

# UPskill – Java+.NET

Programação Orientada a Objetos – 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

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

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

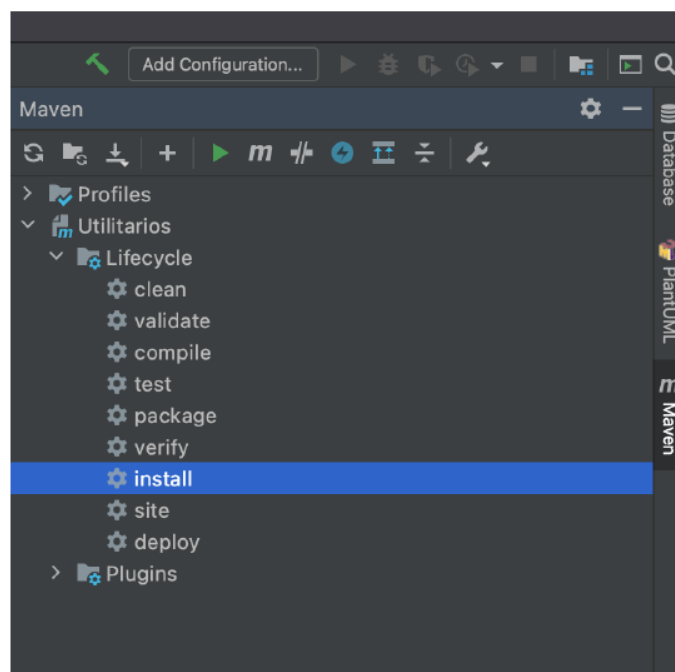
- Considere as seguintes coordenadas para o projeto Maven:
  - groupId: org.upskill
  - artifactId: Utilitarios

#### 2º Passo: Criar as classes Data e Tempo

- Crie o package org.upskill.utils e crie neste package as classes Data e Tempo
- Copie para as classes criadas o código das classes Data e Tempo disponibilizadas

#### 3º Passo: Disponibilizar as dependências no repositório local Maven

- Selecione o menu View/Tool Windows/Maven. De seguida, seleccione: Utilitarios -> Lifecycle -> install



#### 4º Passo: Criar um novo projeto (Empregado)

- Crie um projeto Maven com o nome Empregado

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

- No ficheiro pom.xml coloque as seguintes dependências:

```
<dependencies>
<dependency>
<groupId>org.upskill</groupId>
<artifactId>Utilitarios</artifactId>
<version>1.0-SNAPSHOT</version>
</dependency>
</dependencies>
```

#### **6º Passo: Utilização da dependência local noutro projeto (Empregado)**

- Importar as classes Data e Tempo no projeto Empregado:

```
import org.upskill.utils.Data;
import org.upskill.utils.Tempo;
```

**Nota 1:** Sempre que houver alterações no projeto Utilitarios é necessário repetir o passo 3.

**Nota 2:** Se o IDE não reconhecer as classes Data e Tempo (imports), remover os ficheiros com o nome “\_maven.repositories” ou “\_remote.repositories” (o que existir) da pasta onde se encontra o artefacto (\$home/.m2/repository/org/upskill/Utilitarios/1.0-SNAPSHOT); e por fim, reiniciar o IDE.