



Painel > Meus cursos > TP1_2022/1 > Tópico 7 - Herança e Polimorfismo > Atividade 7.2 - Lista de exercícios práticos - Java

Iniciado em	domingo, 7 ago 2022, 14:16
Estado	Finalizada
Concluída em	sábado, 13 ago 2022, 15:30
Tempo empregado	6 dias 1 hora
Avallar	10,00 de um máximo de 10,00(100%)

Questão 1

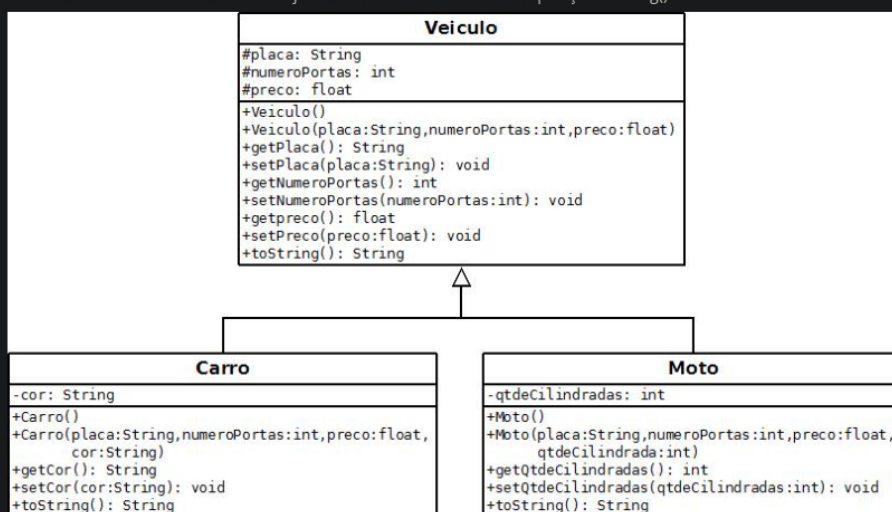
Correto

Atingiu 5,00 de 5,00

Marcar questão

Veículos

Crie um programa de acordo com o diagrama que classes abaixo, que contenha a superclasse Veiculo, nela deve conter os atributos protegidos Placa (*String*), Quantidade de Portas (*int*), Preço (*float*). Faça os métodos gets e sets para cada atributo, e dois construtores para a inicialização dos atributos da classe, um para a inicialização de todos os atributos com valores default e outro para a inicialização recebendo todos os argumentos de acordo com os atributos. Faça a subclasse carro, com o atributo privado cor (*String*) e os métodos get, set e construtores, como descritos para a superclasse. Crie a subclasse moto, com o atributo privado Quantidade de Cilindradas (*int*) e os métodos get, set e construtores, como descritos para a superclasse. Faça a classe principal para utilizar essas classes considerando o conceito de herança. Crie três objetos. O primeiro objeto é do tipo Veiculo, use o construtor para inicializar os atributos com valores default. Depois solicite ao usuário os dados de entrada, de acordo com todos os atributos da superclasse veiculo e altere os valores dos atributos usando os setters. O segundo objeto é do tipo Carro, solicite todos os dados ao usuário, de acordo com os atributos da subclasse e inicialize os atributos usando os dados como argumentos para o construtor. O terceiro objeto é do tipo Moto, solicite todos os dados ao usuário, de acordo com os atributos da subclasse e inicialize os atributos usando os dados como argumentos para o construtor. Depois apresente os valores dos atributos considerando os três objetos usando o método de sobreposição toString().



Entrada:

A primeira linha corresponde aos dados de acordo com os atributos da classe Veiculo (placa, número de portas, preço), na mesma linha e separados por espaço. A segunda linha corresponde aos dados de acordo com os atributos da subclasse Carro e sua superclasse (placa, número de portas, preço, cor), na mesma linha e separados por espaço. A terceira linha corresponde aos dados de acordo com os atributos da subclasse moto e sua superclasse (placa, número de portas, preço, quantidade de cilindradas), na mesma linha e separados por espaço.

Saída:

Os dados do objeto do tipo Veiculo são apresentados na primeira linha, os dados do objeto do tipo Carro na segunda linha e os dados do objeto do tipo Moto na terceira linha. Para cada objeto, os valores são separados por um espaço, e são apresentados de acordo com a sequência e texto usados no exemplo.

Objetivos de estudo:

- Herança e polimorfismo: o uso das palavras-chave extends e super, o modificador protected, sobreposição e sobrecarga;
- Considere testes fora do coderunner para analisar outros exemplos.

For example:

Input	Result
BRA1E10 2 2500	Placa: BRA1E10 Número de portas: 2 Preço: R\$2500.0
ABC1D34 4 50000 Branco	Placa: ABC1D34 Número de portas: 4 Preço: R\$50000.0 Cor: Branco
EF62H23 0 20000 150	Placa: EF62H23 Número de portas: 0 Preço: R\$20000.0 Quantidade de cilindradas: 150

Answer: (penalty regime: 0, 0, 0, 1, 2, 3 %)

```
1 //
2 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3 * Click nbfs://nbhost/SystemFileSystem/Templates/Project/Maven2/JavaApp/src/main/java/${packagePath}/
```

Navegação do questionário



Mostrar uma página por vez

Terminar revisão

```

4 //
5
6 import java.util.Scanner;
7
8 class Veiculo{
9     protected String placa;
10    protected int numeroPortas;
11    protected float preco;
12
13    public Veiculo(){
14

```

	Input	Expected
✓	BRA1E10 2 2500 ABC1D34 4 50000 Branco EFG2H23 0 20000 150	Placa: BRA1E10 Número de portas: 2 Preço: R\$2500.0 Placa: ABC1D34 Número de portas: 4 Preço: R\$50000.0 Cor: Branco Placa: EFG2H23 Número de portas: 0 Preço: R\$20000.0 Quantidade de cilindrada

Passou em todos os testes! ✓

Correto

Notas para este envio: 5,00/5,00.

Questão 2

Correto

Atingiu 5,00 de 5,00

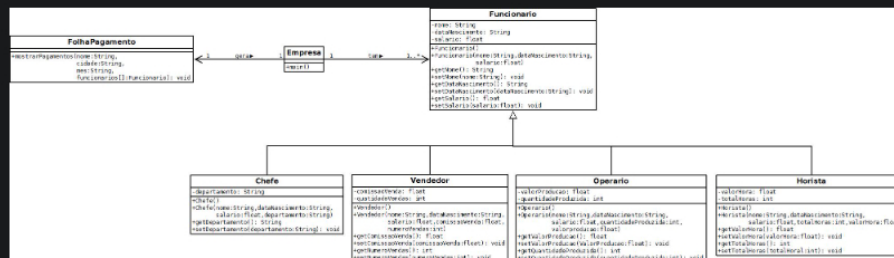
Marcar questão

Empresa

Uma empresa tem quatro tipos de funcionários: chefes, vendedores, operários e horistas. Cada um desses funcionários ganha seus salários de acordo com as seguintes regras:

- Chefe: salário fixo;
- Vendedor: salário fixo + comissão x quantidade de vendas;
- Operário: valor por produção x quantidade produzida;
- Horista: valor por hora x total de horas trabalhadas.

Faça um programa de acordo com o diagrama de classes abaixo. A superclasse Funcionario deve incluir os campos privados nome, dataNascimento e salario. As subclasses devem conter os atributos necessários de acordo com o tipo de funcionário. Considere o construtor de cada subclasse para calcular e alterar o salário de acordo as regras mencionadas acima. Além disso, crie uma classe Empresa (classe principal), que conterá o método main(). A empresa tem vários funcionários, logo crie um vetor (array) de quatro funcionários. Adicione a ele ao menos um tipo de cada funcionário definido anteriormente. Criar também uma classe FolhaPagamento, que mostra a folha de pagamento da empresa de acordo com o vetor de funcionários, considerando informações específicas de acordo com o tipo de Funcionario (subclasse). A folha de pagamento contém o nome e cidade da empresa, o mês do pagamento e os dados de cada funcionário. No programa principal, solicite ao usuário o nome e cidade da empresa (String), o mês do pagamento (String). Crie um vetor com quatro objetos do tipo Funcionario, cada um instanciado a partir de uma subclasse de Funcionario. Para isso, solicite ao usuário os dados de cada funcionário. Em seguida, crie e instancie um objeto da classe FolhaPagamento, e invoque o método mostrarPagamentos, passando como argumento os dados informados pelo usuário: nome, cidade e mês (String), e também o vetor de funcionários.



Entrada:

A primeira linha corresponde aos dados da empresa e da folha de pagamento, nome da empresa, cidade da empresa e mês do pagamento. As próximas linhas são os dados para inicializar os objetos das subclasses de Funcionario, com a seguinte ordem: Chefe, Vendedor, Operario e Horista, um objeto por linha. Para cada linha os dados são separados por espaço, seguindo a ordem de atributos apresentados no diagrama de classes.

Saida:

Os dados da empresa (nome e cidade) e o mês do pagamento. Depois de uma linha em branco, apresentar os dados de cada objeto do tipo funcionário, separados por uma linha em branco. Para cada objeto, os valores são apresentados um por linha, de acordo com a sequência e texto usados no exemplo.

Objetivos de estudo:

- Herança e polimorfismo: o uso da palavra-chave Instanceof;
- Considere testes fora do coderunner para analisar outros exemplos.

For example:

Input	Result
Comp Brasilia Outubro Pedro 10/10/1960 20000 CEO Paulo 20/04/2000 3000 100 5 Joaquim 01/11/1994 0 0.5 1000 Ana 19/12/1963 0 25 224	Nome da Empresa: Comp Endereço: Brasilia Mês: Outubro Nome: Pedro Data de Nascimento: 10/10/1960 Salário: R\$20000.0 Departamento: CEO Cargo: chefe Nome: Paulo Data de Nascimento: 20/04/2000 Salário: R\$3500.0 Cargo: vendedor Quantidade Vendas: 100 Comissao: 5.00

```
Comissão por Venda: R$100.0
Número de vendas: 5

Nome: Joaquim
Data de Nascimento: 01/11/1994
Salário: R$500.0
Cargo: Operário
Valor por Produção: R$0.5
Quantidade produzida: 1000

Nome: Ana
Data de Nascimento: 19/12/1963
Salário: R$5600.0
Cargo: horista
Valor por Hora: R$25.0
Total de horas: 224
```

Answer: (penalty regime: 0, 0, 0, 1, 2, 3 %)

```
1 import java.util.Scanner;
2
3 /**
4  *
5  * @author leo_r
6  */
7 class FolhaPagamento{
8     public void mostrarPagamentos(String nome, String cidade, String mes, Funcionario[] funcionarios){
9
10        System.out.println("Nome da Empresa: " + nome + "\n" + "Endereço: " + cidade + "\n" + "Mês: "
11        for(Funcionario funcionario: funcionarios){
12            System.out.println("\n" + "Nome: " + funcionario.getNome());
13            System.out.println("Data de Nascimento: " + funcionario.getDataNascimento());
14            if(funcionario instanceof Chefe){
```

Input	Expected	Got
✓ Comp Brasilia Outubro Pedro 10/10/1960 20000 CEO Paulo 20/04/2000 3000 100 5 Joaquim 01/11/1994 0 0.5 1000 Ana 19/12/1963 0 25 224	Nome da Empresa: Comp Endereço: Brasilia Mês: Outubro Nome: Pedro Data de Nascimento: 10/10/1960 Salário: R\$20000.0 Departamento: CEO Cargo: chefe Nome: Paulo Data de Nascimento: 20/04/2000 Salário: R\$3500.0 Cargo: vendedor Comissão por Venda: R\$100.0 Número de vendas: 5 Nome: Joaquim Data de Nascimento: 01/11/1994 Salário: R\$500.0 Cargo: Operário Valor por Produção: R\$0.5 Quantidade produzida: 1000 Nome: Ana Data de Nascimento: 19/12/1963 Salário: R\$5600.0 Cargo: horista Valor por Hora: R\$25.0 Total de horas: 224	✓ Nome da Empresa: Comp Endereço: Brasilia Mês: Outubro Nome: Pedro Data de Nascimento: 10/10/1960 Salário: R\$20000.0 Departamento: CEO Cargo: chefe Nome: Paulo Data de Nascimento: 20/04/2000 Salário: R\$3500.0 Cargo: vendedor Comissão por Venda: R\$100.0 Número de vendas: 5 Nome: Joaquim Data de Nascimento: 01/11/1994 Salário: R\$500.0 Cargo: Operário Valor por Produção: R\$0.5 Quantidade produzida: 1000 Nome: Ana Data de Nascimento: 19/12/1963 Salário: R\$5600.0 Cargo: horista Valor por Hora: R\$25.0 Total de horas: 224

Passou em todos os testes! ✓

Correto

Notas para este envio: 5,00/5,00.

Terminar revisão

Seguir para... ▾