

	Universidade Federal de Rondônia
	Departamento de Ciências de Computação – DACC/NT-PVH
	Disciplina: Programação Orientada a Objetos – POO
	Código: INF31098/INF31030
	Prof. Dr. Jonathan Ramos
	Semestre: 2022.1: Trabalho 01/Lista de Exercícios

Trabalho 01/Lista de Exercícios

Leia as Instruções:

- Usar boas práticas do JAVA: Encapsular todos os atributos da classe (criar os métodos set e get), definir construtores vazios e com parâmetros, documentar o código onde for necessário. Adicionar seu nome no JavaDoc de todas as classes.
- Onde há necessidade de implementação de códigos, criar um projeto separado para cada exercício (não usar pacotes).
- Os exercício com códigos devem vir testados no main com tudo que for necessário para teste já digitado, com definição de atributos e tudo mais! Se o código não rodar ou não mostrar nada **NÃO ECLIPSE**, a nota será zero. Portanto, teste o bom funcionando do código no eclipse antes de finalizá-lo.
- Este trabalho valerá 20% da nota final. **Exercícios plagiados recebem zero.**
- Dúvidas e resolução de alguns dos exercícios nas aulas do dia 10/10/2022 e no sábado dia 08/10/2022 (gmeet <https://meet.google.com/akm-ugdk-fth>) das 7:50h às 12:00h.
- Primeira prova Bimestral será no dia 17/10/2022: Permitido qualquer material impresso. Qualquer aparelho eletrônico deverá ser desligado. Não será no laboratório, mas sim na sala 106 Bloco 1G.
- Este trabalho deverá ser entregue no SIGAA como um único arquivo .zip contendo todos os exercícios. Nomear o arquivo vide exemplo: JonathanSilvaRamosLista01.zip
- Se fizer a lista, tens altíssimas chances de ir bem na prova. Boa sorte!

Questão 1 (0,2 pontos) Com suas palavras, explique o que é um objeto e o que é uma classe em Java. Dê um ou mais exemplos.

Questão 2 (0,2 pontos) Com suas palavras, explique quais as vantagens e desvantagens de POO com relação a programação estruturada/procedural. Se possível, dê um ou mais exemplos.

Questão 3 (0,2 pontos) Qual a vantagem de usar um IDE (*Integrated Development Environment*)?

Questão 4 (0,2 pontos) Explique, com suas palavras, qual a principal função do Polimorfismo nas classes de POO. Dê pelo menos um exemplo.

Questão 5 (0,2 pontos) Para que servem as Listas em Java? Que tipos de dados recebem? Quais funcionalidade nos proporciona?

Questão 6 (0,2 pontos) Considerando os tipos de dados que podemos usar para os atributos de cada classe, escolha o melhor tipo para representar:

- A altura de uma pessoa em metros.
- O peso de uma pessoa em quilos.
- A temperatura corporal de uma pessoa.
- O sexo de uma pessoa.
- A altura de uma pessoa em milímetros.

- (f) O número de municípios de um estado do Brasil.
- (g) O nome de um estado do Brasil.
- (h) A população de um estado do Brasil.
- (i) A área do Brasil em quilômetros quadrados.
- (j) A população total do mundo.
- (k) O CEP de um endereço no Brasil.
- (l) O nome de uma rua em um endereço no Brasil.

Questão 7 (0,2 pontos) Identifique e explique o(s) erro(s) na classe abaixo:

```
class DoisValores {
    int valor1, valor2;
    int maior() {
        if (valor1 > valor2)
            return true;
        else
            return false;
    }
    void menor() {
        if (valor1 < valor2)
            return valor1;
        else
            return valor2;
    }
}
```

Questão 8 (0,2 pontos) Identifique e explique o(s) erro(s) na classe abaixo:

```
class NumeroComplexo {
    float real, imaginário;
    float valor(){
        return real, imaginário;
    }
}
```

Questão 9 (0,4 pontos) Crie uma classe **Data** com os seguintes atributos: **dia**, **mes** e **ano**. No main, instanciar essa classe com uma data válida. Em seguida, criar um método dentro da classe **Data** **vemAntes(dia, mes, e ano)**, que compara se a data passada vem antes ou não da data existente na classe. Além disso, verificar no construtor e nos gets e sets se o dia, mes e ano passados como parâmetros são válidos. Fazer uma função que verifica se a data passada no **vemAntes()** é válida, para que a comparação não fique enviesada.

Questão 10 (0,5 pontos) Represente uma pessoa (classe), com os atributos privados de nome (**String**), data de nascimento (**Date**) e altura (**float**).

(a) Crie os métodos públicos necessários para sets e gets e também um método para imprimir todos os dados de uma pessoa (**toString()**).

(b) Crie um método para calcular a idade da pessoa, o qual será exibido no **toString()**.

Questão 11 (0,5 pontos) Vamos criar uma classe chamada **Lampada** que possui três estados: acesa, apagada, e meia-luz:

(a) criar variável (atributo da classe) que aceita os três estados dentro da classe;

(b) criar as funções (métodos): **acende()**, **apaga()**, **imprimirEstado()**.

Questão 12 (0,5 pontos) Crie uma classe chamada **ContaBancariaSimplificada**, na qual há três atributos: **nomeDoCorrentista**, **saldo**, e **éContaEspecial**. Está é uma representação de uma conta bancária simplificada, que somente representa o nome do correntista, o saldo da conta e se a conta é especial ou não. Se a conta for especial, o correntista terá o direito de retirar mais dinheiro do que tem no saldo (ficar com o saldo negativo).

- (a) Permitir a abertura da conta com todas as variáveis da classe e com apenas o nome do cliente.
- (b) Criar métodos que **deposita()**, **retira()**, e **mostraDados()**.
- (c) Exibir mensagem de alerta caso o saldo do cliente for ficar negativo.

Questão 13 (0,5 pontos) Escreva uma classe cujos objetos representam alunos matriculados em uma disciplina. Cada objeto dessa classe deve guardar os seguintes dados do aluno: matrícula, nome, 2 notas de prova e 1 nota de trabalho. Escreva os seguintes métodos para esta classe:

- (a) **media**: calcula a média final do aluno (cada prova tem peso 2,5 e o trabalho tem peso 2)
- (b) **final**: calcula quanto o aluno precisa para a prova final (retorna zero se ele não for para a final).

Questão 14 (0,5 pontos) Escreva uma classe **Data** cuja instância (objeto) represente uma data. Esta classe deverá dispor dos seguintes métodos:

- (a) **construtor** define a data que determinado objeto (através de parâmetro), este método verifica se a data está correta, caso não esteja a data é configurada como 01/01/0001
- (b) **compara** recebe como parâmetro um outro objeto da Classe data, compare com a data corrente e retorne:
 - 0 se as datas forem iguais
 - 1 se a data corrente for maior que a do parâmetro;
 - -1 se a data do parâmetro for maior que a corrente.
- (c) **getDia** retorna o dia da data
- (d) **getMes** retorna o mês da data
- (e) **getMesExtenso** retorna o mês da data corrente por extenso
- (f) **getAno** retorna o ano da data
- (g) **isBissexto** retorna verdadeiro se o ano da data corrente for bissexto e falso caso contrário
- (h) **clone** o objeto clona a si próprio, para isto, ele cria um novo objeto da classe Data com os mesmos valores de atributos e retorna sua referência pelo método

Questão 15 (0,5 pontos) Gere uma classe **Agenda** que armazena 10 pessoas e que realize as operações:

- (a) **void armazenaPessoa(String nome, int idade, float altura);**
- (b) **void removePessoa(String nome);**
- (c) **int buscaPessoa(String nome);** Informa em que posição da agenda está a pessoa
- (d) **void imprimeAgenda();** Imprime os dados de todas as pessoas da agenda
- (e) **void imprimePessoa(int index);** Imprime os dados da pessoa que está na i-ésima posição da agenda.

Questão 16 (0,5 pontos) Gere uma classe chamada **Elevador** para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo é 0), total de andares no prédio (desconsiderando o térreo), capacidade do elevador e quantas pessoas estão presentes nele. A classe deve também disponibilizar os métodos:

- (a) **Inicializa**: que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);
- (b) **Entra**: para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
- (c) **Sai**: para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
- (d) **Sobe**: para subir um andar (não deve subir se já estiver no último andar);

(e) **Desce**: para descer um andar (não deve descer se já estiver no térreo);

Questão 17 (0,5 pontos) Crie uma classe **Televisao** e uma classe **ControleRemoto** que pode controlar o volume e trocar os canais da televisão. O controle de volume permite:

- (a) Aumentar ou diminuir a potência do volume de som em uma unidade de cada vez;
- (b) Aumentar e diminuir o número do canal em uma unidade
- (c) Trocar para um canal indicado;
- (d) Consultar o valor do volume de som e o canal selecionado.

Questão 18 (1 pontos) Escreva uma classe em que cada objeto representa um voo que acontece em determinada data e em determinado horário. Cada voo possui no máximo 10 passageiros (para facilitar), e a classe permite controlar a ocupação das vagas. A classe deve ter os seguintes métodos:

- (a) **construtor**: configura os dados do voo (recebidos como parâmetro): número do voo, data (para armazenar a data utilize um objeto da classe Data, criada na questão anterior);
- (b) **proximoLivre**: retorna o número da próxima cadeira livre
- (c) **verifica**: verifica se o número da cadeira recebido como parâmetro está ocupada
- (d) **ocupa**: ocupa determinada cadeira do voo, cujo número é recebido como parâmetro, e retorna verdadeiro se a cadeira ainda não estiver ocupada (operação foi bem sucedida) e falso caso contrário
- (e) **vagas**: retorna o número de cadeiras vagas disponíveis (não ocupadas) no voo
- (f) **getVoo**: retorna o número do voo
- (g) **getData**: retorna a data do voo (na forma de objeto)
- (h) **clone**: o objeto clona a si próprio, para isto, ele cria um novo objeto da mesma classe e faz uma cópia dos valores de seus atributos

Questão 19 (1 pontos) Considere a seguinte classe, cujo método **respostaQuestao** recebe como parâmetro o número de uma questão e retorna a sua resposta correta, proveniente de um gabarito.

```
public class Gabarito {  
    public char respostaQuestao(int numeroQuestao) {  
        // Códigos aqui  
    }  
}
```

Escreva uma classe **Prova** em que cada objeto representa uma prova feita por um aluno. Esta prova possui 15 questões de múltipla escolha (letras A a E). As 10 primeiras questões valem 0,5 ponto e as 5 últimas questões valem 1 ponto. Esta classe deverá controlar as questões respondidas pelo aluno. Para isto, a classe deve implementar os métodos:

- (a) **construtor**: recebe como parâmetro um objeto da classe Gabarito contendo o gabarito da prova
- (b) **respostaAluno**: recebe como parâmetro a resposta dada pelo aluno a uma questão; este método não recebe entre os parâmetros o número da questão, ele mesmo deve estabelecer um controle interno de modo que as questões sejam inseridas sequencialmente, ou seja, a primeira vez que o método é chamado, insere a primeira questão, a segunda, insere a segunda questão, e assim por diante.
- (c) **acertos**: retorna a quantidade de questões que o aluno acertou
- (d) **nota**: retorna a nota que o aluno tirou na prova
- (e) **maior**: recebe como parâmetro um outro objeto da classe Prova e retorna a nota do aluno que acertou mais questões; se houver empate, retorna a maior nota; se houver empate novamente, retorna a nota do aluno representado no objeto corrente

Questão 20 (2 pontos) Use o exemplo da classe Venda feita em sala para incrementar o exercício 18. Porém, agora o produto que é vendido será a vaga no voo (atributos de produto mudará, assim como outras coisas deverão ser adaptadas para encaixar no caso em questão).

(a) Ao fazer uma nova venda, verificar se há disponibilidade no voo e se a vaga escolhida está disponível, exibir mensagens de erro.

(b) O preço de cada passagem será definido por um valor base de mil reais + uma taxa de 500 se o voo for feito em horário comercial (das 8 as 18). Além disso, um adicional entre 1 e 10% é adicionado aleatoriamente sobre o valor total da passagem.

(c) Criar funções métodos de relatórios para o administrador na classe do Voo: valor total vendido, nome dos passageiros e seus respectivos assentos, horário do voo etc.

(d) Além disso, deve haver uma opção (método) para o usuário ver os dados do próprio voo, horário, valor pago etc.