



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Departamento de Sistemas e Computação

Graduação em Ciência da Computação

Exercício de revisão de Java 01

Objetivo: Relembrar os conceitos de Java. Consulte a documentação da classe `java.util.Date`, quando necessário. Procure usar como fonte de estudo suas aulas do semestre passado (P2 e LP2). Este exercício não precisa ser entregue. Entretanto, é importante que você resolva com atenção pois diversos conceitos e práticas de programação serão essenciais no decorrer da disciplina. Dessa forma, se você tiver qualquer dúvida, procure o professor ou monitor. Utilize o horário da aula para resolver os exercícios.

1. (SEM ECLIPSE) Qual a diferença entre `overload` (sobrecarga) e `override` (sobreposição)? EXEMPLIFIQUE.
2. (SEM ECLIPSE) Quando usar o `override`? EXEMPLIFIQUE.
3. (SEM ECLIPSE) O que são classes abstratas e quais situações requerem seu uso? EXEMPLIFIQUE.
4. (SEM ECLIPSE) O que são *casts*? Para que servem *casts* e *instanceof*? EXEMPLIFIQUE.
5. (SEM ECLIPSE) No que interfaces diferem de classes abstratas? Que situações requerem o uso de interface e não de classes abstratas?
6. (SEM ECLIPSE) Considerando os trechos de código-fonte abaixo responda os itens de I até V.

```
public interface I1{
    public abstract void m3(String x);
}
public abstract class A implements I1{
    public abstract void m1(int x);
    public abstract void m2(double x);
}
public abstract class B extends A{
    public abstract void m1();
    public void m2(double x){
        System.out.println("m2 executado");
    }
}
```

```
public interface I2{
    public abstract void m4();
}
public abstract class C extends A
    implements I2{
    public void m1(int x){
        System.out.println("m1 executado");
    }
}
public class E extends C {
}
public class D extends B {
}
```

- I) Quais métodos (com nomes e parâmetros) deverão ser implementados pela classe E?
- II) Quais métodos (com nomes e parâmetros) deverão ser implementados pela classe D?
- III) O comando: `I1 x = new E();` vai dar erro de compilação? Por que?
- IV) Reescreva o trecho abaixo retirando os possíveis erros de *compilação* utilizando *casts*.

```
I1 x = new D();
x.m3("5");
x.m2(2.0);
E y = x;
I2 k = x;
```

- V) Considerando o trecho abaixo, qual o resultado produzido pelos comandos de impressão na tela?

```
I1 x = new E();
System.out.println(x instanceof B);
System.out.println(x instanceof C);
System.out.println(x instanceof A);
System.out.println(x instanceof I1);
System.out.println(x instanceof I2);
```

7. (COM ECLIPSE) Agora refaça a questão anterior usando o eclipse.
8. (COM ECLIPSE) Observe o trecho de código abaixo e, antes de escreve-lo no Eclipse, tente responder aos questionamentos em comentário e justificar o porque. Depois escreva-o no Eclipse e tente descobrir se voce acertou todas as respostas.

```
Date d1 = new Date();
Date d2 = d1;
d2.setDate(5);
System.out.println(d2.getDate()); //que valor vai ser impresso?
System.out.println(d1.getDate()); //que valor vai ser impresso?
d2 = null;
System.out.println(d1.getDate()); //que valor vai ser impresso?
System.out.println(d2.getDate()); //o que acontece?
```

9. (COM ECLIPSE) Conceitos matemáticos como funções, somatório e produtório, por exemplo, podem ser facilmente implementados em Java. A idéia é que cada estrutura matemática seja representada por uma construção (classe, metodo, comando, etc.) da linguagem Java. Implemente a seguinte função em Java.

$$f(n, m) = \sum_{i=1}^n i^2 + \prod_{j=1}^{m/2} j$$

onde

$$\sum_{i=1}^k e = e + e + \dots + e \text{ (} k \text{ vezes)} \quad \prod_{i=1}^k e = e * e * \dots * e \text{ (} k \text{ vezes)}$$

10. (COM ECLIPSE) Imagine que seu sistema vai lidar com dois tipos de produtos: produto perecível e não-perecível. Um produto perecível possui uma data de validade (pode ser tratada como uma String). O produto não-perecível possui uma informação extra sobre o acondicionamento do mesmo (também tratada como uma String). A árvore de herança abaixo mostra a estrutura das classes com herança. Implemente as classes ProdutoPerecivel e ProdutoNaoPerecivel utilizando herança

