



UNIVERSIDADE FEDERAL DO CEARÁ

Abril de 2024

Disciplina: Programação Orientada à Objetos
Professor: Filipe Fernandes dos Santos Brasil de Matos

Semestre: 2024.1

Aluno: _____ **Matrícula:** _____

Instruções para uma boa prova:

- A avaliação é individual e não é pesquisada. É proibida a comunicação e a troca de materiais entre alunos, bem como consultas a materiais bibliográficos, cadernos ou anotações de qualquer espécie;
- Use caneta esferográfica de tinta azul ou preta para responder às questões. Questões respondidas a lápis serão corrigidas, porém não serão aceitas reclamações sobre as correções;
- O aluno só poderá entregar a prova após 40 minutos do início de sua realização. Após esse momento, nenhum outro aluno poderá entrar para realizar a prova;
- Coloque seu nome e matrícula também nas folhas de respostas. Essa identificação é importante para vincular as respostas aos alunos que a produziram;
- Por favor, escreva com letra legível. Serão corrigidas somente as respostas compreensíveis;
- Um arquivo com os código-fontes deverão ser comprimidos em formato zip e enviados para mim via SIGAA ao final da prova. Neste momento, chame o professor para acompanhar o processo de envio do material.
- O arquivo com os código-fontes deve estar estruturado da seguinte maneira:
 - a. `\[matricula]\` (pasta raiz com o numero da matrícula do aluno)
 - b. `\[matricula]\qX\` (um subpasta para cada questão prática, onde X é o número da questão)
 - c. Dentro de `\[matricula]\qX\` você adiciona todos arquivos que respondem a questão X. **ATENÇÃO** as particularidades de cada questão.
 - d. Não é necessário inserir arquivos `.class` e classes `Main` nas respostas das questões

1ª Atividade Avaliativa Parcial (1ª AP)

1. Responda as perguntas abaixo resumidamente: (2.5 ponto)
 - a. Qual é a diferença entre uma classe e um objeto em Java?
 - b. Quais são as diferenças entre atributos em Java e variáveis locais?
 - c. O que são construtores em Java e por que são usados?
 - d. O que é encapsulamento em Java e por que é importante?
 - e. Quais são as vantagens de usar pacotes em Java?
2. Imagine que você está desenvolvendo um sistema para uma empresa de telefonia na cidade de Crateús. Desenvolva uma possível classe em Java desse sistema.

Sua classe deve ter dois atributos e um método que opera sobre esses atributos. Explique resumidamente sob que contexto sua classe seria útil ao sistema. OBS: O métodos deve ser diferente de métodos construtores e gets/sets (1.0 ponto)

3. Replique o código abaixo e implemente o método *ordenaVetor()* que recebe, como entrada, um vetor de N elementos e retorna, como saída, o vetor ordenado: (1.5 pontos)

```
public class Questao3 {  
    public static int[][] ordenaVetor(int[] vet1) {  
        // Implemente o código-fonte aqui!  
    }  
}
```

4. Crie um pacote transporte e insira dentro dele as seguintes classes: (2.0 pontos)

a. Caminhão:

i. Atributos (todos privados):

1. Marca (String), Ano (int) e Quilometragem (double)

ii. Métodos:

1. Gets/Sets de cada um dos atributos supracitados

b. Carro:

i. Atributos (todos privados):

1. Modelo (String), Número de Lugares (int) e Preço (double)

ii. Métodos

1. Gets/Sets de cada um dos atributos supracitados

c. Bicicleta:

i. Atributos (todos privados):

1. Cor (String), Número de Marchas (int) e Preço (double)

ii. Métodos

1. Gets/Sets de cada um dos atributos supracitados

5. Crie a classe Horario cuja instância (objeto) represente um horário: (3.0 pontos)

a. Atributos:

- i. Hora (int), Minuto (int) e Segundo (int);

b. Construtores:

- i. Padrão - Objeto criado com um valor padrão 23:59:59

- ii. Três parâmetros - Objeto criado com a hora, o minuto e o segundo informados (nessa ordem) no construtor

c. Métodos:

- i. *valida()* - Retorna true se o horário corrente for válido e false se o horário corrente for inválido. OBS: Assuma hora no intervalo 0 e 23.
- ii. *formata()* - Retorna uma String no formato "HH:MM:SS" do horário corrente, caso válido. Caso o horário corrente seja inválido, retorne uma String com a mensagem "Horário Inválido!"
- iii. *compara()*: Recebe como entrada um outro objeto da Classe Horario e o compara com o horário corrente e retorna: 0 se os horários forem iguais; 1 se o horário corrente for maior que a do parâmetro; -1 se o horário do parâmetro for maior que a corrente.