Universidade Católica de Pernambuco
Unicap Icam-TECH
Ciência da Computação e Sistemas para Internet

Lista de Exercícios

Identificação da Aula

Professor: Diego Pinheiro, Ph.D.
Disciplina: Projeto de Extensão Tecnológica (PET) UNICAP-SIDI
Atividade: 01 – Padrões de Projeto

Instruções

1. Você **deve utilizar o projeto java disponibilizado** me formato .ZIP no classroom da disciplina.

2. Sua implementação deve estar dentro da pasta src/**main**/java (**não** coloque detro da pasta src/**test**/java).

3. A submissão deve conter **todo o projeto** compactado em formato **.ZIP**

4. O nome do arquivo .ZIP submetido deve respeitar a nomenclatura recebida exceto **NOME_SOBRENOME** que devem ser substituídos por seu nome e sobrenome, respectivamente.

5. A submissão **não deve ser feita após o prazo** (nem 1 minuto a mais)

Atividade

Parte 1

Utilize o padrão **Strategy** para definir uma família de algorítmos **Operation**. Encapsule as operações de suma e multiplicação nas subclasses **OperationSum** e **OperationMultiplication**, respectivamente.

Parte 2

Utilizando o padrão de projeto **AbstractFactory** para possibilitar a criação de estratégias **OperationSum** e **OperationMultiplication** através da interface ***FactoryOperations***. Para a **FactoryOperationsSumThenMultiplication,** os métodos *createBottomOperation e createTopOperation* criam **OperationSum** e **OperationMultiplication,** respectivamente. Para a **FactoryOperationsMultiplicationThenSum,** os métodos *createBottomOperation e createTopOperation* criam **OperationMultiplication** e **OperationSum**, respectivamente.

Parte 3

Utilize o padrão **Composite** para encapsular uma árvore de operações tratando objetos terminais **ResultLeaf** e composições de objetos **ResultComposite** de maneira uniforme através de uma interface comum **ResultComponent**. Ou seja, a árvore de operações é constituída de um nó superior (**ResultComposite**) que contém 2 subárvores inferiores: uma à esquerda e uma à direita. Cada uma dessas subárvores (**ResultComposite)** contém múltiplos nós do tipo folha (**ResultLeaf)**.

Parte 4

Utilize o padrão **Observer** para que alterações em **ResultComponent** propaguem-se atravaes de notificações apenas para os **ResultComponent** dependentes para manter os resultados de todos **ResultComponent** consistentes sem que seja necessário rerecalcular o resultado dos **ResultComponent** de toda a árvore. Para isso, **ResultComponent** deve implementar as *interfaces* **ISubject** e **IObserver**. A interface **ISubject** define os métodos **attach**(IObserver observer), **detach**(IObserver observer), e **notifyObservers**(). A interface **IObserver** define o método update().