



## Lista de Exercícios 01 – Classes e Objetos CCF313 Programação Orientada a Objetos

### QUESTÕES

1) Criar as classes CondicionadorDeAr e Termostato.

a. A classe CondicionadorDeAr tem um atributo termostato do tipo Termostato e um atributo boolean chamado ligado. O construtor de CondicionadorDeAr não recebe nenhum parâmetro, mas instancia um termostato e coloca ligado em false. Crie um método de acesso para ligado e outro para termostato. Não precisa fazer os métodos modificadores. Crie um método ligar, que muda ligado para true, e um desligar, que muda ligado para false. Crie um método aumentarTemperatura, que aumenta a temperatura do termostato em um grau cada vez que é chamado até o limite de 28 graus. Crie um método reduzirTemperatura que reduz a temperatura em um grau cada vez que é chamado até o limite de 15 graus. Não se esqueça de verificar se o condicionador está ligado antes de aumentar ou diminuir a temperatura ou imprimi-la.

b. A classe Termostato tem um atributo privado temperatura. Seu construtor não recebe parâmetros, mas instancia a temperatura em 20. Crie métodos de acesso e modificador.

c. Crie a classe Usuario, com o método main. Neste método você deve instanciar um CondicionadorDeAr, aumentar a temperatura para 30 graus (receber mensagem de erro), reduzir a temperatura para 10 graus (receber mensagem de erro). Aumentar a temperatura para 25 graus e imprimir a temperatura.

2) Vetores de caracteres podem ser utilizados para representar cadeias de DNA. As quatro bases encontradas em um DNA são Adenina (A), Citosina (C), Guanina (G) e Timina (T). Dada as sequências da tabela a seguir, construa um programa que verifique em quais sequências a Adenina aparece mais que 2 vezes.

Sequência	Bases								
DNA 1	A	T	G	C	A	A	C	T	A
DNA 2	G	G	C	C	A	A	T	A	T
DNA 3	C	A	A	T	G	C	C	A	C
DNA 4	T	T	G	C	C	C	T	T	C

Para tanto, empregue as classes DNA e TesteDNA.

Classe DNA:

Atributo sequência (vetor de char, 9 caracteres)

Método construtor (sem parâmetros) deverá chamar o método lerDNA.

Método lerDNA (sem parâmetros de entrada e sem retorno): realiza a leitura, via teclado, das bases a serem armazenadas no atributo sequência.

Métodos get e set

Método toString: retorna uma String contendo todas as bases armazenadas no atributo sequência.

Método obterFrequencia: tem como parâmetro de entrada um caractere que corresponde a uma base. Retorna a frequência (número de ocorrências) da base no atributo sequência.

3) Diz-se que uma dada sequência de DNA possui um certo grau de similaridade de acordo com o número de bases que aparecem na mesma ordem. Por exemplo as duas sequencias da figura abaixo possuem similaridade 3.

Sequência	Bases					
DNA 1	A	T	C	G	T	C
DNA 2	A	C	T	G	T	T
	=	=	=	=	=	=
Similar	1	0	0	1	1	0

Modifique o exercício anterior de modo que um novo método, denominado verificarSimilaridade, deverá ter como parâmetro de entrada um objeto DNA. O método deverá comparar a sequência do parâmetro de entrada com a sequência do objeto que invocou o método. O método deverá retornar o grau de similaridade das duas sequências.

Considere as seguintes sequências de DNA:

Sequência	Bases								
DNA 1	A	T	G	C	A	A	C	T	A
DNA 2	G	G	C	C	A	A	T	A	T
DNA 3	C	A	A	T	G	C	C	A	C
DNA 4	T	T	G	C	C	C	T	T	C

Empregando o método verificarSimilaridade forneça os valores de similaridade entre as sequências de DNA da tabela a seguir:

	DNA 1	DNA 2	DNA 3	DNA 4
DNA 1	9			
DNA 2		9		
DNA 3			9	
DNA 4				9

4) Crie dois pacotes conforme descrito a seguir:

**Pacote: br.ufv.universidade**

- Crie a classe Professor com seu construtor, métodos de acesso e modificadores e os atributos privados nome (String), idade (int). Crie o método toString que retorna os valores dos atributos (String).
- Crie a classe Disciplina com seu construtor, métodos de acesso e modificadores e os atributos privados nome (String), prática (boolean). Crie o método toString que retorna os valores dos atributos (String).
- Crie a classe Atribuicao com seu construtor, métodos de acesso e modificadores e os atributos privados professor, do tipo Professor, e disciplina, do tipo Disciplina. Crie o método toString que retorna os valores dos atributos (String).

**Pacote: br.ufv.teste**

- Crie a classe TesteAtribuicao com o método main que instancia um Professor, uma Disciplina e uma Atribuicao. Imprima os dados da Atribuicao.

5) Crie uma classe Texto contendo os seguintes métodos:

- Método contarVogais que recebe como parâmetro um String retorna o número de vogais maiúsculas e minúsculas da String.
- Método conjugarVerbo que recebe como parâmetro um verbo regular terminado em AR. O método deve retornar uma String contendo a conjugação do verbo no presente.

Exemplo:

Verbo: andar  
 Eu ando  
 Tu andas  
 Ele anda

Nós andamos

Vós andais

Eles andam

- c. Método obterInverso que recebe uma String como parâmetro e retorna a String de trás para frente.
- d. Método que recebe uma String como parâmetro e retorna verdadeiro caso a String seja um palíndromo, e falso caso contrário. (Um palíndromo é uma palavra que tem a propriedade de poder ser lida tanto da direita para a esquerda como da esquerda para a direita. Exemplos: ovo, arara, rever, asa, osso)

Crie uma classe de Teste e utilize todos os métodos da classe Texto.

6) Crie as seguintes classes:

- a. Crie a classe Etiqueta com os atributos privados marca e tamanho, ambos do tipo String. Crie seu construtor, que recebe parâmetros para inicializar os atributos. Crie os métodos de acesso e modificadores.
- b. Crie a classe Produto com os atributos privados nome, do tipo String, preço, do tipo double e etiqueta, do tipo Etiqueta. Crie um construtor, que receba o nome, preço e um objeto Etiqueta como parâmetros. Crie outro construtor, que receba o nome e preço do produto, e também a marca e tamanho da etiqueta como parâmetros (o objeto Etiqueta deve ser instanciado no construtor). Crie os métodos de acesso e modificadores.
- c. Crie uma classe TesteProduto com um método main, onde você deve ler os dados de um produto, instanciar um produto e exibir na tela os atributos do produto criado. Leia os dados de outro produto, instancie outro produto e exiba na tela os atributos do produto criado (utilize os dois construtores). Depois altere os atributos do produto criado anteriormente. Exiba novamente na tela os dados do produto.