

# Introduction aux tests dynamiques

- 1 Principes généraux
- 2 Les tests statiques
- 3 Introduction aux tests dynamiques**
- 4 Les tests fonctionnels
- 5 Les tests structurels
- 6 Les tests et les langages objets
- 7 Le TDD

# Les tests dynamiques

## Principe

Les tests dynamiques consistent à exécuter des jeux de test.

## Jeux de test

Un jeu de test est constitué d'un ensemble de couple :

- des données de test ;
- le résultat attendu.

# L'oracle

## Principe

Les résultats obtenus doivent être comparés aux résultats souhaités. Il faut un oracle pour faire fonctionner le test.

Il n'y a pas de méthode miracle. Conséquence du théorème de non-décidabilité de Gödel : il n'existe pas d'algorithme permettant d'affirmer l'exactitude d'un programme quelconque.

# Exemple d'oracle

## Un oracle en C

```
#include <assert.h>

void main() {
    assert(Max(0, 1) == 1);
    assert(Max(1, 0) == 1);
    assert(Max(1, 1) == 1);
}
```

# Un test de Macro

## La macro Max

On peut définir la macro Max avec le code suivant :

```
#define Max(a , b) a < b ? b : a
```

Est-ce correct ? Comment détecter/éviter ce genre d'erreur ?

# Exemple d'oracle

## Un oracle en Java

```
public class TestMax {  
  
    public static void main (String[] args){  
        Calculateur c = Calculateur.getInstance();  
        assert c.Max(0, 1) == 1 : "Test_échoué";  
        assert c.Max(1, 0) == 1 : "Test_échoué";  
        assert c.Max(1, 1) == 1 : "Test_échoué";  
    }  
}
```

# Exemple d'oracle

## Un oracle en Java avec JUnit 4

```
import org.junit.*;
import static org.junit.Assert.*;

public class TestMax {
    @Test
    public void test() {
        Calculateur c = Calculateur.getInstance();
        assertTrue(c.Max(0, 1) == 1);
        assertEquals(c.Max(1, 0), 1);
        assertTrue(c.Max(1, 1) == 1);
    }
}
```