# **Paire ou impair**

**Fait travailler** : *(ex :boucles, conditions, POO...)*

* boucles, conditions
* modulo

**Enoncé du Dojo :**

Un tableau d’au moins 3 entiers contenant des nombres (positifs et/ou négatifs). Ce tableau ne doit être constitué QUE de nombres pairs ou impairs sauf un. Le but est de retourné ce nombre qui sera “l’intrus”.

Remarque : zéro est un chiffre pair.

**Exemples** :

Entrée [5, -83, 12, 7, 971] Retour : 12 (tous les nombres sont impairs sauf 12)

Entrée [2, 3, 4] Retours : 3 (tous les nombres sont pairs sauf 3

**Correction :**

**class** PairImpair

{

**public function** evenOrOdd($tab)

{

**foreach** ($tab **as** $value){

*// si modulo vaut 0, le nombre est pair, sinon il est impair*

**if** ($value%2 == 0) {

$type = **'pair'**;

}**else** {

$type=**'impair'**;

}

$tabPairImpair[$type][] = $value;

}

*// si le tableau ne contient qu'un seul enregistrement (count =1),*

*// on renvoie cette seule ligne (index 0 du tableau)*

**if** (*count*($tabPairImpair[**'pair'**])==1) {

**return** $tabPairImpair[**'pair'**][0];

} **else** {

**return** $tabPairImpair[**'impair'**][0];

}

}

}

**Tests :**

**class** PairImpairTest **extends** TestCase

{

**public function** testEvenOrOdd()

{

$pairImpair = **new** PairImpair();

$this->assertEquals(2, $pairImpair-> evenOrOdd([1,2,3]));

$this->assertEquals(3, $pairImpair-> evenOrOdd([-2, 12, 3, 0]));

$this->assertEquals(4, $pairImpair-> evenOrOdd([1, 4, 3, 5]));

$this->assertEquals(4, $pairImpair-> evenOrOdd([-1, 4, 3, 3]));

$this->assertEquals(136, $pairImpair-> evenOrOdd([-125, 453, 136, 3]));

}

}