Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação III
AP1 2° semestre de 2015.

Nome –

Assinatura –

Observações:

a) Prova sem consulta e sem uso de máquina de calcular.
b) Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
c) Você pode usar lápis para responder as questões.
d) Ao final da prova devolva as folhas de questões e as de respostas.
e) Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

Questão 1) (4.0 pontos)
Identifique no código abaixo as classes utilizadas, mas não definidas, e os métodos sem implementação:

```
public class TesteQuestao1 {
  public static void imprimirMensagens(final Mensagem[] msgs) {
    for (int i=0; i<msgs.length; i++) msgs[i].imprimir();
  }

     public static int contarReferenciasDuplicadas(final Mensagem[]
msgs) {
    //TODO: Implementar este método.
  }

  public static int contarMensagensDuplicadas(final Mensagem[] msgs)
{
    //TODO: Implementar este método.
  }

  public static void main(String args[]) {
    String textoEmail = "Olá,\n este é um e-mail de teste.";
    Email email = new Email(textoEmail);
```

```
    String textoChat = “Oi, vamos tc?”;
    FacebookChat chat01 = new FacebookChat(textoChat);
    FacebookChat chat02 = new FacebookChat(textoChat);
    Mensagem[] mensagens = new Mensagem[5];
    mensagens[0] = email;
    mensagens[1] = chat01;
    mensagens[2] = email;
    mensagens[3] = chat02;
    mensagens[4] = chat02;
    imprimirMensagens(mensagens);
    System.out.println(contarReferenciasDuplicadas(mensagens));
    System.out.println(contarMensagensDuplicadas(mensagens));
  }
}
```

Para esta questão:
(a) Escreva as três classes utilizadas pelo código fornecido, mas cuja implementação não é definida. **Atenção: observe o código fornecido para extrair dele quais métodos e construtores deverão estar presentes nas classes em questão.**

(b) Implemente os dois métodos incompletos da classe **TesteQuestao1**. Estes métodos devem retornar, respectivamente, a quantidade de referências e a quantidade de mensagens duplicadas nos arrays de mensagens passados como argumento para os métodos. **Atenção: Os métodos devem ser implementados de modo a tratar qualquer possibilidade de composição de mensagens nos arrays passados como argumento. Implementações específicas para o exemplo de composição apresentado acima serão consideradas incorretas.**
Dica: a classe **String** implementa o método **equals(String other)**, que retorna **true** caso o texto armazenado na instância corrente seja idêntico ao texto da instância referenciada por **other** e **false** caso contrário.

RESPOSTA:

```
abstract class Mensagem {
   protected String texto;
   public Mensagem(String texto) { this.texto = texto; }
   public String obterTexto() { return this.texto; }
   public void imprimir() { System.out.println(this.texto); }
}

class Email extends Mensagem {
   public Email(String texto) { super(texto); }
}
```

```
class FacebookChat extends Mensagem {
   public FacebookChat(String texto) { super(texto); }
}

public class TesteQuestao1 {
   public static void imprimirMensagens(final Mensagem[] msgs) {
      for (int i = 0; i < msgs.length; i++)
         msgs[i].imprimir();
   }
    public static int contarReferenciasDuplicadas (final Mensagem[]
msgs) {
      int total = 0;
      boolean[] ignorar = new boolean[msgs.length];
      for (int i = 0; i < msgs.length; i++)
         ignorar[i] = (msgs[i] == null);
      for (int i = 0; i < msgs.length - 1; i++) {
         if (!ignorar[i]) {
            boolean duplicado = false;
            for (int j = i + 1; j < msgs.length; j++) {
               if ((!ignorar[j]) && (msgs[i] == msgs[j])) {
                  ignorar[j] = true;
                  duplicado = true;
               }
            }
            if (duplicado) total++;
         }
      }
      return total;
   }

      public static int contarMensagensDuplicadas (final Mensagem[]
msgs) {
     int total = 0;
     boolean[] ignorar = new boolean[msgs.length];
     for (int i = 0; i < msgs.length; i++)
       ignorar[i] = (msgs[i] == null) || (msgs[i].obterTexto() == null);

     for (int i = 0; i < msgs.length - 1; i++) {
        if (!ignorar[i]) {
          boolean duplicado = false;
          for (int j = i + 1; j < msgs.length; j++) {
```

```
            if ((!ignorar[j]) &&
            (msgs[i].obterTexto().equals(msgs[j].obterTexto()))){
                ignorar[j] = true;
                duplicado = true;
              }
           }
           if (duplicado) total++;
         }
       }
       return total;
     }

     public static void main(String args[]) {
       String textoEmail = "Olá,\n este é um e-mail de teste.";
       Email email = new Email(textoEmail);
       String textoChat = "Oi, vamos tc?";
       FacebookChat chat01 = new FacebookChat(textoChat);
       FacebookChat chat02 = new FacebookChat(textoChat);
       Mensagem[] mensagens = new Mensagem[5];
       mensagens[0] = email;
       mensagens[1] = chat01;
       mensagens[2] = email;
       mensagens[3] = chat02;
       mensagens[4] = chat02;
       imprimirMensagens(mensagens);
       System.out.println(contarReferenciasDuplicadas(mensagens));
       System.out.println(contarMensagensDuplicadas(mensagens));
     }
}
```

Questão 2) (3.0 pontos)

a) Crie uma classe Telefone para representar números de telefone. Para dar suporte à generalidade e à expansões futuras, códigos internacionais (o do Brasil, por exemplo, é 55), códigos de área (o do Rio, também por exemplo, é 21) e o número do telefone devem ser armazenados separadamente, como strings.

b) Nesta classe, crie 3 construtores: 1o.) inicializando os 3 campos; 2o.) inicializando área e o número, deixando 55 como padrão para o código internacional; 3o.) inicializando o número, deixando 55 e 21 como padrão para os códigos internacional e de área, respectivamente.

c) Crie também um método toString() que retorne uma string no seguinte formato: "+55 21 78684527", onde 78684527 é o número do telefone.

d) Suponha que os telefones de uma dada cidade passem a ter os números precedidos do número 9, para números de celulares (números que iniciem com valores entre

6 e 9). Altere o(s) construtor(es) de forma a codificar esta adição. O número que começar com algum desses valores só pode ter 8 caracteres. Caso tenha 9, significa que o número já foi modificado. Dica: o método para cálculo do tamanho de uma string é *length()*, enquanto que para testar o início de uma string é *startsWith(String)*.

RESPOSTA:

```
class Telefone {
    String codInternacional;
    String codArea;
    String numero;
    public Telefone(String codInternacional, String codArea, String numero) {
            this.codInternacional = codInternacional;
            this.codArea = codArea;
            this.numero = numero;
            if (numero.length() == 8 &&
                    (numero.startsWith("6") ||
                     numero.startsWith("7") ||
                     numero.startsWith("8") ||
                     numero.startsWith("9")))
             this.numero = "9" + this.numero;
    }
    public Telefone(String codArea, String numero) {
            this("55", codArea, numero);
    }
    public Telefone(String numero) {
            this("55", "21", numero);
    }
    public String toString() {
       return "+" + codInternacional + " " + codArea + " " + numero;
    }
}
```

Questão 3) (3.0 pontos)
Considere a classe abaixo, a qual é utilizada num sistema que controla uma frota de carros para transporte:

```
class Carro {
       String motorista;
       String fabricante;
       String cor;
       boolean quatro_portas;

       public Carro(String motorista, String fabricante, String cor, boolean
quatro_portas) {
              this.motorista = motorista;
              this.fabricante = fabricante;
              this.cor = cor;
              this.quatro_portas = quatro_portas;
       }
}
```

Imagine que esta transportadora iria ampliar sua frota e também trabalhará com caminhões. Para este tipo de veículo deve-se armazenar: os nomes do motorista e do ajudante do motorista, o fabricante do caminhão, se a carroceria é do tipo baú e se possui cama para pernoite. Crie a classe Caminhão, podendo criar outra(s) de forma que características em comum sejam definidas apenas uma única vez.

RESPOSTA:

```
class Veiculo {
	String motorista;
	String fabricante;

	public Veiculo(String motorista, String fabricante) {
		this.motorista = motorista;
		this.fabricante = fabricante;
	}
}

class Carro extends Veiculo {
	String cor;
	boolean quatro_portas;

	public Carro(String motorista, String fabricante, String cor, boolean
quatro_portas) {
		super(motorista, fabricante);
		this.cor = cor;
		this.quatro_portas = quatro_portas;
	}
}

class Caminhao extends Veiculo {
	String ajudante;
	boolean bau;
	boolean cama_pernoite;


	public Caminhao(String motorista, String ajudante, String fabricante,
boolean bau, boolean cama_pernoite) {
		super(motorista, fabricante);
		this.ajudante = ajudante;
		this.bau = bau;
		this.cama_pernoite = cama_pernoite;
	}
}
```