Exercícios - Semana 12

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

No projeto desta semana, as características do produto (implementação) estão separadas do produto (abstração). Ter classes únicas levaria uma explosão de classes: CocaCola600ml, CocaCola1litros, CocaCola2litros, CocaCola3litros, Pepsi600ml, Pepsi1litros, Pepsi2litros, Pepsi3litros, etc, etc, etc...

Desafio 1
Implementar um implementação(Característica) para **Tamanho2litros**.
Crie testes para suas implementações.

Desafio 2
Implementar um implementação(Característica) para **Tamanho3litros**.
Crie testes para suas implementações.

Desafio 3
Implementar uma abstração(Produto) para **Dolly**.
Crie testes para suas implementações.

Desafio 4
Implementar uma abstração(Produto) para **GuaranaAntartica**.
Crie testes para suas implementações.

Desafio 5
Ajustar o código **produtos.py** para que exiba as novas combinações a partir dos produtos e características gerados acima.

Desafio 6
Na classe abstrata **Produtos,** altere o construtor para que a implementação permitida seja apenas subclasses de **Caracteristicas**. Caso o parâmetro não seja sublcasse de Caracerística, deve-se gerar uma *Exception*. Crie testes para esta melhoria.

Desafio 7
Verifique a cobertura de testes (com uso do pytest-cov). A cobertura de testes deve ser superior a 90%.
Verifique se sua implementação está de acordo com o guia de estilos da linguagem python (PEP-8). Utilize o flake8 ou outra ferramenta similar.

Desafio 8
“Quanto maior a classe se torna, mais difícil é de entender como ela funciona, e mais tempo se leva para fazer mudanças. As mudanças feitas para uma das variações de funcionalidade podem precisar de mudanças feitas em toda a classe, o que quase sempre resulta em erros ou falha em lidar com efeitos colaterais.
O padrão Bridge permite que você divida uma classe monolítica em diversas hierarquias de classe. Após isso, você pode modificar as classes em cada hierarquia independentemente das classes nas outras. Essa abordagem simplifica a manutenção do código e minimiza o risco de quebrar o código existente.”

Trecho do livro “Mergulho nos padrões de projeto”

Você identifica algum código/cenário dentro da Napp em que o padrão de projeto estudado poderia ser aplicado para melhora do produto ou redução de manutenção ?