

Lista de exercícios (2017.1)

Para cada exercício prático, crie um novo projeto/diretório. Lembre-se de documentar suas classes, e de escrever uma classe para testar seus programas.

1. Defina com suas palavras classe e objeto?
2. Quais as partes integrantes de uma classe?
3. O que são o estado e o comportamento de um objeto? Onde eles são definidos?
4. Quais os tipos primitivos da linguagem Java?
5. Qual o nome do arquivo que contém a classe abaixo?

```
public class UmaClasse
{
    public void imprimir()
    {
        System.out.println("Bla!!!");
    }
}
```



6. Qual o nome do arquivo gerado com a compilação da classe UmaClasse?



7. Escreva um método em Java que mostre na tela os seus dados pessoais e os dados do curso que você frequenta.

8. Para declarar uma variável inteira com o valor inicial 2, você escreve:

- (a) `int number = 2L;`
- (b) `int number = 2l;`
- (c) `int number = 2;`
- (d) `int number = 2.0;`

9. Para atribuir uma variável double d para uma variável inteira x, você escreve:

- (a) `x = (long)d;`
- (b) `x = (int)d;`
- (c) `x = d;`
- (d) `x = (float)d;`

10. Qual o valor impresso pelo método abaixo? Por quê?

```
public void calcula(){
    int i;
    i = 30;
    float f;
    f = i/4;
    System.out.println(f);
}
```

11. O que será impresso pelo código a seguir? Desenhe a heap para melhor entender.

```
Character c1 = new Character('a');
Character c2 = c1;
Character c3 = new Character('b');
c1 = c3;
c3 = c2;
c2 = new Character('c');
c1 = c2;
System.out.println("O valor de c1 é " + c1.charValue());
System.out.println("O valor de c2 é " + c2.charValue());
System.out.println("O valor de c3 é " + c3.charValue());
```

12. Dado o seguinte trecho de código em JAVA:

```
Lampada l1 = new Lampada();
Lampada l2 = new Lampada();
l1.ligar();
l2 = l1;
l1.desligar();
if (l2.estaLigada())
    System.out.println("l2 está ligada");
else
    System.out.println("l2 está desligada");
```

responda:

- (a) O que será impresso? por quê?
- (b) Desenhe o heap e a referência das variáveis l1 e l2 após a execução de cada linha.

13. Considerando a String abaixo, responda:

```
String texto = "A turma do IMD está aprendendo Strings.";
```

- (a) Qual o valor retornado por `texto.length()`?
- (b) Qual o valor retornado por `texto.charAt(12)`?
- (c) Qual o valor retornado por `texto.indexOf('S')`?

14. Escreva métodos que receba uma String como parâmetro e:

- (a) Informe qual o tamanho da String.
- (b) Informe qual o primeiro e o último caracter.
- (c) Informe quantas vogais existem.
- (d) Informe quantas palavras existem.

15. Considere a seguinte classe

```
public class Car
{
    public int numberOfSeats;
    private String model;
    private int engineCode;

    public Car(String model)
    {
        model = model;
    }

    public int getSeats()
    {
        return numberOfSeats;
    }

    private String getModel()
    {
        return model;
    }

    public void setEngineCode(int code)
    {
        int n = code * 2;
        if(code >= 100) {
            engineCode = n;
        }
        else {
            engineCode = code;
        }
    }
}
```

Agora considere os 10 tópicos abaixo. Para cada tópico, decida a alternativa correta (A, B ou C). Somente uma das opções é correta.

1. Compilação

- (a) A classe não irá compilar (existe um erro de compilação).
- (b) A classe irá compilar, mas terá um erro quando você tentar criar uma instância.
- (c) A classe irá compilar e rodar sem problemas.

2. Métodos de acesso e métodos modificadores

- (a) O método `getSeats` é um método de acesso.
- (b) O método `getSeats` é um método modificador.
- (c) O método `getSeats` funciona como método de acesso e como método modificador.

3. Atribuição

A instrução `model = model;` no construtor...

- (a) ...atribui o parâmetro `model` ao campo chamado `model`.
- (b) ...não modifica o campo `model`.
- (c) ...irá causar um erro de compilação.

4. Chamada de método

Qual das opções abaixo é a maneira correta de chamar o método `getSeats`. (Assuma que `car` representa um objeto de `Car`.)

- (a) `String s = car.getSeats();`
- (b) `int n = car.getSeats();`
- (c) `car.getSeats();`

5. Visibilidade

O fato que `numberOfSeats` é um campo público...

- (a) ...significa que a classe não irá compilar – isso não é permitido.
- (b) ...é considerado uma má prática (estilo).
- (c) ...faz com que o campo seja inicializado automaticamente.

6. Visibilidade

O fato que o método `getModel` é privado...

- (a) ...é considerado uma má prática (estilo).
- (b) ...aumenta a coesão.
- (c) ...significa que o método só pode ser chamado de dentro da classe `Car`.

7. Inicialização de campos

- (a) O método `getSeats` irá retornar 0 (zero).
- (b) O método `getSeats` irá disparar uma exceção, por que o campo `numberOfSeats` não é inicializado.
- (c) O método `getSeats` irá retornar um número aleatório.

8. Lógica

Depois de chamar o método `setEngineCode` ...

- (a) ...o valor de `engineCode` não poderá ser 120.
- (b) ...o valor de `engineCode` sempre será positivo.
- (c) ...o valor de `engineCode` sempre será um número par.

9. Variáveis

No método `setEngineCode`, a variável `n` is...

- (a) ...um parâmetro.
- (b) ...uma variável local.
- (c) ...um campo.

10. Tipos

Quando você chama o método `setEngineCode` com um valor de 3.5:

```
car.setEngineCode(3.5);
```

a seguinte situação acontecerá:

- (a) A instrução não irá compilar, uma vez que o tipo do parâmetro não bate com o tipo esperado.
- (b) O campo `engineCode` terá o valor 7.
- (c) O campo `engineCode` terá o valor 3, por que 3.5 será truncado para 3.

16. Escreva um método que receba tres números inteiros que representem uma data (dia, mes e ano) e escreva a data por extenso (menos o ano).

Dica: Você pode utilizar a classe `StringBuffer` para “montar” a data.

17. Escreva um programa que leia 10 nomes, armazenando-os em um array e mostre-os em ordem alfabética.

18. O que faz cada programa a seguir?

(a)

```
public class Main {  
    public static void main(String[] args) {  
        java.util.Scanner sc = new java.util.Scanner(System.in);  
        System.out.println("Digite um número");  
        int numero = sc.nextInt();  
        int base=2,i=1;  
        while (i++<numero)  
            base<=<=1;  
        System.out.println(base);  
    }  
}
```

```
(b)
public class Main {
    public static void main(String[] args) {
        java.util.Scanner sc = new java.util.Scanner(System.in);
        System.out.println("Digite um número");
        int numero = sc.nextInt();
        int base=1;
        base<=numero;
        System.out.println(base);
    }
}
```

```
(c)
public class Main {
    public static void main(String[] args) {
        java.util.Scanner sc = new java.util.Scanner(System.in);
        System.out.println("Digite um número");
        int numero = sc.nextInt();
        for (int i = 31 ; i >= 0 ; i--)
            System.out.print(((1<<i)& numero) >> i);
        System.out.println();
    }
}
```

19. Compile e execute o programa a seguir.

```
public class Main {
    public static void main(String[] args) {
        char a,b,c;
        a = '\u2208';
        b = '\u2f55';
        c = '\u2fd3';
        System.out.println("x " + a + " X");
        System.out.println(b);
        System.out.println(c);
    }
}
```

A linguagem JAVA suporta caracteres UNICODE, leiam sobre o que significa isso em <http://unicode.org/unicode/standard/translations/portuguese.html>

20. Escreva um programa em Java que leia um número inteiro e mostre a quantidade bits do número com valor 1.

21. Escreva um programa em JAVA que leia um número inteiro em 4 partes de 8 bits e mostre o caractere equivalente a cada parte.

22. Crie uma classe para representar uma pessoa, com os atributos privados de nome, data de nascimento e altura. Crie os métodos públicos necessários para sets e gets e também um método para imprimir todos dados de uma pessoa. Crie um método para calcular a idade da pessoa.

Crie uma classe **Agenda** que pode armazenar 10 pessoas e seja capaz de realizar as seguintes operações:

```

void armazenaPessoa(String nome, int idade, float altura);
void removePessoa(String nome);
int buscaPessoa(String nome); // informa em que posição da agenda está a
pessoa
void imprimeAgenda(); // imprime os dados de todas as pessoas da agenda
void imprimePessoa(int index); // imprime os dados da pessoa que está na
posição "i" da agenda.

```

23. Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio (desconsiderando o térreo), capacidade do elevador e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

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)
Entra	para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
Sai	para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
Sobe	para subir um andar (não deve subir se já estiver no último andar);
Desce	para descer um andar (não deve descer se já estiver no térreo);

Lembre-se de encapsular todos os atributos da classe (criar os métodos set e get).

24. Crie uma classe **Televisao** e uma classe **ControleRemoto** que pode controlar o volume e trocar os canais da televisão. O controle remoto permite:

- aumentar ou diminuir a potência do volume de som em uma unidade de cada vez;
- aumentar e diminuir o número do canal em uma unidade
- trocar para um canal indicado;
- consultar o valor do volume de som e o canal selecionado.

25. Crie uma classe Calculadora que possua a seguinte especificação:

- Atributos inteiros privados x e y
- Métodos de acesso para x e y
- Método somar
 - Sem parâmetro de entrada
 - Deve retornar a soma de x e y
- Método subtrair
 - Sem parâmetro de entrada
 - Deve retornar a subtração de x e y
- Método multiplicar
 - Sem parâmetro de entrada
 - Deve retornar a multiplicação de x e y
- Método dividir
 - Sem parâmetro de entrada
 - Deve retornar a divisão de x e y

Crie uma classe CalculadoraTester que utilize a nova classe Calculadora.

26. Crie uma classe Conta que irá manipular contas de um banco. Siga a especificação abaixo:

Atributos

numero	Número da conta
nome	Nome do titular da conta
tipo	Tipo da conta (poupança ou conta corrente)
saldo	O saldo da conta

Construtores

Construtor	Parâmetros	Descrição
1	Nenhum	Inicializa o saldo com zero
2	Número da conta Nome do titular	Inicializa o saldo com zero e atribui o numero da conta e o nome do titular a conta

Métodos

Implemente os métodos de acesso para os atributos

Além dos métodos de acesso, implemente os seguintes métodos abaixo:

Método	Parâmetros	Descrição
Depositar	Valor a ser depositado	Adiciona no saldo o valor passado
Sacar	Valor a ser sacado	Retira do saldo o valor passado

Crie uma classe chamada **Banco** que vai manipular contas. Esta classe deve realizar as seguintes tarefas:

- Criar uma conta simples de número 123 para Tio Patinhas
- Apresentar o seguinte menu para o dono da conta Tio Patinhas:
 - Realizar depósito
 - Realizar saque
 - Mostrar saldo
 - Mostrar informações do titular

Exemplo: caso o usuário escolha a opção de realizar depósito, o programa deve perguntar o valor que ele quer depositar.

27. Elabore uma classe ContaBancaria, com os seguintes membros:

- atributo String cliente
- atributo int num_conta
- atributo float saldo
- método sacar (o saldo não pode ficar negativo)
- método depositar

Agora acrescente ao projeto duas classes herdadas de ContaBancaria: ContaPoupanca e ContaEspecial, com as seguintes características a mais:

- Classe ContaPoupança
 - atributo int dia de rendimento
 - método calcularNovoSaldo, recebe a taxa de rendimento da poupança e atualiza o saldo.
- Classe ContaEspecial
 - atributo float limite
- método sacar (permitindo saldo negativo até o valor do limite)

Escreva uma classe de teste, e construa pelo menos um caso de teste positivo para cada método criado (tente pensar também em alguns casos de teste negativos).

28. Faça as classes para um programa que monta uma agenda de contatos (AgendaDeContatos) que possibilite o cadastro de 3 tipos de contatos com as seguintes características:

- a) Amigo: com nome, celular, email, idade
- b) Empresa: com nome, celular, telefone fixo, email, endereço
- c) Professor: nome, email, celular, site

Seu programa deve permitir a inserção, listagem, e remoção de contatos. Deve permitir ainda a busca de contatos por nome e a listagem de um tipo de específico de contato.

Dica: utilize o código gerado para o exercício 1 como ponto de partida.

29. Crie uma classe chamada Ingresso que possui um valor em reais e um método imprimeValor().

- a. crie uma classe VIP, que herda Ingresso e possui um valor adicional. Crie um método que retorne o valor do ingresso VIP (com o adicional incluído).
- b. crie uma classe Normal, que herda Ingresso e possui um método que imprime: "Ingresso Normal".
- c. crie uma classe CamaroteInferior (que possui a localização do ingresso e métodos para acessar e imprimir esta localização) e uma classe CamaroteSuperior, que é mais cara (possui valor adicional). Esta última possui um método para retornar o valor do ingresso. Ambas as classes herdam a classe VIP.
- d. Crie uma classe de Teste com o método main. Neste método: crie um ingresso. Peça para o usuário digitar 1 para normal e 2 para VIP. Conforme a escolha do usuário, diga se o ingresso é do tipo normal ou VIP. Se for VIP, peça para ele digitar 1 para camarote superior e 2 para camarote inferior. Conforme a escolha do usuário, diga se que o VIP é camarote superior ou inferior. Imprima o valor do ingresso.

30. Um animal contém um **nome**, **comprimento**, número de **patas** (o padrão é 4), uma **cor**, **ambiente** e uma **velocidade** (em m/s).

Um peixe é um animal, tem 0 patas, o seu ambiente é o mar (padrão), além disso, o peixe tem como **característica**: barbatanas e cauda;

Um mamífero é um animal, o seu ambiente é a terra (padrão);

Um urso é um mamífero, cor castanho e o seu **alimento** preferido é o mel.

Codifique as classes animal, peixe e mamífero.

Para a classe Animal, codifique os métodos:

Animal(String nome, String cor, String ambiente, int comprimento, float velocidade, int patas);

- void alteraNome(String nome);
- void alteraComprimento(int comprimento);
- void alteraPatas(int patas);
- void alteraCor(String cor);
- void alteraAmbiente(String ambiente);
- void alteraVelocidade(float velocidade);
- void dados(); // imprime os dados do animal

E os seus atributos são:

- String nome;

- int comprimento;
- int patas;
- String cor;
- String ambiente;
- float velocidade;

Para a classe Peixe, codifique:

- Peixe(String nome, String caracteristica, int comprimento, float velocidade);
- void alteraCaracteristica(String caracteristica);
- String caracteristica(); // retorna a características de um determinado peixe;
- void dados(); // imprimir na tela todos os dados

Para a classe Mamifero, codifique:

- Mamifero (String nome, String cor, String alimento,int comprimento, float velocidade, int patas);
- void alteraAlimento(String alimento);
- String alimento(); // retorna o alimento de um determindado uso
- void dados(); // imprimir na tela todos os dados

Por último, crie um arquivo de teste (por exemplo TesteAnimais.java) de forma a ter um jardim zoológico com os seguintes animais: camelo, tubarão, urso-do-canadá.

Exemplo de execução:

Zoo:

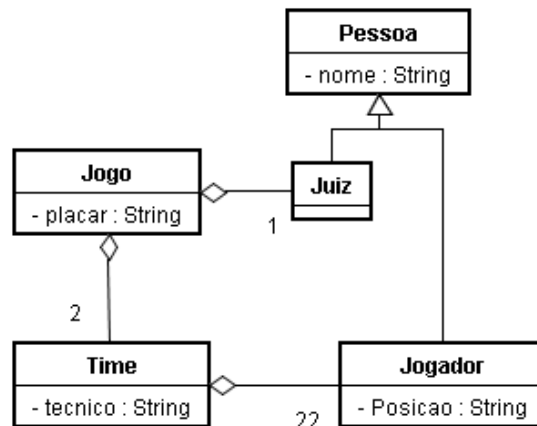
```

-----
Animal: Camelo
Comprimento: 150 cm
Patas: 4
Cor: Amarelo
Ambiente: Terra
Velocidade: 2.0 m/s
-----
Animal: Tubarão
Comprimento: 300 cm
Patas: 0
Cor: Cinzento
Ambiente: Mar
Velocidade: 1.5 m/s
Caracteristica: Barbatanas e cauda.
-----
Animal: Urso-do-canadá
Comprimento: 180 cm
Patas: 4
Cor: Vermelho
Ambiente: Terra
Velocidade: 0.5 m/s
Alimento: Mel
-----

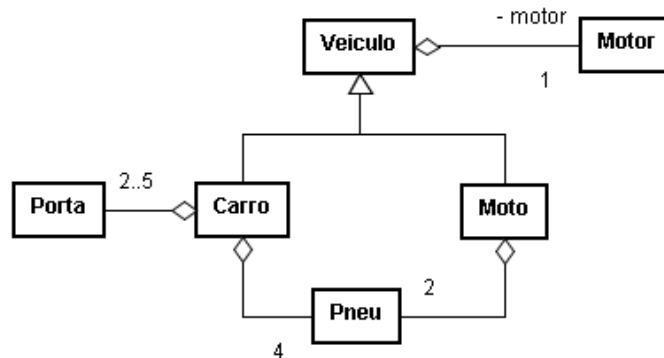
```

31. Implemente uma aplicação que corresponda aos seguintes diagramas de classes.

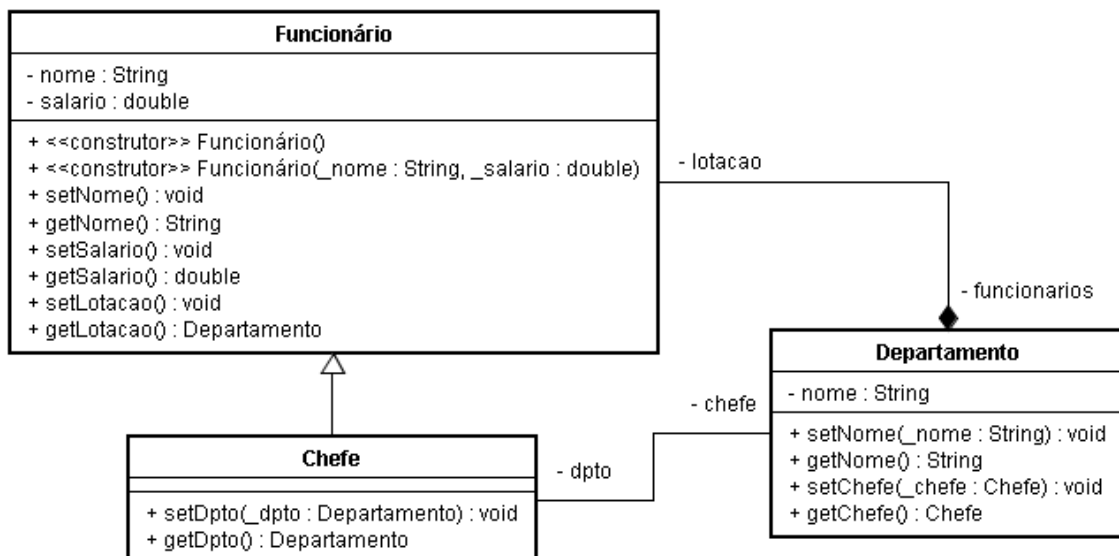
a)



b)



c)



32. A Locavel é uma locadora de veículos que está dando início a suas atividades no mercado local. Ela trabalha com aluguel de carros e motos. A empresa está necessitando de um software para o gerenciamento dos seus veículos. O dono da Locavel lhe procurou para construir esse software. Ele deseja que esse sistema a princípio seja capaz de:

- Cadastrar as motos e os carros para aluguel.
 - o No caso dos carros ele deseja cadastrar a marca, o modelo, a potência do motor e a quantidade de portas.
 - o Para as motos ele deseja cadastrar a marca, o modelo, tipo da partida (manual ou elétrica) e as cilindradas.
- Exibir quantos veículos estão cadastrados, listando quantas motos e quantos carros.
- Alugar os veículos para os clientes.
 - o Do cliente é solicitado o nome e o número da carteira de motorista, mas para o caso de motos com menos de 50 cilindradas não é solicitada a carteira de motorista.
 - o No caso do carro deve ser informado o modelo e a potência do motor.
 - o Para as motos é preciso informar o modelo e as cilindradas.
- Imprimir o recibo do aluguel informando:
 - o O modelo do veículo
 - o O custo, para motos calculados com base nas cilindradas (R\$ 2,00 para cada cilindrada) e carros calculados com base na potência (R\$ 100 multiplicado pela potência, sendo a potência um valor entre 1,0 e 3,0)
 - o A data da devolução (3 dias para motos 2 dias para carros)

Faça o sistema proposto usando os conceitos de classe, objeto, método, atributo, visibilidade (pública e privada), herança e polimorfismo.

33. Considere um polinômio de grau n:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0 x^0$$

Escreva uma classe **Termo** que represente um termo deste polinômio com os seguintes métodos:

<i>construtor</i>	Recebe dois parâmetros : a_i e i , e cria um objeto em memória na forma $a_i x^i$.
<i>insere</i>	Recebe um objeto da classe Termo e substitui os valores $a_i x^i$ do termo corrente por aqueles do termo recebido como parâmetro.
<i>calcula</i>	Recebe um valor de x como parâmetro e retorna o valor do termo calculado.

Escreva uma classe **Polinomio** que representa polinômio completo na forma de uma sequência de objetos da classe **Termo**, com os seguintes métodos:

<i>construtor</i>	Recebe um objeto da classe Termo e cria um polinômio em memória na forma: $P(x) = a_i x^i$.
<i>insere</i>	Recebe um objeto da classe Termo e adiciona o termo $a_i x^i$ ao polinômio recebido como parâmetro. O polinômio pode ter um termo $a_q x^q$ cujo valor de q seja igual a i , neste caso o método deve unificar ambos em um único termo.
<i>calcula</i>	Recebe como parâmetro outro objeto da classe Polinomio e realiza a fusão

	do polinômio recebido como parâmetro com o polinômio corrente.
--	--

Acrescente os métodos que achar necessários nas classes solicitadas.

34. Escreva uma classe **PolinomioComSub** que represente um polinômio com sub-polinômios. Considere o seguinte exemplo:

$$P(x) = ((a_5 x^5 + a_3 x^3) + b_0 x^0) + c_1 x^1 + (d_3 x^3 + d_1 x^1)$$

Neste exemplo, o polinômio $P(x)$ contém 3 sub-polinômios. Os parênteses aparecem apenas para identificar sub-polinômios dentro de polinômios. O primeiro sub-polinômio, por sua vez, contém dois sub-polinômios.

A classe **PolinomioComSub** deve ser capaz manter uma lista de termos (objetos da classe **Termo**) ou sub-polinômios (objetos da classe **Polinomio** ou **PolinomioComSub**). Ela possui os seguintes métodos:

insere	Recebe um objeto da classe Termo e adiciona o termo $a_i x^i$ ao polinômio recebido como parâmetro. O polinômio pode ter um termo $a_q x^q$ cujo valor de q seja igual a i , neste caso o método deve unificar ambos em um único termo.
insere	Recebe um objeto da classe Polinomio ou PolinomioComSub e o adiciona como subpolinômio.
calcula	Recebe um valor de x como parâmetro e retorna o valor de $P(x)$.
fusao	Não recebe parâmetros. Realiza uma fusão de todos os sub-polinômios do objeto corrente, de tal modo que este objeto corrente passe a ter um polinômio sem sub-polinômios, sintetizando os termos de todos os sub-polinômios que sofreram fusão.
fusao	Recebe como parâmetro um objeto da classe Polinomio ou PolinomioComSub e se comporta como o método função anterior, acrescentando o objeto recebido como parâmetro no processo.

Acrescente os métodos que achar necessários nas classes solicitadas. Em alguns casos pode ser interessante definir novas classes. Analise a possibilidade de adaptar a classe **PolinomioComSub** para que se torne herdeira de **Polinomio** e avalie as vantagens e desvantagens.

ATENÇÃO: Decisões de re-projeto fazem parte desta questão.