

# Dossier de tests

---

## Sommaire

- 1. Introduction
- 2. Description de la procédure de test
- 3. Description des informations à enregistrer pour le test
- 4. Campagne de test
  - 1. Tests
    - `anneeEstBissextile()`
    - `nbJoursDansMois()`
    - `Date()`
    - `compareTo()`
    - `dateDuLendemain()`
    - `dateDeLaVeille()`
  - 2. Résultats
- 5. Conclusion

## 1. Introduction

Ce dossier de test est destiné à tester les différentes classes du TP. Comme actuellement seuls les packages `vue`, `modele` et plus tard `controlleur` sont utilisés, il s'agira de tester les méthodes de la version des classes `Date` du package `modele`. Notez que l'énoncé nous demande de tester la méthode `dernierJourDuMois()` mais que le nom de cette fonction dans le projet actuel est `nbJoursDansMois()` ; cela n'affecte pas le comportement de la fonction et donc le test associé.

## 2. Description de la procédure de test

Les tests seront tous conçus dans ce dossier puis développés, exécutés et enfin versionnés un par un (ou en groupe lors de dépendances d'un test sur un autre) afin de régler au plus tôt les erreurs de conception des tests de manière récursive. L'exécution sera ensuite ordonnée par un schéma de dépendances et cette ordre sera appliqué dans la classe de test à l'aide des décorateurs intégrés à JUnit.

## 3. Description des informations à enregistrer pour le test

Le résultat des tests sont directement visibles dans l'onglet du volet latéral de Visual Studio Code fourni par l'extension Tests Explorer après exécution au même endroit.

## 4. Campagne de test

### Contexte

Configuration logicielle :	Visual Studio 1.77, Java Extension Pack 0.9, Test Explorer, JUnit intégré par Maven (Java Extension Pack), OpenJDK 11, git 2.34.1, Pop!_OS 22.04 LTS
Configuration matérielle :	AMD® Ryzen 5 4500u with radeon graphics × 6, 8 GiB RAM && 12th Gen Intel® Core™ i3-1215U × 8, 16 GiB RAM
Date de début : 31/03/2023	Date de finalisation : indéterminée

Contexte	
Tests à appliquer :	NA
Responsable de la campagne de test :	Breval Ferrari

1. Tests

anneeEstBissextile()

Information	
Identification : AB	Version : 0.1
Description :	Renvoie un boolean indiquant la bissextilité de l'année. Quatre années de chaque catégorie seront testées.
Resesources requises :	Classe Date du package modele situé dans /src/main/java/modele, Java JDK + JRE
Responsable :	Breval Ferrari

Classe	Année	Résultats Attendus
P1	Bissextile	V
P1	Classique	F

Classe	A	R
P0	2016	V
P1	2020	V
P2	2024	V
P3	2028	V
P4	2017	F
P5	2018	F
P6	2019	F
P7	2023	F

nbJoursDansMois()

Informations	
Identification : JM	Version : 0.1
Description :	Cette fonction retourne le nombre de jours dans le mois de l'instance de la classe Date.
Resesources requises :	Classe Date du package modele situé dans /src/main/java/modele, Java JDK + JRE
Responsable :	Breval Ferrari

Classe	Mois	Année	Résultats Attendus
--------	------	-------	--------------------

Classe		Mois	Année	Résultats Attendus
P1	Janvier/Mars/Mai/Juillet/Août/Octobre/Décembre		Classique	31
P2	Janvier/Mars/Mai/Juillet/Août/Octobre/Décembre		Bissextile	31
P3	Avril/Juin/Septembre/Novembre		Classique	30
P4	Avril/Juin/Septembre/Novembre		Bissextile	30
P5	Février		Classique	28
P6	Février		Bissextile	29

Classe	M	A	R
P0	01	2021	31
P1	03	2021	31
P2	05	2021	31
P3	07	2021	31
P4	08	2021	31
P5	10	2021	31
P6	12	2021	31
P7	01	2020	31
P8	03	2020	31
P9	05	2020	31
P10	07	2020	31
P11	08	2020	31
P12	10	2020	31
P13	12	2020	31
P14	04	2021	30
P15	06	2021	30
P16	09	2021	30
P17	11	2021	30
P18	04	2020	30
P19	06	2020	30
P20	09	2020	30
P21	11	2020	30
P22	02	2021	28
P23	02	2020	29

Date()

## Informations

Identification :  
DA

Version : 0.1

Description :

Constructeur de la classe Date sans paramètre qui retourne la date courante. Il conviendra pour ce test d'utiliser une autre méthode de récupération de date ; comme les tests sont exécutés sur une distribution Linux, la commande date de l'OS sera utilisée.

Resesources  
requises :

Classe Date du package modele situé dans /src/main/java/modele, Java JDK + JRE

Responsable :

Breval Ferrari

## Classe Résultats Attendus

P1

Date actuelle

compareTo()

## Informations

Identification : CT

Version : 0.1

Description :

Implémentation de la fonction compareTo incluse dans les librairies standard.

Resesources requises  
:

Classe Date du package modele situé dans /src/main/java/modele, Java JDK + JRE

Responsable :

Breval Ferrari

## Classe Jour Mois Année Résultats Attendus

P0

>

>

>

>

P1

<

>

>

>

P2

<

<

>

>

P3

<

<

<

<

P4

>

<

<

<

P5

>

>

<

<

P6

=

>

>

>

P7

=

<

>

>

P8

=

=

>

>

P9

=

=

<

<

P10

=

=

=

=

P11

<

=

=

<

P12

>

=

=

>

P13

=

<

=

<

Classe	Jour	Mois	Année	Résultats Attendus
P14	=	>	=	>
P15	>	=	>	>
P16	<	=	>	>
P17	<	=	<	<
P18	>	>	=	>
P19	<	>	=	>
P20	<	<	=	<

Partons d'une instance correspondant au 02/02/02

Classe	J	M	A	R
P0	1	3	3	01
P1	1	3	3	01
P2	1	1	3	01
P3	1	1	1	-1
P4	3	1	1	-1
P5	3	3	1	-1
P6	2	3	3	01
P7	2	1	3	01
P8	2	2	3	01
P9	2	2	1	-1
P10	2	2	2	00
P11	1	2	2	-1
P12	3	2	2	01
P13	2	1	2	-1
P14	2	3	2	01
P15	3	2	3	01
P16	1	2	3	01
P17	1	2	1	-1
P18	3	3	2	01
P19	1	3	2	01
P20	1	1	2	-1

dateDuLendemain()

**Informations**

### Informations

Identification : DL	Version : 0.1
Description :	Renvoie une instance de Date correspondant à la date du lendemain.
Resesources requises :	Classe Date du package modele situé dans /src/main/java/modele, Java JDK + JRE
Responsable :	Breval Ferrari

Classe	Date initiale	Résultats Attendus
--------	---------------	--------------------

P0	01/01/2001	02/01/2001
----	------------	------------

P1	31/01/2001	01/02/2001
----	------------	------------

P2	28/02/2001	01/03/2001
----	------------	------------

P3	28/02/2000	29/02/2000
----	------------	------------

P4	29/02/2000	01/03/2000
----	------------	------------

P5	31/12/2001	01/01/2002
----	------------	------------

dateDeLaVeille()

### Informations

Identification : DV	Version : 0.1
Description :	Renvoie une instance de Date correspondant à la date de la veille.
Resesources requises :	Classe Date du package modele situé dans /src/main/java/modele, Java JDK + JRE
Responsable :	Breval Ferrari

Classe	Date initiale	Résultats Attendus
--------	---------------	--------------------

P0	02/01/2001	01/01/2001
----	------------	------------

P1	01/02/2001	31/01/2001
----	------------	------------

P2	01/03/2001	28/02/2001
----	------------	------------

P3	29/02/2000	28/02/2000
----	------------	------------

P4	01/03/2000	29/02/2000
----	------------	------------

P5	01/01/2002	31/12/2001
----	------------	------------

## 2. Résultats

### Test 1

Référence du test appliqué :	AB
Responsable :	Breval Ferrari
Date d'application :	08/04/2023
Résultat :	OK

**Test 1**

Occurrences des résultats :      Systématique

**Test 2**

Référence du test appliqué :    JM

Responsable :                      Breval Ferrari

Date d'application :              08/04/2023

Résultat :                          OK

Occurrences des résultats :      Systématique

**Test 3**

Référence du test appliqué :    DA

Responsable :                      Breval Ferrari

Date d'application :              08/04/2023

Résultat :                          OK

Occurrences des résultats :      Systématique

**Test 4**

Référence du test appliqué :    CT

Responsable :                      Breval Ferrari

Date d'application :              08/04/2023

Résultat :                          OK

Occurrences des résultats :      Systématique

**Test 5**

Référence du test appliqué :    DL

Responsable :                      Breval Ferrari

Date d'application :              08/04/2023

Résultat :                          OK

Occurrences des résultats :      Systématique

**Test 6**

Référence du test appliqué :    DV

Responsable :                      Breval Ferrari

Date d'application :              08/04/2023

Résultat :                          OK

Occurrences des résultats :      Systématique

## 5. Conclusion

Certains tests ont permis de repérer des cas particuliers où le comportement des fonctions était anormal, ce qui a conduit à une réécriture du code. Comme les tests sont relativement indépendants, les tests n'ont que peu de hiérarchie et seuls les tests DL et DV ont été subordonnés à CT. Enfin, ce dossier de test m'a permis d'apprendre beaucoup sur la structure d'un projet sous Maven ainsi que l'organisation des tests sous JUnit.