

Exercícios - Semana 13



Nas semanas 10 e 11, você teve contato com um projeto que faz uso do padrão de projeto *Strategy*. Na semana 12, conheceremos padrão Bridge.

Bridge é um **padrão de projeto estrutural** que permite que você divida uma classe grande ou um conjunto de classes intimamente ligadas em duas hierarquias separadas — abstração e implementação — que podem ser desenvolvidas independentemente umas das outras.

Utilize o padrão Bridge quando você quer dividir e organizar uma classe monolítica que tem diversas variantes da mesma funcionalidade (por exemplo, se a classe pode trabalhar com diversos servidores de base de dados). Utilize o Bridge também se você precisar ser capaz de trocar implementações durante o momento de execução. É comum confundir o Bridge com o padrão Strategy. Mas um padrão é mais que apenas uma maneira de estruturar suas classes. Ele também pode comunicar intenções e resolver um problema.

Projeto da semana

No projeto desta semana, simulamos a extração de dados de diversos ERPs fictícios (ERP1, ERP2) e a capacidade de gerar relatório em diversos formatos (texto, arquivo csv, outro banco de dados, etc..). A classe *Abstracao* é instanciada pelo cliente. Imagine quantas combinações de classes seriam necessárias sem o padrão Bridge: *ERP1RelatorioTXT*, *ERP1RelatorioCSV*, *ERP2RelatorioTXT*, *ERP2RelatorioCSV*, e assim por diante...

Desafio 1 (1 ponto)

Ajustar a extração do ERP1 para que retorne somente o total e vendido_em. Não é necessário extrair o produto vendido.

Crie ou ajuste os testes da sua implementação.

Desafio 2 (1 ponto)

Ajustar a extração do ERP2 para que retorne somente o total e vendido_em. Não é necessário extrair o produto vendido.

Crie ou ajuste os testes da sua implementação.

Desafio 3 (2 pontos)

O método **execute** da classe **ExtrairDados** é um método abstrato. Entretanto, os comandos SQL não estão *hard-coded*. Faria sentido refatorar este código e migrar o método **execute** para a classe base? Caso sim, refatore o código. Caso não, justifique sua resposta.

R: Pelo meu entendimento da questão, não faria sentido refatorar o código, pois a classe “cliente” está importando a “Abstração” em que a mesma está importando a classe “Extrair”.

Desafio 4 (3 pontos)

No código **cliente.py**, ao descomentar a linha 20 e executar novamente, ocorrerá uma *Exception*. O formato esperado de relatório não está implementado. Implemente a classe **Relatorio_CSV**, uma classe que possui como classe base **Relatorios**. O método **criar_relatorio** deve seguir o padrão de implementação recomendado na classe base. O relatório gerado será um arquivo csv.

Crie ou ajuste os testes da sua implementação.

Dica: descomentar a linha 11 do arquivo **Abstracao.py**

Desafio 5 (3 pontos)

Este desafio não envolve codificação. Explique com suas palavras como seria a implementação de uma classe que extraia dados de um ERP3, em que o banco de dados é o MySQL. Quais dados seriam necessários estar presentes nas credenciais ?

R:

No script Extrair criaria uma classe chamada ERP3 com parâmetro ExtrairDados;

Uma função para se conecta ao MySQL no qual diferente do SQLite, ele não é um arquivo, mas

Um serviço que precisa de conexão;

As funções `get_query`, `get_query_report` e `execute` assim como está nesse código de exemplo, com isso pegaria os dados do DB, e os dados necessários nas credências:

```
# Nome do ERP
```

```
erp==>ERP3
```

```
# Conexão do banco de dados
```

```
host='localhost'
```

```
database='cliente'
```

```
user='root'
```

```
password='123'
```

```
# Formato do relatório.
```

```
relatorio==>csv
```

```
# Nome do arquivo. Espera-se ao final um arquivo erp3.csv
```

```
relatorio_nome==>erp3
```