

# Les Assertions en Java

Olivier Cailloux

LAMSADE, Université Paris-Dauphine

Version du 6 janvier 2017

# Principe

- Assertion : s'assurer d'une condition en cours de programme
- Exemple : vérifier que  $i$  est bien positif
- En Java : avec `assert`

## Assertion sans explication

```
assert i > 0;
```

## Assertion avec explication

```
assert i > 0 : "Integer must be strictly positive,  
received " + i + ".";
```

- Usage sans explication acceptable *si* vous êtes à peu près certain que l'assertion ne sera jamais levée
- Utilité dans ce cas ?

# Principe

- Assertion : s'assurer d'une condition en cours de programme
- Exemple : vérifier que  $i$  est bien positif
- En Java : avec `assert`

## Assertion sans explication

```
assert i > 0;
```

## Assertion avec explication

```
assert i > 0 : "Integer must be strictly positive,  
received " + i + ".";
```

- Usage sans explication acceptable *si* vous êtes à peu près certain que l'assertion ne sera jamais levée
- Utilité dans ce cas ? Échec rapide si jamais vous vous êtes trompé ; documentation du code

# Fonctionnement

- Si assertion non satisfaite : arrêt de l'exécution + affichage de l'appel fautif (et des appels parents)
- Assertions non exécutées par défaut
- Exécutées ssi JVM reçoit paramètre `-ea`
- Intérêt exécution conditionnelle ?

# Fonctionnement

- Si assertion non satisfaite : arrêt de l'exécution + affichage de l'appel fautif (et des appels parents)
- Assertions non exécutées par défaut
- Exécutées ssi JVM reçoit paramètre `-ea`
- Intérêt exécution conditionnelle ? Pas de ralentissement en production

# Fonctionnement

- Si assertion non satisfaite : arrêt de l'exécution + affichage de l'appel fautif (et des appels parents)
- Assertions non exécutées par défaut
- Exécutées ssi JVM reçoit paramètre `-ea`
- Intérêt exécution conditionnelle ? Pas de ralentissement en production
- Inconvénient ?

# Fonctionnement

- Si assertion non satisfaite : arrêt de l'exécution + affichage de l'appel fautif (et des appels parents)
- Assertions non exécutées par défaut
- Exécutées ssi JVM reçoit paramètre `-ea`
- Intérêt exécution conditionnelle ? Pas de ralentissement en production
- Inconvénient ?
  - Changement de comportement en production !

# Assert pour vérifier préconditions ?

assert : solution *imparfaite* pour tester les préconditions en Java

## Inconvénients assert

- Changement de comportement en production !
- Pas de distinction préconditions et autres vérifications
- En outre, gain de performance souvent négligeable



# Usage

- ⇒ Assertions à réserver pour tests couteux en temps
- Ou pour conditions assurées par déductions
  - $\neq$  préconditions : conditions assurées par contrat
  - Pour vérifier précondition ?

# Usage

- ⇒ Assertions à réserver pour tests couteux en temps
- Ou pour conditions assurées par déductions
  - $\neq$  préconditions : conditions assurées par contrat
  - Pour vérifier précondition ? Utiliser exception !

# Licence

Cette présentation, et le code LaTeX associé, sont sous [licence MIT](#). Vous êtes libres de réutiliser des éléments de cette présentation, sous réserve de citer l'auteur.  
Le travail réutilisé est à attribuer à [Olivier Cailloux](#), Université Paris-Dauphine.