



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Programação III**

**AP1 1º semestre de 2010.**

**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) (3.0 pontos)**

Implemente um programa que receba, como parâmetro de entrada, um número **n** e um número **m**, e retorne, num vetor, todas as potências de **m**, menores que **n**. Por exemplo, se **n = 1000** e **m = 2**, o vetor retornado deve ser **v = {1, 2, 4, 8, 16, 32, 64, 128, 256, 512}**.

**RESPOSTA:**

```
public class Pot{

    public static void main(String[] args){

        int n = Integer.parseInt(args[0]), m = Integer.parseInt(args[1]);

        if((n != -1) && (n != 1) && (n != 0) && (m > 0)){

            int aux = 1, tam = 0;

            while(aux <= m){

                tam++;

                aux *= n;

            }

        }

    }

}
```

```

        int vet[] = new int[tam], i;

        aux = 1;

        for(i = 0; i< tam; i++){

            vet[i] = aux;

            aux *= n;

        }

        for(i = 0; i< tam; i++) System.out.print(vet[i] + " ");

        System.out.println();

    }

}
}

```

### Questão 2) (3.0 pontos)

Desenvolva um programa que receba, como parâmetros de entrada, os três coeficientes, a, b e c, de uma equação de 2º grau  $ax^2 + bx + c = 0$ , e retorne as raízes desta equação, se elas existirem.

### RESPOSTA:

```

public class Raizes{

    public static void main(String[] args){

        double a = Double.parseDouble(args[0]);
        double b = Double.parseDouble(args[1]);
        double c = Double.parseDouble(args[2]);

        if(a != 0){

            double delta = (b * b) - (4.0 * a * c);

            if(delta < 0)

                System.out.println("Nao existem raizes reais");

            else if(delta == 0){

                double x = (b * -1.0) / (2.0 * a);

```

```

        System.out.println("So existe uma raiz real igual a " + x);
    }

    else{

        double x1 = ((b * -1.0) + Math.sqrt(delta)) / (2.0 * a);
        double x2 = ((b * -1.0) - Math.sqrt(delta)) / (2.0 * a);

        System.out.println("Existem duas raizes reais: x' = " + x1 + "
x'' = " + x2);
    }

}

else{

    System.out.println("Erro: Nao e equacao de segundo grau");
}

}

}

```

### Questão 3) (4.0 pontos)

Suponha a classe Livro definida abaixo, a qual será utilizada num sistema que manipula informações de publicações:

```

public class Livro {
    int isbn;
    String titulo;
    String autor;
    String editora;
    java.util.GregorianCalendar datapublicacao;
    float preco_compra;
}

```

(Obs.: A classe `java.util.GregorianCalendar` é utilizada para a manipulação de datas em Java)

Além de livros, o sistema deve armazenar também informações de revistas (título, isbn, mês, ano e preço de compra), gibis (isbn, título, edição e preço de compra) e cds e dvds (issn, título, tamanho e preço de compra).

- Crie novas classes de forma que estas informações possam ser manipuladas.
- Implemente construtores que inicializem todos os atributos das classes.
- Imagine que, futuramente, pretendemos fazer um levantamento do valor total do acervo. Faça com que as classes criadas sejam obrigadas a definir um método para a obtenção do preço de um item (por exemplo, *float retornaPreco()*).

Reutilize construções, utilizando os mecanismos de O.O. em Java, sempre que possível. A classe fornecida também pode ser modificada.

**RESPOSTA:**

Explicações com respeito a esta possível solução são descritas ao longo do código abaixo, como comentários.

```
import java.util.ArrayList;
import java.util.GregorianCalendar;
import java.util.List;

// Implementa a interface necessária para qualquer publicação
interface IPublicacao {
    float retornaPreco();
}

// Classe abstrata que modela a estrutura básica de uma publicação
abstract class Publicacao implements IPublicacao {
    int cod;
    String titulo;
    float preco_compra;

    public Publicacao (int c, String t, float p) {
        cod = c; titulo = t; preco_compra = p;
    }

    public float retornaPreco() {
        return preco_compra;
    }
}

// Os campos isbn (cod), titulo e preço de compra passam a ser
// herdados da classe abstrata 'Publicacao'
class Livro extends Publicacao {
    String autor;
    String editora;
    GregorianCalendar datapublicacao;

    public Livro (int i, String t, String a, String e,
GregorianCalendar d, float p) {
        super(i, t, p);
        autor = a; editora = e; datapublicacao = d;
    }
}

class Revista extends Publicacao {
    int mes, ano;

    public Revista (int i, String t, float p, int m, int a) {
        super(i, t, p);
        mes = m;
        ano = a;
    }
}
```

```

class Gibi extends Publicacao {
    String edicao;

    public Gibi (int i, String t, float p, String e) {
        super(i, t, p);
        edicao = e;
    }
}

class CdDvd extends Publicacao {
    int tamanho;

    public CdDvd (int i, String t, float p, int tam) {
        super(i, t, p);
        tamanho = tam;
    }
}

// Classe principal que exemplifica o uso das classes declaradas
// Não é necessária na solução da questão !!
public class AP1_2010_1_Q3 {
    public static void main(String[] args) {
        List<Publicacao> publicacoes;
        publicacoes = new ArrayList<Publicacao> ();
        // Neste ponto ocorre a adição de livros, gibis, cds e dvds
        // Observe que para este trecho de código sempre funcionar,
        // os objetos dentro da coleção 'publicacoes' precisam
        // saber responder ao método 'retornaPreco()'
        float total = 0;
        for (Publicacao p : publicacoes) {
            total = total + p.retornaPreco();
        }
        System.out.println("Soma total de publicações: " + total);
    }
}

```