

**Instrução Criar o repositório no seu git com nome ListaExercíciosSegundoBimestre;
Para cada exercício criar um projeto novo. Deverá ser enviado no dia 26/04/2023 até as 23:59 hrs.**

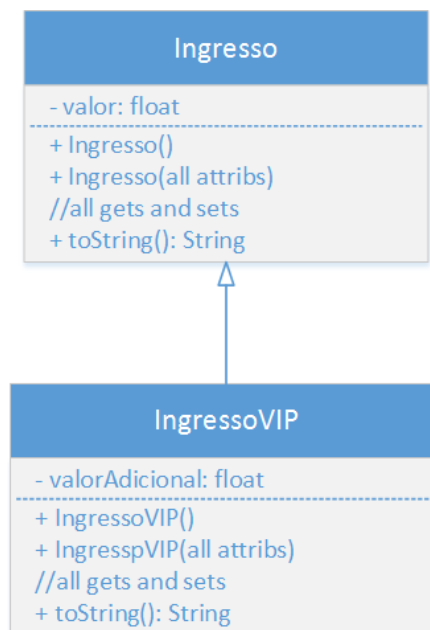
Por favor Envie o link do projeto no git de cada exercício de acordo com o Enunciado do exercício e com o RA e Nome completo no e-mail;

A Resposta de Cada Exercício deverá ser o Link do Projeto no Git dentro do Repositório Primeira Lista de Exercícios segundo bimestre;

1. Exercício sobre Ingressos

Crie uma classe chamada Ingresso que possua um atributo valor e um método toString que retorne a informação do valor do ingresso.

- Crie uma classe IngressoVIP, que herda de Ingresso e possui um atributo valor Adicional. O método toString da classe IngressoVIP deve considerar que o valor do ingresso é o valor da superclasse somado ao valor Adicional do IngressoVIP.
- Crie uma classe para testar os objetos das classes Ingresso e IngressoVIP.



2. Exercício sobre Empregados

Crie classes de forma a representar o diagrama a abaixo:



- a) A classe Empregado deve possuir dois atributos, nome e salario. Salario deve ser do tipo protected. Crie os métodos get e set para classes e o método toString.
- b) As classes Gerente deve herdar da classe Empregado. Crie os métodos get e set para a classe e o método toString. O método toString da classe Gerente deve incluir a informação do departamento, além dos dados da superclasse. O construtor da classe deve receber por parâmetro, além as informações da superclasse, a informação do departamento.
- c) A classe Vendedor deve herdar também da classe Empregado. Deve possuir ainda um método denominado calcularSalario. Esse método deve retornar um valor do tipo float, correspondente ao valor do salário acrescido do respectivo percentual de comissão. O construtor da classe deve receber por parâmetro, além as informações da superclasse, a informação do percentual de comissão do vendedor. O método toString da classe deve apresentar as informações de nome do empregado, salário sem comissão, salario com comissão e percentual de comissão.
- d) Crie uma classe para testar objetos das classes implementadas.

3. Exercício sobre Conta bancária

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: ContaPoupança 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
- ✓ redefinição do método sacar, permitindo saldo negativo até o valor do limite.

Após a implementação das classes acima, você deverá implementar uma classe Contas.Java, contendo o método main. Nesta classe, você deverá implementar:

- a) Incluir dados relativos a(s) conta(s) de um cliente;
- b) Sacar um determinado valor da(s) sua(s) conta(s);
- c) Depositar um determinado valor na(s) sua(s) conta(s);
- d) Mostrar o novo saldo do cliente, a partir da taxa de rendimento, daqueles que possuem conta poupança;
- e) Mostrar os dados da(s) conta(s) de um cliente;

4. Exercício sobre Animais

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), cor cinzenta (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);**

E os seus atributos são:

- ✓ **String nome();**
- ✓ **int comprimento();**
- ✓ **int patas();**
- ✓ **String cor();**
- ✓ **String ambiente();**
- ✓ **float velocidade();**
- ✓ **void dados(); // imprime os dados do animal**

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

Característica: Barbatanas e cauda.

Animal: Urso-
do-canadá
Comprimento:
180 cm
Patas: 4

Cor:
Vermelho
Ambiente:
Terra
Velocidade:
0.5 m/s
Alimento:
Mel

Faça outro arquivo (**TesteAnimais2.Java**) para que possam ser incluídos quantos animais o usuário desejar, no máximo 10 animais de cada espécie. (**DICA:** Utilize vetores de objetos);