

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

### 3ª Avaliação Individual – 2015.1

**Questão 1)** (2,0) Você está desenvolvendo um jogo de guerra onde os jogadores devem ser notificados sobre mudanças na posição (coordenadas  $x$  e  $y$ ) de diversos tanques espalhados no campo de batalha. Novos tanques e novos jogadores devem poder ser adicionados em *run-time*, sem modificação no código-fonte. Na ocorrência de uma mudança de posição de um tanque, as novas coordenadas  $x$  e  $y$  devem ser notificadas (enviadas) a todos os jogadores. Apresente o diagrama de classes e a API da sua solução para este problema (1,0). Qual(is) padrão(ões) de projeto você utilizou? (0,5). Apresente um código exemplo do cliente (0,5).

**Questão 2)** (2,0) Ainda em relação ao jogo de guerra apresentado na questão 1, você precisa suportar diferentes algoritmos para a definição da estratégia de ataque. Deve ser possível incluir novos algoritmos e selecionar o seu uso em *run-time*, sem a interrupção do jogo. Complemente o diagrama anterior com as classes e APIs que compõem a sua solução (1,5). Qual(is) padrão(ões) de projeto você utilizou? (0,5).

**Questão 3)** (2,0) Apresente a API de um *Iterator* convencional e as APIs de um *Cursor* e sua classe *Collection* correspondente (1,5). Apresente as vantagens e desvantagens do uso do *Cursor* (0,5).

**Questão 4)** (2,0) Ainda em relação ao jogo apresentado na questão 1, percebeu-se a necessidade de flexibilizar algumas partes do funcionamento do jogo. A realização de uma partida deve seguir os seguintes passos: 1) animação fixa de introdução do jogo; 2) posicionamento fixo dos jogadores; 3) execução de uma estratégia variável de jogo; 4) execução de uma animação fixa de término; e 5) execução de um algoritmo variável de cálculo da pontuação obtida. Proponha uma solução para este problema, apresentando o diagrama de classes e as APIs envolvidas (1,5). Qual(is) padrão(ões) de projeto você utilizou? (0,5).

**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*.

Boa Sorte !