean` argument.
`static String`	**`valueOf`**`(char c)` Returns the string representation of the `char` argument.
`static String`	**`valueOf`**`(char[] data)` Returns the string representation of the `char` array argument.
`static String`	**`valueOf`**`(char[] data, int offset, int count)` Returns the string representation of a specific subarray of the `char` array argument.
`static String`	**`valueOf`**`(double d)` Returns the string representation of the `double` argument.
`static String`	**`valueOf`**`(float f)` Returns the string representation of the `float` argument.
`static String`	**`valueOf`**`(int i)` Returns the string representation of the `int` argument.
`static String`	**`valueOf`**`(long l)` Returns the string representation of the `long` argument.
`static String`	**`valueOf`**`(Object obj)` Returns the string representation of the `Object` argument.

DIGITAL INNOVATION ONE

Curiosidade

Sobrecarga x Sobrescrita

DIGITAL INNOVATION ONE

Exercitando

Cria uma aplicação que calcula a área dos 3 quadriláteros notáveis: quadrado, retângulo e trapézio.

Obs: Use sobrecarga.

DIGITAL
INNOVATION
ONE

Aula 3: Retornos

Métodos

DIGITAL INNOVATION ONE

Objetivos

1. Entender como funcionam

DIGITAL INNOVATION ONE

Relembrando

-
- É uma instrução de interrupção
- Simbologia: return

DIGITAL INNOVATION ONE

Funcionamento

O método executa seu retorno quando:

- Completa todas suas instruções internas
- Chega a uma declaração explícita de retorno
- Lança uma exceção

DIGITAL INNOVATION ONE

Considerações

- O tipo de retorno do método é definido na sua criação e pode ser um tipo primitivo ou objeto;
- O tipo de dado do return deve ser compatível com o do método;
- Se o método for sem retorno(void), pode ou não ter um "return" para encerrar sua execução.

DIGITAL INNOVATION ONE

Exemplos

```
public String getMensagem() {
   return "Ola!";
}
public double getJuros() {
   return 2.36;
}
public int getParcelas() {
   return 1.36f;
}
```

```
public void setIdade(...) {
    return 10;
}
public void executar() {
  ...
  return;
  ....
 }
```

DIGITAL INNOVATION ONE

Exercitando

Recrie a aplicação que calcula a área dos 3 quadriláteros notáveis. Agora faça os métodos retornarem valores.

DIGITAL INNOVATION ONE

Dúvidas durante o curso?

> Fórum do curso
> Comunidade online (discord)