

4ª Avaliação Individual – 2016.1

Instruções:

- Todos os códigos-fonte produzidos, exceto arquivos produtos da compilação (ex: .class), devem ser enviados em um único arquivo .zip para sandro.andrade@gmail.com ao final da avaliação.
- O e-mail deve obrigatoriamente ter o subject: INF011–20161-P4.

Questão 1) (6,0) Implemente o padrão *Observer* de modo a suportar cadeias de *observers* em modo *push*. Para isso um objeto deve poder atuar ao mesmo tempo tanto como *subject* quanto como *observer*. Realize a implementação utilizando as classes *Observer* e *Observable* do Java de modo que o código abaixo funcione:

```
MySubject s1 = new MySubject();  
Object so1 = new MySubjectObserver()  
Object so2 = new MySubjectObserver()  
s1.attach((Observer) so1);  
((Subject) so1).attach((Observer) so2);  
s1.changeState(); // Deve imprimir:
```

“s1 mudou estado para X, estou mudando o meu também para X” pelo objeto so1

“so1 mudou estado para X, estou mudando o meu também para X” pelo objeto so2

OBS: X deve ser uma String e pode assumir qualquer valor.

Pontuação:

- Codificação correta do *MySubject* (2 pontos)
- Codificação correta do *MySubjectObserver* (4 pontos)

Questão 2) (4,0) Implemente um *template method* para o processamento de formulários em sistemas web. O *template method* deve verificar se o usuário está autorizado a executar a operação (parte fixa), realizar a validação dos dados do formulário (parte variável), fazer o log da operação (parte fixa) e implementar a lógica de negócio (processamento propriamente dito – parte variável). Implemente um *main* contendo um exemplo (pode ser apenas com `System.out.println`) de validação e processamento.

- Implementação do *template method* (2 pontos)
- Definição dos métodos *hook* (1 ponto)
- Implementação dos métodos *hooks* e do *main* (1 ponto)

Boa sorte !