Enoncé

Le but est de convertir un sigle non abbrégé en sa version abbrégée (majuscules séparées par des points). Exemple : ‘Wild Code School’ => ‘W.C.S.’

Dans un second temps, ajouter un 1er cas particulier

Société Nationale du Chemin de Fer Français doit donner ‘S.N.C.F.’ et non ‘S.N.D.C.D.F.F’

Tandis que

‘Electricité de France’ doit bien donner ‘E.D.F.’

Puis un autre cas particulier

‘World Wide Web Consortium’ => W.3.C. (et les autres sigles fonctionnent bien entendu toujours comme attendus)

CORRECTION

**class** Abbr

{

**public function** shorten($sentence)

{

$words = *explode*(**' '**,$sentence);

$nb = *count*($words);

$sigle = **""**;

**foreach** ($words **as** $word) {

**if** (*strlen*($word) > 3 || $nb<=4) {

$firstLetter = $word[0];

$sigle .= $firstLetter . **"."**;

}

}

$letters = *str\_split*($sigle, 2);

$previousLetter = $result = **""**;

$i = 1;

**foreach** ($letters **as** $letter){

**if**($letter == $previousLetter){

$i++;

} **else**{

**if** ($i > 1) {

$result .= $i. **"."** . $letter;

$i = 1;

} **else** {

$result .= $letter;

}

}

$previousLetter = $letter;

}

**return** *strtoupper*($result);

}

}

TESTS

**public function** testAbbr()

{

$abbr = **new** Abbr ();

$this->assertEquals (**'W.C.S.'**, $abbr->shorten(**'Wild Code School'**));

$this->assertEquals (**'W.C.S.'**, $abbr->shorten(**'wild code school'**));

$this->assertEquals (**'S.N.C.F.'**, $abbr->shorten(**'societe nationale du chemin de fer français'**));

$this->assertEquals (**'E.D.F.'**, $abbr->shorten(**'electricite de france'**));

$this->assertEquals (**'W.3.C.'**, $abbr->shorten(**'World Wide Web Consortium'**));

}