

Programação Orientada a Objetos

Bacharelado em Ciência da Computação

Prof. Dr. Eduardo Takeo Ueda

9ª Lista de Exercícios

(Entregar apenas o indicado com (1.0 ponto))

1. Implemente um programa Java que leia um arquivo texto chamado “**entrada.txt**” e imprima, em outro arquivo texto, denominado “**saida.txt**”, o total de letras, vogais, consoantes, espaços em branco, palavras e o total de linhas encontradas no primeiro arquivo.
2. Uma loja possui 4 filiais, cada uma com um código de 1 a 4. Um arquivo contendo todas as vendas feitas durante o dia nas quatro filiais é gerado a partir de uma planilha, sendo que cada linha do arquivo contém o número da filial e o valor de uma venda efetuada, separados por vírgula. Exemplo:
1, 189.90
1, 45.70
3, 29.90
4, 55.00

No exemplo acima, a filial 1 fez duas vendas, a primeira totalizando R\$ 189,90 e a segunda R\$ 45,70. A filial 3 fez uma venda de R\$ 29,90 e a 4 também uma de R\$ 55,00. Implemente um programa que leia este arquivo e informe, ao final, o total e o valor médio das vendas de cada filial.

3. Utilizando as classes `ObjectOutputStream`, `FileOutputStream` e `BufferedOutputStream`, crie um programa que escreva em um arquivo instâncias de um conceito Cachorro (idade - inteiro, peso - vírgula flutuante de dupla precisão, nome - cadeia de caracteres). Os objetos devem ser guardados como um todo, i.é, deve ser possível recuperá-los como unidades. Para tal, a classe Cachorro deve implementar a interface `Serializable`. Reabra o arquivo e apresente os valores anteriormente escritos (leia-os utilizando `ObjectInputStream`, `FileInputStream` e `BufferedInputStream`). As classes `ObjectInputStream` e `ObjectOutputStream` disponibilizam, respectivamente, métodos para leitura e escrita de objetos que sejam instâncias de tipos serializáveis: `readObject` e `writeObject`. O primeiro pode lançar `ClassNotFoundException`, `InvalidClassException`, `StreamCorruptedException`, `OptionalDataException`, `IOException`. O segundo pode lançar `InvalidClassException`, `NotSerializableException` e `IOException`. Não se esqueça de tratar as exceções e de fechar os arquivos.
4. Considere uma classe abstrata `Forma` que generaliza as classes concretas `Circulo`, `Quadrado`, `Retangulo` e `Triangulo`. Cada uma das classes concretas contém os seguintes parâmetros: `Circulo` - raio; `Quadrado` - lado; `Retângulo` - lado e altura; `Triângulo` - lado1, lado2 e lado3. Uma classe `ColecaoDeFormas` armazena várias formas em cada uma de suas instâncias. Implemente um programa que gere um arquivo binário contendo as informações de uma instância de `ColecaoDeFormas` e que seja capaz de recuperar suas informações do arquivo binário correspondente.
5. (1.0 ponto) Implemente uma classe chamada `Placa`, que encapsula o conceito de uma placa de carro, assumindo que ela tenha os seguintes atributos: o número da placa, o estado e sua cor. Implemente um programa cliente que crie três objetos `Placa`, escreva-os em um arquivo como objetos e depois leia-os a partir deste arquivo como objetos, imprima uma descrição de cada objeto usando o método `toString` (que a classe `Placa` deverá sobrescrever) e imprima o número de objetos. Ao ler os objetos, você deverá assumir que não conhece *a priori* o número de objetos no arquivo.