

1. A correta definição de arquitetura é fundamental em sistemas de softwares complexos, com efeitos na qualidade do produto de software gerado. Descreva:
  - (a) cada uma das camadas típicas da arquitetura de um sistema de informação, os seus propósitos e de que forma ocorre a comunicação entre elas em uma aplicação implementada em Java
  - (b) o conceito de operação de sistema, e em que contexto esse conceito se insere.
2. DDD (Domain Driven Design) define vários blocos de construção, que são padrões táticos para serem usados no desenvolvimento de aplicações orientadas a objetos. Descreva, dando exemplos:
  - (a) os padrões Value Object (Objeto Valor) e Entity (Entidade). Quais são suas semelhanças e diferenças? Para o primeiro, descreva detalhes acerca de aspectos de implementação que devem ser considerados.
  - (b) o padrão Repository (Repositório). Com que propósito ele deve ser usado? Que operações típicas são encontradas em classes desse tipo?
  - (c) os padrões Aggregate (Agregados) e Factory (Fábrica). Em que situações eles devem ser usados?
3. Em sala de aula, foi apresentada a classe `PeriodoLetivo` do estudo de caso SCA. Essa classe representa um período letivo. Por exemplo, o período letivo atual é o 2016.1. Faz sentido saber se um período é anterior ou posterior a outro período letivo. Por exemplo, 2016.1 é posterior a 2011.2. Faz sentido saber quais são os períodos letivos sucessor e antecessor de um período letivo fornecido. Por exemplo, para o período letivo atual, 2015.2 e 2016.2 são seus antecessor e sucessor, respectivamente. Um dos construtores dessa classe tem a assinatura `PeriodoLetivo(Integer ano, EnumPeriodo periodo)`. Uma parte da implementação da classe `PeriodoLetivo` relevante para esta questão é apresentada na figura a seguir.

```
public final class PeriodoLetivo implements Comparable<PeriodoLetivo> {  
    public enum EnumPeriodo {  
        PRIMEIRO, SEGUNDO;  
    };  
  
    final private Integer ano;  
  
    @Enumerated(EnumType.ORDINAL)  
    final private EnumPeriodo periodo;  
}
```

- (a) Implemente os métodos da classe `PeriodoLetivo` cujas assinaturas e objetivos são apresentados a seguir.

**int compareTo (PeriodoLetivo outro)**

Retornar um de três valores inteiros:

- 1, se o objeto `this` é anterior ao objeto `outro`
- 0, se `this` e `outro` representam o mesmo período letivo
- 1, se o objeto `this` é posterior ao objeto `outro`.

Se `outro` for null, deve disparar a exceção `IllegalArgumentException`

**PeriodoLetivo sucessor()**

Retornar o objeto `PeriodoLetivo` sucessor de `this`.

- (b) Considerando a utilização do framework JUnit, apresente um esboço de implementação da classe `PeriodoLetivoTest`, com quatro métodos de teste, dois para cada um dos métodos acima. Para cada método de teste, descreva o que está sendo testado.