



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação III
AP2 2º semestre de 2015.

Nome –

Assinatura –

Observações:

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

Questão 1) (4.0 pontos)

Considere um sistema que gerencia questões de provas. As classes abaixo implementam parte do sistema:

```
class Questao {
    private String enunciado;
    public Questao(String enunciado) {
        this.enunciado = enunciado;
    }
    public String getEnunciado() {
        return this.enunciado;
    }
}

class Discursiva extends Questao {
    private String gabarito;
    public Discursiva(String enunciado, String gabarito) {
        super(enunciado);
        this.gabarito = gabarito;
    }
    public String getGabarito() {
        return this.gabarito;
    }
    public String toString() {
        return this.getEnunciado() + "\n";
    }
}

class ItemVerdadeiroFalso extends Questao { // !!! COMPLETAR !!!
    private boolean gabarito;
}

class VerdadeiroFalso extends Questao { // !!! COMPLETAR !!!
```

```

}

public class AP2_2015_2_Q1 {
    public static void main(String[] args) {
        List<Questao> prova = new ArrayList<Questao>();
        prova.add(new Discursiva("Qual é a capital do Maranhão?", "Sao Luis"));
        List<ItemVerdadeiroFalso> itens = new ArrayList<ItemVerdadeiroFalso>();
        itens.add(new ItemVerdadeiroFalso("A linguagem de programação Java foi lançada há
20 anos atrás", true));
        itens.add(new ItemVerdadeiroFalso("Palmas não é a capital de Tocantins", false));
        itens.add(new ItemVerdadeiroFalso("Eclipse é um editor de texto multilinguagem",
true));

        Questao VouF = new VerdadeiroFalso(itens);
        prova.add(VouF);
        int num = 1;
        for (Questao q : prova) {
            System.out.print(num + " ");
            System.out.println(q.toString());
            num++;
        }
    }
}

```

Complete as classes ItemVerdadeiroFalso e VerdadeiroFalso para que o código funcione. Não é necessário reescrever nem modificar as outras classes na solução, apenas as pedidas. Dica: Preste atenção a todos os detalhes do código apresentado para que as classes sejam definidas corretamente.

RESPOSTA:

```

import java.util.ArrayList;
import java.util.List;

class ItemVerdadeiroFalso extends Questao {
    private boolean gabarito;

    public ItemVerdadeiroFalso(String enunciado, boolean gabarito) {
        super(enunciado);
        this.gabarito = gabarito;
    }

    public boolean getGabarito() {
        return this.gabarito;
    }

    public String toString() {
        return "( ) " + this.getEnunciado();
    }
}

class VerdadeiroFalso extends Questao {
    List<ItemVerdadeiroFalso> itens;

    public VerdadeiroFalso(List<ItemVerdadeiroFalso> itens) {
        super("Assinale verdadeiro ou falso para os itens:");
        this.itens = itens;
    }

    public String toString() {
        String texto = "";
        for (ItemVerdadeiroFalso item : itens)
            texto = texto + item.toString() + "\n";
        return this.getEnunciado() + "\n" + texto;
    }
}

```

Questão 2) (2.0 pontos)

Examine o código apresentado abaixo:

```
1 abstract public class Animal {
2     abstract public void greeting();
3 }
4
5 public class Cat extends Animal {
6     @Override
7     public void greeting() {
8         System.out.println("Meow!");
9     }
10 }
11
12 public class Dog extends Animal {
13     @Override
14     public void greeting() {
15         System.out.println("Woof!");
16     }
17
18     public void greeting(Dog another) {
19         System.out.println("Woouooooof!");
20     }
21 }
22 public class BigDog extends Dog {
23     @Override
24     public void greeting() {
25         System.out.println("Woow!");
26     }
27
28     @Override
29     public void greeting(Dog another) {
30         System.out.println("Woowwww!");
31     }
32 }
```

Indique o conteúdo que será impresso na saída padrão (i.e., qual linha imprime o quê) e os erros (i.e., qual linha leva a qual erro e qual o motivo) no seguinte programa de teste:

```

1 public class TestAnimal {
2     public static void main(String[] args) {
3         // Uso de subclasses
4         Cat cat1 = new Cat();
5         cat1.greeting();
6         Dog dog1 = new Dog();
7         dog1.greeting();
8         BigDog bigDog1 = new BigDog();
9         bigDog1.greeting();
10
11        // Uso de polimorfismo
12        Animal animal1 = new Cat();
13        animal1.greeting();
14        Animal animal2 = new Dog();
15        animal2.greeting();
16        Animal animal3 = new BigDog();
17        animal3.greeting();
18        Animal animal4 = new Animal();
19
20        // Uso de casting de referência
21        Dog dog2 = (Dog)animal2;
22        BigDog bigDog2 = (BigDog)animal3;
23        Dog dog3 = (Dog)animal3;
24        Cat cat2 = (Cat)animal2;
25        dog2.greeting(dog3);
26        dog3.greeting(dog2);
27    }
28 }

```

RESPOSTA:

```

linha 5: Imprime "Meow!"
linha 7: Imprime "Woof!"
linha 9: Imprime "Woow!"
linha 13: Imprime "Meow!"
linha 15: Imprime "Woof!"
linha 17: Imprime "Woow!"
linha 18: Erro! Animal é uma classe abstrata. Logo, não pode ser
instanciada.
linha 24: Erro! Impossível interpretar uma instância de Dog como Cat.
linha 25: Imprime "Wooooooof!"
linha 26: Imprime "Woowwww!"

```

Questão 3) (4.0 pontos)

Um polinômio pode ser representado usando uma lista armazenando-se o coeficiente e o expoente de cada termo usando uma classe auxiliar como a descrita abaixo:

```

class PolyTerm {
    protected int grau;
    protected double coef;
    protected PolyTerm prox;
    PolyTerm(int g, double c){
        grau = g;
        coef = c;
    }
    public int getGrau(){ return grau; }
    public double getCoef(){ return coef; }
}

```

Implemente a classe **Poly** (1 ponto) usando uma lista que suporte operações de soma (1 ponto), subtração (1 ponto) e multiplicação (1 ponto) de polinômios. A lista já possui disponível os métodos: **void insereInicio (PolyTerm p)**, **void insereFim (PolyTerm p)**, **PolyTerm removeInicio()**, **PolyTerm removeFim()**, **PolyTerm ConsultaElemento(int pos)**;

RESPOSTA:

```
import java.util.Scanner; //para testar a classe. Nao precisava...
```

```

class PolyTerm {
    protected int grau;
    protected double coef;
    protected PolyTerm prox;
    PolyTerm(int g, double c){
        grau = g;
        coef = c;
    }
    public int getGrau(){ return grau; }
    public double getCoef(){ return coef; }

    //para conferir a implementacao
    public String toString(){ return coef + "X^" + grau + " "; }
}

```

```
//ERA PARA SER IMPLEMENTADA...
```

```

class Poly{
    //ERA PARA SER IMPLEMENTADA...
    PolyTerm ini, fim;

    //ERA PARA SER IMPLEMENTADA...
    Poly(){ ini = fim = null; }

    //para conferir a implementacao
    public String toString(){
        String resp = "";
        PolyTerm p = ini;
        while(p != null){
            resp += p.toString();
            p = p.prox;
        }
        return resp;
    }
}

```

```

PolyTerm Copia(PolyTerm outro){
    return new PolyTerm(outro.grau, outro.coef);
}

//ja existia...
void insereInicio (PolyTerm p){
    PolyTerm copia = Copia(p);
    copia.prox = ini;
    ini = copia;
    if(fim == null) fim = ini;
}

//ja existia...
void insereFim (PolyTerm p){
    PolyTerm copia = Copia(p);
    copia.prox = null;
    if(fim == null){ ini = fim = copia; return; }
    fim.prox = copia;
    fim = copia;
}

//ja existia...
PolyTerm removeInicio(){
    if(ini == null) return null;
    PolyTerm resp = ini;
    ini = ini.prox;
    if(ini == null) fim = null;
    return resp;
}

//ja existia...
PolyTerm removeFim(){
    if(fim == null) return null;
    PolyTerm resp = ini, p = null;
    while(resp.prox != null){
        p = resp;
        resp = resp.prox;
    }
    if(p == null) ini = fim = null;
    else{
        p.prox = null;
        fim = p;
    }
    return resp;
}

//ja existia...
PolyTerm ConsultaElemento(int pos){
    if(ini == null) return null;
    PolyTerm p = ini;
    while((p != null) && (p.grau != pos)) p = p.prox;
    return p;
}

```

```

//ERA PARA SER IMPLEMENTADA...
void Soma(Poly p){
    PolyTerm x = p.ini;
    while(x != null){
        PolyTerm y = ConsultaElemento(x.grau);
        if (y == null) insereFim(x);
        else y.coef += x.coef;
        x = x.prox;
    }
}

//ERA PARA SER IMPLEMENTADA...
void Subtrai(Poly p){
    PolyTerm x = p.ini;
    while(x != null){
        PolyTerm y = ConsultaElemento(x.grau);
        if (y == null){
            PolyTerm z = Copia(x);
            z.coef *= -1.0;
            insereFim(z);
        }
        else y.coef -= x.coef;
        x = x.prox;
    }
}

//ERA PARA SER IMPLEMENTADA...
void Multiplica(Poly p){
    Poly resp = new Poly();
    PolyTerm p1 = ini, p2;
    while(p1 != null){
        p2 = p.ini;
        while(p2 != null){
            PolyTerm z = resp.ConsultaElemento(p1.grau + p2.grau);
            if(z == null){
                z = new PolyTerm(p1.grau + p2.grau, p1.coef * p2.coef);
                resp.insereFim(z);
            }
            else z.coef = z.coef + (p1.coef * p2.coef);
            p2 = p2.prox;
        }
        p1 = p1.prox;
    }
    this.ini = resp.ini;
}
}

```

```

//para conferir a implementacao
public class Q3_AP2_2015_2{
    public static void main(String[] args){
        int grau, i;
        double coef;
        Poly pol1 = new Poly(), pol2 = new Poly();
    }
}

```

```

Scanner sc = new Scanner(System.in);
grau = sc.nextInt();
for(i = 0; i <= grau; i++){
    coef = sc.nextDouble();
    PolyTerm x = new PolyTerm(i, coef);
    pol1.insereFim(x);
}
grau = sc.nextInt();
for(i = 0; i <= grau; i++){
    coef = sc.nextDouble();
    PolyTerm x = new PolyTerm(i, coef);
    pol2.insereFim(x);
}
sc.close();
pol1.Soma(pol2);
System.out.println(pol1);
System.out.println(pol2);
pol1.Subtrai(pol2);
System.out.println(pol1);
System.out.println(pol2);
pol1.Multiplica(pol2);
System.out.println(pol1);
System.out.println(pol2);
}
}

```