# **Géométrie**

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

* **POO**
* héritage, abstract...

**Enoncé du Dojo :**

Ce dojo n’est pas vraiment de l’algo, mais plus une mise en pratique des 1ers cours sur la POO

**Exemples** :

Créer une classe Shape (abstraite) contenant les méthodes perimeter et area

Créer les classes filles Rectangle et Circle implémentant perimeter et area. La classe Rectangle aura en plus une méthode isSquare indiquant si le rectangle est ou non un carré.

**Correction :**

*/\*\**

*\* Classe Shape*

*\*/*

**abstract class** Shape

{

**abstract public function** perimeter():float;

**abstract public function** area():float;

}

*/\*\**

*\* Classe Circle*

*\*/*

**class** Circle **extends** Shape

{

**private $radius**;

**public function** \_\_construct($radius)

{

$this->**radius** = $radius;

}

**public function** getRadius(): float

{

**return** $this->**radius**;

}

**public function** setRadius($radius)

{

$this->**radius** = $radius;

}

**public function** perimeter(): float

{

**return** 2\**pi*()\*$this->getRadius();

}

**public function** area(): float

{

**return** *pi*()\**pow*($this->getRadius(),2);

}

}

*/\*\**

*\* Classe Rectangle*

*\*/*

**class** Rectangle **extends** Shape

{

**private $length**;

**private $width**;

**public function** \_\_construct($length, $width)

{

$this->setLength($length);

$this->setWidth($width);

}

**public function** perimeter(): float

{

**return** ($this->getWidth() + $this->getLength()) \* 2;

}

**public function** getWidth(): float

{

**return** $this->**width**;

}

**public function** setWidth($width)

{

$this->**width** = $width;

}

**public function** getLength(): float

{

**return** $this->**length**;

}

**public function** setLength($length)

{

$this->**length** = $length;

}

**public function** area(): float

{

**return** $this->getWidth() \* $this->getLength();

}

**public function** isSquare(): bool

{

**return** $this->getLength() === $this->getWidth();

}

}

**Tests :**

*la correction du dojo doit toujours être donnée avec les tests correspondants*

**class** ShapeTest **extends** TestCase

{

**public function** testRectangle()

{

$rectangle= **new** Rectangle(6,4);

$this->assertEquals(20,$rectangle->perimeter());

$this->assertEquals(24,$rectangle->area());

$this->assertFalse($rectangle->isSquare());

}

**public function** testCarre()

{

$rectangle= **new** Rectangle(5,5);

$this->assertEquals(20,$rectangle->perimeter());

$this->assertTrue($rectangle->isSquare());

}

**public function** testCircle()

{

$circle= **new** Circle(5);

$this->assertEquals(10\**pi*(), $circle->perimeter());

$this->assertEquals(25\**pi*(), $circle->area());

}

}