function testBefore(<Test> a, var b = 5, int c = 10)

{

a->method1();

return b + c;

}

namespace Test;

use RuntimeException as RE;

/\*\*

\* Example comment

\*/

class Test extends CustomClass implements TestInterface

{

const C1 = null;

// Magic constant: http://php.net/manual/ru/language.constants.predefined.php

const className = \_\_CLASS\_\_;

public function method1()

{

int a = 1, b = 2;

return a + b;

}

// See fn is allowed like shortcut

public fn method2() -> <Test>

{

call\_user\_func(function() { echo "hello"; });

[1, 2, 3, 4, 5]->walk(

function(int! x) {

return x \* x;

}

);

[1, 2, 3, 4, 5]->walk(

function(\_, int key) { echo key; }

);

array input = [1, 2, 3, 4, 5];

input->walk(

function(\_, int key) { echo key; }

);

input->map(x => x \* x);

return this;

}

}