**Enonce:**

Le but du dojo est d’afficher le score courant d’un match de tennis.

On a en entrée un tableau qui indique la séquence des points marqués.

Par exemple le tableau ["1", "1", "2", "2", "1"] indique que le joueur 1 a marqué les deux premiers points puis le joueur 2 en a marqué 2 et le joueur 1 marque le dernier point du jeu.

Le score est retourné sous la forme [nameJoueur1 => scoreJoueur1, nameJoueur2 => scoreJoueur2].

**Fonction:**

<?php

include 'kata\_object\_class.php';

function getScore($resultat, $J1, $J2) {

$tabscore=["0","15","30","40"];

for ($i=0; $i < count($resultat) ; $i++) {

if ($resultat[$i]==1) {

$J1->setScore($J1->getScore()+1);

}

else {

$J2->setScore($J2->getScore()+1);

}

}

return array($J1->getNom() => $tabscore[$J1->getScore()], $J2->getNom() => $tabscore[$J2->getScore()]);

}

function getScore($score, $player1, $player2){

$count = 0;

while (isset($score[$count])){

if ($score[$count]==1){

if ($player1->score >= 30)

$player1->score = 40;

else

$player1->score += 15;

}

if ($score[$count]==2){

if ($player2->score >= 30)

$player2->score = 40;

else

$player2->score += 15;

}

$count++;

}

return [$player1->name => $player1->score, $player2->name => $player2->score];

}

**Class:**

<?php

/\*\*

\* KataTennis attribut public

\*/

class Player

{

public $name;

public $score;

function \_\_construct($name, $score)

{

$this->name = $name;

$this->score = $score;

}

}

/\*\*

\* KataTennis attribut private

\*/

class Player\_2

{

private $name;

private $score;

function \_\_construct($name, $score)

{

$this->name = $name;

$this->score = $score;

}

public function getName(){

return $this->name;

}

public function setName($name)

{

$this->name = $name;

return $this;

}

public function getScore(){

return $this->score;

}

public function setScore($score)

{

$this->score = $score;

return $this;

}

}

**Tests:**

<?php

include 'kata\_object.php';

include 'kata\_object\_player.php';

class TennisTest extends PHPUnit\_Framework\_TestCase

{

public function testTrue()

{

$this->assertEquals(true, true);

}

public function test1()

{

$florian = new Player('Florian', 0);

$jack = new Player('jack', 0);

$this->assertEquals(['Florian' => 15, 'Jack' => 0], getScore([1], $florian, $jack));

}

public function test2()

{

$florian = new Player('Florian', 0);

$jack = new Player('jack', 0);

$this->assertEquals(['Florian' => 15, 'Jack' => 15], getScore([1,2], $florian, $jack));

}

public function test3()

{

$florian = new Player('Florian', 0);

$jack = new Player('jack', 0);

$this->assertEquals(['Florian' => 30, 'Jack' => 15], getScore([1,1,2], $florian, $jack));

}

public function test4()

{

$florian = new Player('Florian', 0);

$jack = new Player('jack', 0);

$this->assertEquals(['Florian' => 40, 'Jack' => 15], getScore([1,1,2,1], $florian, $jack));

}

public function test5()

{

$florian = new Player('Florian', 0);

$jack = new Player('jack', 0);

$this->assertEquals(['Florian' => 30, 'Jack' => 30], getScore([1,1,2,2], $florian, $jack));

}

public function test6()

{

$florian = new Player('Florian', 0);

$jack = new Player('jack', 0);

$this->assertEquals(['Florian' => 40, 'Jack' => 40], getScore([1,1,2,1,2,2], $florian, $jack));

}

}