UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

DEPARTAMENTO DE CIÊNCIAS EXATAS

ÁREA DE INFORMÁTICA

MÓDULO: EXA 855 – PROJETO DE SISTEMAS

PROFESSOR: ROBERTO ALMEIDA BITTENCOURT

ALUNO(A):

PERÍODO: 2013.1 TURMA:

PROVA FINAL

Instruções:

* Esta prova NÃO admite consulta a livros, apostilas ou anotações ou qualquer outro material.
* A compreensão dos problemas nas questões faz parte da avaliação.

**1ª. QUESTÃO (2,5 pontos)**

Requisitos: Uma clínica odontológica da universidade oferece atendimento ambulatorial para a comunidade. Cada paciente, caracterizado, por nome e RG, tem um prontuário na clínica. No prontuário, além do registro lógico se o paciente tem ou não problemas de saúde (hipertensão, diabetes e alergia), será possível inserir anotações ao longo do tempo, através do computador. Cada anotação terá um texto e data/hora da inserção. Cada anotação é realizada sob a responsabilidade de uma dupla de estudantes (cada um com nome e número de matrícula), sendo cada dupla supervisionada por um dos professores da clínica. Cada professor supervisiona até cinco duplas.

QUESTÃO: Construa o modelo conceitual do problema explicitado através dos requisitos acima.

Obs: Atributos devem ter seus tipos de dados e relacionamentos devem ter seus nomes e cardinalidade. Você deve utilizar a notação correta em UML para diagramas de classes.

**2ª. QUESTÃO (2,5 pontos)**

Dada uma classe em Java chamada Fila, projetada com os métodos public void adicionar(Object o), public Object remover(), public int tamanho(), e public boolean estáVazia(), construa um uma classe de teste de unidade em JUnit que teste as operações da Fila. A classe de teste, que poderia ser chamada FilaTest, deve testar apenas a classe Fila (não importa para os testes se a classe Fila usa outras classes internamente), usando os métodos acima.

A classe de teste deve ter pelo menos os três seguintes métodos de teste:

1. Um método que teste apenas a correção dos métodos adicionar() e remover();
2. Um método que teste a correção do método tamanho(), podendo usar adicionar() e remover();
3. Um método que teste a correção do método estáVazia(), podendo usar adicionar() e remover().

Todos os métodos de teste devem testar uma variedade de situações que permitam dar confiança de que o teste trata as situações possíveis numa Fila.

(Continua no verso).

**3ª. QUESTÃO (2,5 pontos)**

Em um jogo bidimensional que você está construindo, uma *Nave* é um tipo de dados abstrato que tem atributos concretos de posição, velocidade e aceleração, além de métodos abstratos para virar à esquerda e virar à direita, e métodos concretos para conhecer o valor da posição e da velocidade. De uma *Nave*, são derivados tipos concretos, como *NaveDePasseio*, e outros tipos abstratos, como *NaveDeGuerra*. Uma *NaveDePasseio* tem métodos específicos como acelerar ou frear. Já uma *NaveDeGuerra* tem atributos adicionais como número de balas, e número de bombas, além de métodos abstratos como atirar e lançar uma bomba. Finalmente, tipos concretos como *NaveMae* e *NaveDeAtaque* são derivados do tipo *NaveDeGuerra*, devendo implementar todos os métodos abstratos, sendo que a NaveMae tem o atributo adicional de uma coleção de soldados. Uma *Nave* deve implementar a interface *Colidivel*, que possui um método chamado *colidiu* que recebe um outro objetoe retorna um valor lógico explicitando se a nave colidiu ou não com o objeto.

QUESTÃO: Faça um diagrama de classes em UML deste projeto. Atenção: esta questão será corrigida considerando a correção da sua descrição da solução do problema, mas também a correção na utilização da notação UML para diagramas de classes.

**4ª. QUESTÃO (2,5 pontos)**

Suponha que você precisa escrever, para um sistema de contagem de passagem de pessoas em um shopping center, uma nova classe Java chamada **Sensor** que deve funcionar como uma fonte de eventos de Passagem (**PassagemEvent**). Os eventos são disparados assim que uma pessoa passa e o sensor percebe, e são tratados pelo(s) **PassagemListener**(s) apropriados (que implementam o método public void pessoaPassou(PassagemEvent e)). Escreva todos os trechos do código em Java da classe **Sensor** (e apenas da classe Sensor) que implementa a fonte de eventos de passagem. (Obs: não é necessário se preocupar com aspectos de concorrência).