Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
AD1 de Programação III
1° semestre de 2008

Nome:
Matrícula:
Pólo:

Exercício:
O Ministério da Defesa de um determinado país contratou você para desenvolver um protótipo de sistema de informação em Java que cadastre os militares existentes em suas Forças Armadas:

(a) Crie uma classe abstrata **Militar** com um atributo numérico único (isto é, não existem dois militares de uma determinada divisão com mesmo número de matrícula) representando sua matrícula e outro atributo representando sua patente. As patentes existentes neste sistema, e que devem ser criadas por você, são as seguintes classes (apresentadas em ordem crescente de importância): Soldado, Cabo e Tenente. Garanta que todas as subclasses concretas de Militar implementem os métodos de impressão adequados a cada subclasse e a verificação de que um militar está habilitado a progredir na carreira;

(b) Implemente a classe concreta **MilitarAeronautica** que será usada para representar militares dessa divisão das Forças Armadas. Um militar da Aeronáutica está em condições de progredir se ele tem mais de dois anos naquela patente e se acumulou mais de 100 horas de vôo neste período;

(c) Implemente a classe concreta **MilitarExercito** que será usada para representar militares dessa divisão das Forças Armadas. Note que um militar do Exército está em condições de progredir se ele participou de uma guerra naquela patente e o seu país ganhou a guerra;

(d) Implemente a classe concreta **MilitarMarinha** que será usada para representar militares dessa divisão das Forças Armadas. Um militar da Marinha só progride na carreira se ele participou de um conserto de um navio em alto mar naquela patente;

(e) Utilize um vetor de militares para implementar um programa que:
- crie vários militares passando, como parâmetro de entrada, sua patente;
- implemente algumas situações de mudanças de patentes;
- apresente os dados de todos os militares que podem progredir de patente;
- modifique a patente dos militares que podem progredir; e
- apresente os dados de todos os militares da Aeronáutica.

```
=============
RESPOSTA (a):
=============
interface Patente{
  abstract public String toString();
}

class Soldado implements Patente{
  public String toString(){
    return "Soldado";
  }
}

class Cabo implements Patente{
  public String toString(){
    return "Cabo";
  }
}

class Tenente implements Patente{
  public String toString(){
    return "Tenente";
  }
}

abstract class Militar{
  private int id;
  private static int prox_id = 0;
  private Patente patente;

  Militar(){
    this.id = ++prox_id;
    patente = new Soldado();
  }

  Militar(Patente patente){
    this.id = ++prox_id;
    this.patente = patente;
  }

  Patente pegaPatente(){
    return patente;
  }

  void mudaPatente(Patente patente){
    this.patente = patente;
  }

  public String toString(){
    return "Matricula: " + id + " Patente: " + patente;
  }

  abstract boolean podeProgredir();

  abstract void mudaPatente();
}
```

```
=============
RESPOSTA (b):
=============
class MilitarAeronautica extends Militar{
  private int horas_voo;
  private int tempo_patente;

  MilitarAeronautica(Patente patente){
    super(patente);
    horas_voo = 0;
    tempo_patente = 0;
  }

  public String toString(){
    return "Classe: Aeronautica " + super.toString() + " Tempo de
Patente: " + tempo_patente + " Horas de voo: " + horas_voo;
  }

  boolean podeProgredir(){
    Patente aux = super.pegaPatente();
    if(aux instanceof Tenente) return false;
    if((horas_voo > 100) && (tempo_patente > 2)) return true;
    return false;
  }

  void mudaPatente(){
    Patente aux = super.pegaPatente();
    if(aux instanceof Tenente) return;
    if((horas_voo > 100) && (tempo_patente > 2)){
      horas_voo = tempo_patente = 0;
      if(aux instanceof Soldado) super.mudaPatente(new Cabo());
      if(aux instanceof Cabo) super.mudaPatente(new Tenente());
    }
  }

  void muda_horas_voo (int incr){
    horas_voo += incr;
  }

  void muda_tempo_patente (int incr){
    tempo_patente += incr;
  }
}

=============
RESPOSTA (c):
=============
class MilitarExercito extends Militar{
  private boolean participou_guerra;
  private boolean pais_ganhou_guerra;

  MilitarExercito(Patente patente){
    super(patente);
    participou_guerra = pais_ganhou_guerra = false;
  }

  void muda_participou_guerra (boolean valor){
```

```
    participou_guerra = valor;
  }


  void muda_pais_ganhou_guerra (boolean valor){
    pais_ganhou_guerra = valor;
  }

  public String toString(){
    return "Classe: Exercito " + super.toString() + " Participou
de Guerra? " + participou_guerra + " Pais ganhou Guerra? " +
pais_ganhou_guerra;
  }

  boolean podeProgredir(){
    Patente aux = super.pegaPatente();
    if(aux instanceof Tenente) return false;
    return (pais_ganhou_guerra && participou_guerra);
  }

  void mudaPatente(){
    Patente aux = super.pegaPatente();
    if(aux instanceof Tenente) return;
    if(pais_ganhou_guerra && participou_guerra){
      participou_guerra = pais_ganhou_guerra = false;
      if(aux instanceof Soldado) super.mudaPatente(new Cabo());
      if(aux instanceof Cabo) super.mudaPatente(new Tenente());
    }
  }
}
```

==========
RESPOSTA (d):
==========

```
class MilitarMarinha extends Militar{
  private boolean consertou_navio;

  MilitarMarinha(Patente patente){
    super(patente);
    consertou_navio = false;
  }

  boolean podeProgredir(){
    Patente aux = super.pegaPatente();
    if(aux instanceof Tenente) return false;
    return consertou_navio;
  }

  void mudaPatente(){
    Patente aux = super.pegaPatente();
    if(aux instanceof Tenente) return;
    if(consertou_navio){
      consertou_navio = false;

      if(aux instanceof Soldado) super.mudaPatente(new Cabo());
      if(aux instanceof Cabo) super.mudaPatente(new Tenente());
    }
```

```
  }

  void muda_consertou_navio (boolean valor){
    consertou_navio = valor;
  }

  public String toString(){
    return "Classe: Marinha " + super.toString() + " Consertou
navio? " + consertou_navio;
  }
}
```

==========
RESPOSTA (e):
==========

```
public class Teste{
  public static void main (String[] args){
    Militar vet_mil[] = new Militar[9];
    vet_mil[0] = new MilitarAeronautica(new Soldado());
    vet_mil[1] = new MilitarAeronautica(new Cabo());
    vet_mil[2] = new MilitarAeronautica(new Tenente());

    System.out.println("TESTE PARA MILITAR DA AREONAUTICA");
    for(int i = 0; i <= 2; i++){
      System.out.println(vet_mil[i]);
      ((MilitarAeronautica) vet_mil[i]).muda_horas_voo(150);
      ((MilitarAeronautica) vet_mil[i]).muda_tempo_patente(5);
    }

    System.out.println("\n\nAtualizando dados do Militar...\n");

    System.out.println("\nMilitares da Areonautica que podem
progredir na carreira");
    for(int i = 0; i <= 2; i++){
      if(vet_mil[i].podeProgredir()){
        System.out.println(vet_mil[i]);
        vet_mil[i].mudaPatente();
        System.out.println("Progressao --> " + vet_mil[i] + "\n");
      }
    }

    vet_mil[3] = new MilitarExercito(new Soldado());
    vet_mil[4] = new MilitarExercito(new Cabo());
    vet_mil[5] = new MilitarExercito(new Tenente());

    System.out.println("\n\n");
    System.out.println("TESTE PARA MILITAR DO EXERCITO");
    for(int i = 3; i <= 5; i++){
      System.out.println(vet_mil[i]);
      ((MilitarExercito) vet_mil[i]).muda_participou_guerra(true);
      ((MilitarExercito)
vet_mil[i]).muda_pais_ganhou_guerra(true);
    }

    System.out.println("\n\nAtualizando dados do Militar...\n");
```

```
    System.out.println("\nMilitares do Exercito que podem
progredir na carreira\n");
    for(int i = 3; i <= 5; i++){
      if(vet_mil[i].podeProgredir()){
        System.out.println(vet_mil[i]);
        vet_mil[i].mudaPatente();
        System.out.println("Progressao --> " + vet_mil[i] + "\n");
      }
    }

    vet_mil[6] = new MilitarMarinha(new Soldado());
    vet_mil[7] = new MilitarMarinha(new Cabo());
    vet_mil[8] = new MilitarMarinha(new Tenente());

    System.out.println("\n\n");
    System.out.println("TESTE PARA MILITAR DA MARINHA");
    for(int i = 6; i <= 8; i++){
      System.out.println(vet_mil[i]);
      ((MilitarMarinha) vet_mil[i]).muda_consertou_navio(true);
    }

    System.out.println("\n\nAtualizando dados do Militar...\n");

    System.out.println("\nMilitares do Exercito que podem
progredir na carreira\n");
    for(int i = 6; i <= 8; i++){
      if(vet_mil[i].podeProgredir()){
        System.out.println(vet_mil[i]);
        vet_mil[i].mudaPatente();
        System.out.println("Progressao --> " + vet_mil[i] + "\n");
      }
    }

    System.out.println("\n\n\nIMPRIMINDO SO OS MILITARES DA
AREONAUTICA...");
    for(int i = 0; i <= 2; i++) System.out.println(vet_mil[i]);
  }
}
```