



UPskill - JAVA

Linguagem de Programação Java

EXERCÍCIO VENCIMENTOS

As classes *Trabalhador*, *TrabalhadorPeca*, *TrabalhadorComissao* e *TrabalhadorHora* foram construídas com base na necessidade de desenvolver um programa que calculasse os vencimentos mensais dos seguintes tipos de trabalhadores:

- Trabalhador à peça (*TrabalhadorPeca*): vencimento dependente do número de peças produzidas. Foram considerados dois tipos de peças e com pagamentos diferentes, sendo atualmente estes pagamentos iguais a 6,50€ e 9,50€;
- Trabalhador à comissão (*TrabalhadorComissao*): vencimento igual a um salário base mais uma percentagem das vendas. Foi considerado, por omissão, o salário base de 580,00€;
- Trabalhador à hora (*TrabalhadorHora*), vencimento proporcional às horas de trabalho. Foi considerado, por omissão, o pagamento à hora de 10,00€.

Para tal, definiu-se uma classe para cada tipo de trabalhador com uma implementação apropriada do método *calcularVencimento()* e uma superclasse com o nome *Trabalhador*.

Analise o código destas classes e depois faça o seguinte:

1. Esboce um diagrama de classes, em notação UML, ilustrando todas as classes fornecidas.
2. Crie uma nova classe, chamada *MainVencimentos*, para invocar as funcionalidades implementadas.
3. Nesta classe, crie 3 objetos, um para cada tipo de trabalhador:
 - Objeto do tipo *TrabalhadorPeca*, de nome Jorge Silva que produziu 53 peças do tipo 1 e 62 do tipo 2;
 - Objeto do tipo *TrabalhadorComissao*, de nome Susana Ferreira, com o salário base de 650,00€ e uma comissão de 4,25% sobre as vendas efetuadas que totalizaram o valor de 2731,50€;
 - Objeto do tipo *TrabalhadorHora*, de nome Carlos Miguel, que ganha por hora 4,50€ e com um total de horas de trabalho igual a 168.
4. Crie um contentor de objetos para guardar os objetos existentes. Este contentor deve ser um *array* do tipo *Object* com comprimento 10.
5. Guarde os objetos existentes no contentor.
6. Programe as seguintes listagens independentes, obtidas através do varrimento do contentor com uma instrução *for* tradicional:
 - Listagem das representações textuais dos trabalhadores;
 - Listagem das representações textuais apenas dos trabalhadores à hora;
 - Listagem dos nomes dos trabalhadores existentes, acompanhadas dos respetivos vencimentos (com duas casas decimais).

Linguagem de Programação Java

Herança e Polimorfismo

7. Substitua o tipo *Object* do *array* pelo tipo *Trabalhador* (é o tipo mais genérico da hierarquia de classes).
8. Simplifique o código das listagens.