

Aluno: \_\_\_\_\_

### 3ª Avaliação Individual – 2014.1

**Questão 1)** (2,0) Indique como o uso conjunto do *Observer* e do *Mediator* pode viabilizar a implementação de um servidor de listas de discussão. Apresente a API de todas as classes envolvidas. Apresente um diagrama e indique, através de setas, os objetos que tomam a iniciativa na comunicação.

**Questão 2)** (2,0) Você trabalha na *LazyDevelopers Inc.* implementando um jogo de corridas estilo *SuperTux Kart*. Cada carrinho do jogo realiza as seguintes operações: *correr*, *atirarArma* e *frear*. Cada carro do jogo, em um determinado momento, está em algum estado, por exemplo: estado normal, estado bombardeado, estado com baixa de combustível. As operações *correr*, *atirarArma* e *frear* devem ser executadas de modo diferente a depender do estado do carro. Por exemplo, um carro no estado bombardeado pode ter sua velocidade máxima reduzida à metade ou perder completamente o freio. Novos estados devem poder ser facilmente adicionados no futuro. Proponha uma solução para este problema. Apresente a API de todas as classes envolvidas.

**Questão 3)** (2,0) Qual o papel do *Template Method* na implementação de um *framework* ? Cite um exemplo prático real e inédito (código) de uso do *Template Method* em um *framework*.

**Questão 4)** (2,0) Apresente um exemplo inédito (código) de uso do *Strategy*.

**Questão 5)** (2,0) Sabe-se que o *Visitor* é um excelente padrão para suportar novas operações de visitação aos elementos de um agregado, porém acomodar novos tipos de elementos requer a modificação de todos os *visitors* concretos. Proponha uma solução para este problema. Indique qual(is) padrão(ões) foi(or)am utilizado(s) e apresente a nova API dos *visitors*.



"I'd keep an eye out on your new software developer. He just asked me how to turn the computer on."

Boa Sorte !