



UPskill - JAVA

Linguagem de Programação Java

UPskill 2020/2021

Linguagem de Programação Java

Agregação e composição

EXERCÍCIO EMPREGADO

Pretende-se uma classe de nome **Empregado** que satisfaça os seguintes requisitos:

- Possua os atributos: *primeiroNome*, *ultimoNome*, *dataContrato*, *horaEntrada* e *horaSaida*. O atributo *dataContrato* deve ser um objeto da classe *Data* fornecida. Os atributos *horaEntrada* e *horaSaida* devem ser objetos da classe *Tempo* também disponibilizada;
- As classes *Data* e *Tempo* devem ser reutilizadas por composição;
- Disponibilize funcionalidades para:
 - Calcular o número de horas de trabalho semanal de um empregado, considerando que todos os empregados trabalham 5 dias por semana;
 - Determinar há quanto tempo o empregado foi contratado.

Implemente uma solução usando o seguinte procedimento:

1. Elabore um diagrama de classes para visualizar as relações entre as classes *Empregado*, *Data* e *Tempo*.
2. Crie uma biblioteca de classes contendo as classes *Data* e *Tempo*, seguindo as instruções referidas no final deste documento.
3. Implemente a classe *Empregado*.
4. Construa outra classe com o nome *MainEmpregado* que permita invocar as funcionalidades da classe *Empregado*. Para isso:
 - a) Crie e visualize uma instância de *Data* para representar a data atual;
 - b) Crie e visualize duas instâncias de *Tempo*;
 - c) Crie duas instâncias de *Empregado* usando para o efeito as instâncias de *Data* e de *Tempo* criadas anteriormente;
 - d) Verifique se os atributos do tipo *Data* dos empregados criados possuem referências partilhadas;
 - e) Verifique se os atributos do tipo *Tempo* dos empregados criados possuem referências partilhadas;
 - f) Altere o conteúdo do objeto *Data* e o conteúdo dos objetos *Tempo* inicialmente criados;
 - g) Visualize as instâncias de *Data* e de *Tempo*, bem como as instâncias de *Empregado* criadas;
 - h) Altere a data de contrato e as horas de entrada e de saída do segundo empregado;
 - i) Armazene as instâncias de *Empregado* criadas num contentor de objetos do tipo *ArrayList*;
 - j) Liste todos os empregados;
 - k) Liste o nome, o número de horas de trabalho por semana e a antiguidade de cada um dos empregados.
5. Crie testes unitários para os métodos relevantes da classe *Empregado*.

DEPENDÊNCIAS MAVEN NO NETBEANS

A: Criar um projeto Maven para as classes Data e Tempo

1º Passo: Criar um novo projeto Maven (Utilitarios)

- Selecionar **File > New Project**
- Em *Choose Project* seleccionar:
Categories: Maven
Projects: Java Application
Prosseguir fazendo clique em *Next*.
- Em *Name and Location* definir:
Project Name: Utilitarios
Project Location: <seleccionar o local onde será guardada a pasta do projeto>

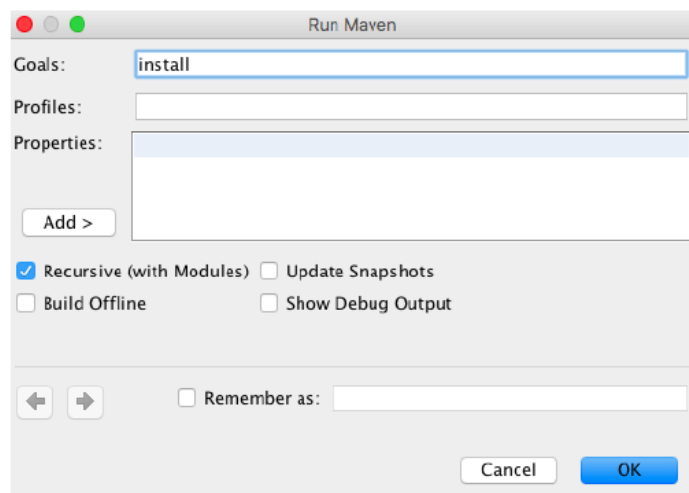
Concluir a criação do novo projeto fazendo clique em *Finish*.

2º Passo: Criar as classes Data e Tempo

- No package *Source Packages* criar as classes Data e Tempo.
- Copiar para as classes criadas o código das classes Data e Tempo disponibilizadas.

3º Passo: Disponibilizar as dependências no repositório local Maven

- Fazer clique direito no projeto Utilitarios->Run Maven->Goals
Aparece a seguinte janela, e fazer install em Goals:



Linguagem de Programação Java

Agregação e composição

B: Criar um projeto Maven para as classes do Exercício Empregado

4º Passo: Criar um novo projeto (Empregado)

- Selecionar **File > New Project**
- Em *Choose Project* selecionar:

*Categories: **Maven***

*Projects: **Java Application***

Prosseguir fazendo clique em *Next*.

- Em *Name and Location* definir:

Project Name: Empregado

Project Location: <selecionar o local onde será guardada a pasta do projeto>

Concluir a criação do novo projeto fazendo clique em *Finish*.

5º Passo: Utilização da dependência local noutro projeto (Empregado)

No ficheiro pom.xml colocar as dependências:

```
<dependencies>
  <dependency>
    <groupId>grupold-do-projeto</groupId>
    <artifactId>Utilitarios</artifactId>
    <version>1.0-SNAPSHOT</version>
  </dependency>
</dependencies>
```

O *grupold-do-projeto* é o mesmo do que já existe neste pom.xml (Empregado).

Nota: Sempre que houver alterações no projeto Utilitarios é necessário repetir o passo 3.