

	Prova: Recuperação	Descontos:	Nota:
	Nome:		
	Data: 10/12/2012	Matrícula:	
	Disciplina: Algoritmos e Laboratório de Programação II	Professor: Gustavo Kuhn Andriotti	

Discriminação de descontos por descumprimento de regras:

Desconto	Justificativa

Regras:

1. Cabeçalho:

Não esqueça de preencher seu nome e matrícula, faltando qualquer um destes dados corre-se o risco de ter sua PROVA ANULADA.

2. Celulares:

O celulares deverão ser DESLIGADOS durante a prova. O aluno que não observar esta regra terá pontos descontados e corre o risco de ter sua PROVA ANULADA. O mesmo se aplica a celulares no modo silencioso.

3. Autenticidade:

Qualquer aluno pego copiando ou pedindo informações a colegas terá sua PROVA ANULADA e será imediatamente retirado da sala de aula. O mesmo se aplica a qualquer aluno conivente com a cópia, isto é, que não alertou o professor da tentativa de cópia.

4. Consulta:

Fica liberada toda e qualquer consulta ao material individual. Entretanto é VETADO o compartilhamento de qualquer material com terceiros. Qualquer desrespeito a esta regra implica na ANULAÇÃO DA PROVA de qualquer aluno envolvido.

5. Respostas:

As respostas podem ser fora de ordem, mas devidamente numeradas. Respostas à lápis serão DESCONSIDERADAS. Faça tudo à caneta.

1. **Em orientação à objetos existem 2 conceitos fundamentais: classes e objetos. Qual a principal diferença entre eles?**
2. **Explique a função dos atributos de um objeto. Além disso, qual a função dos métodos?**
3. **Qual a principal aplicação para interfaces (dê exemplos) e como estas estão relacionadas com herança de classes?**
4. **O conceito “Generics” (ou abstração de tipos) é um artifício de Java para produzir que tipo de abstração dentro de uma classe que o use (exemplo List)?**
5. **Produza o código para o método `compare()` (estrutura básica dada abaixo) para que se possa ordenar uma lista de inteiros de forma crescente colocando pares antes de ímpares (exemplo abaixo). O código não precisa estar 100% correto sintaticamente, mas o algoritmo sim.**

Antes da execução: (1,2,5,7,12,34,23,11)

Após execução: (2,12,34,1,5,7,11,23)

```
public class MeuComprator implements Comparator<Integer> {  
    /**  
     * resultado == 0 : o1 == o2  
     * resultado < 0  : o1 < o2  
     * resultado > 0  : o1 > o2  
     */  
    public int compare(Integer o1, Integer o2) {  
        int resultado = 0;  
        //TODO seu código  
        return resultado;  
    }  
}  
  
public void meuOrdenamento(List<Integer> entrada) {  
    Comparator<Integer> meuComp = null;  
    meuComp = new MeuComprator();  
    Collections.sort(entrada, meuComp);  
}
```

6. **Explique a diferença entre se efetuar a leitura de arquivos utilizando InputStream e Reader**
7. **Dado o teste abaixo, diga que método está sendo testado e qual o comportamento esperado. Encontre também um nome mais apropriado para o método.**

```
@Test
public void test() {
    Classe obj = null;
    String entrada = null;
    String saida = null;
    String esperado = null;

    obj = new Classe();
    entrada = "Esta é uma frase longa e com exceções";
    esperado = "a:4 c:3 e:7 f:1 g:1 l:1 m:2 n:1 o:3 r:1 s:3 t:1 u:1 x:1";
    saida = obj.metodo(entrada);
    Assert.assertEquals(esperado, saida);
}
```

8. **Implemente o método substituaCaracterPorContador() (estrutura básica dada abaixo) para que se atenda o requisito exemplificado abaixo. O código não precisa estar 100% correto sintaticamente, mas o algoritmo sim.**

```
entrada = "Esta é uma frase longa e com exceções";
caracter = 'a';
resultado = "Est1 é um2 fr3se long4 e com exceções";

public String substituaCaracterPorContador(String entrada, char caracter) {
    String resultado = null;
    //TODO seu código
    return resultado;
}
```