

ВОЗМОЖНОСТИ УЛУЧШЕНИЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ PHP С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ ПРЕПРОЦЕССИНГА

Паневин Никита Борисович

студент, Тульский государственный педагогический университет им. Л. Н. Толстого, РФ, г. Тула

PHP – это один из распространённых язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в язык разметки HTML.

Разберем проблемы и улучшения, которые можно добавить в язык программирования PHP для уменьшения его кодовой базы.

Избыточные символы выделения блока инструкций

В грамматике языка PHP определены такие символы как «{» и «}», что являются символами начала и конца блока инструкций. Данный символ используется для указания блоков инструкций структур языка PHP, функций, условий, циклов и пользовательских блоков инструкций.

При решении данной проблемы было принято решение в реализации альтернативного варианта, который предполагает под собой определение границ через отступы в программном коде. Для визуального выделения начала блока (для классов и функций) будет использоваться символ «:».

Пример использования данного правила и эквивалентного кода на целевом языке представлены в таблице 1.

Таблица 1.

Уменьшение избыточных символов выделения блоков инструкций

Исходный язык	Целевой язык
<pre><?php class A: function foo(): if (isset(\$this)): echo 'Переменная \$this определена ('; echo get_class(\$this); echo ")\n"; else:</pre>	<pre><?php class A { function foo() { if (isset(\$this)) { echo 'Переменная \$this определена (';</pre>

<pre>echo "Переменная \\$this не определена.\n";</pre>	<pre>echo get_class(\$this); echo ")\n"; } else { echo "Переменная \\$this не определена.\n"; } }</pre>
--	---

Излишне длинные ключевые слова в синтаксисе функций

Для создания функций или методов класса в языке программирования PHP используется ключевое слово «function». Для уменьшения величины количества символов в строке исходного кода было принято решение, что данное ключевое слово можно сократить до «func».

Пример использования данного правила и эквивалентного кода на целевом языке представлены в таблице 2.

Таблица 2.

Уменьшение избыточности длины ключевого слова в синтаксисе функции

Исходный язык	Целевой язык
<pre><?php func foo() func bar() ...</pre>	<pre><?php function foo() { function bar() { ... } }</pre>

Избыточные ключевые слова в синтаксисе класса

При объявлении класса на языке программирования PHP в синтаксисе класса используются следующие ключевые слова:

- 1) extends;
- 2) implements;
- 3) public;

4) protected;

5) private.

extends

Ключевое слово «extends» используется для указания родительского класса. Для уменьшения величины количества символов в строке исходного кода было принято решение о том, что данное ключевое слово необходимо упростить, заменив его на круглые скобки «()», внутри которых будут описываться от какого класса было произведено наследование текущего.

Пример использования данного правила и эквивалентного кода на целевом языке представлены в таблице 3.

Таблица 3.

Уменьшение избыточности ключевого слова «extends» в синтаксисе класса

Исходный язык	Целевой язык
<pre><?php class B(A): ...</pre>	<pre><?php class B extends A { ... }</pre>

implements

Ключевое слово «implements» используется для реализации интерфейса для указанного класса. Для уменьшения величины количества символов в строке исходного кода было принято решение о том, что данное ключевое слово необходимо упростить, заменив его на квадратные скобки «[]» внутри которых будут описываться от какого класса было произведено наследование текущего.

Пример использования данного правила и эквивалентного кода на целевом языке представлены в таблице 4.

Таблица 4.

Уменьшение избыточности ключевого слова Implements в синтаксисе класса

Исходный язык	Целевой язык
<pre><?php class C[A]: ... class D[A, B]:</pre>	<pre><?php class C implements A { ... }</pre>

...	class D implements A, B { ... }
-----	--

Ключевые слова public, protected, private

Ключевые слова public, protected, private являются областями видимости, которые указывают на уровень доступ к свойствам и методам класса. Доступ к полям и методам экземпляра класса, объявленным как «public» (общедоступный), разрешён из любого места класса и за его пределами. Модификатор «protected» является защищённым типом доступа к данным и разрешает доступ ним самому классу, в котором он объявлен, наследующим его классам, а также родительским классам. Модификатор «private» представляет из себя закрытый тип доступа к данным и ограничивает область видимости так, что только класс, где объявлен сам элемент, имеет к нему доступ. Для уменьшения величины количества символов в строке исходного кода было принято решение о том, что вышеописанные ключевые слова необходимо упростить, заменив их на разметку UML следующим образом:

- «public» заменяется на символ «+»;
- «protected» заменяется на символ «№»;
- «private» заменяется на символ «-»;

Пример использования данного правила и эквивалентного кода на целевом языке представлены в таблице 7.

Таблица 7.

Уменьшение избыточности ключевых слов уровней доступа в синтаксисе класса

Исходный язык	Целевой язык
<pre><?php class C(A)[B]: +\$var1 -\$var2 +func func1(\$a, \$b): ... №func func2(\$a, \$b): ...</pre>	<pre><?php class C extends A implements B { public \$var1; private \$var2; public function func1(\$a, \$b) { ... }</pre>

```
protected function func1($a, $b)
```

```
{
```

```
...
```

```
}
```

```
}
```

Список литературы:

1. PHP Language Specification [Электронный ресурс]. – Режим доступа: <https://www.phplang.org/spec/19-grammar.html>. – Дата доступа: 26.06.2024.
2. Список меток (tokens) парсера [Электронный ресурс]. – Режим доступа: <https://www.phplang.org/spec/19-grammar.html>. – Дата доступа: 26.06.2024.
3. Область видимости [Электронный ресурс]. – Режим доступа: <https://php.net/manual/ru/language.oop5.visibility.php>. – Дата доступа: 26.06.2024.