

Лекция №1 по дисциплине
«Internet-технологии и язык
программирования Java»
на тему:

**Назначение и возможности языка
PHP. Переменные, константы и типы
данных PHP**

Структура дисциплины «Internet-технологии и язык программирования Java»

6-й семестр

Лк.: 26 ч. (13)

Лб.: 16 ч. (4)

Консультации

Самостоятельная работа

Контрольные точки: 2

Курсовая работа

Краткое содержание дисциплины

- Преимущества языка PHP. Синтаксис языка PHP
- Ассоциированные массивы в PHP
- Работа с файлами и каталогами в PHP
- Работа с базами данных в PHP
- Массивы и коллекции в языке Java
- Обработка исключений в языке Java
- Работа с базами данных с помощью JDBC
- Разработка графического интерфейса при помощи Java Server Page

Литература

- Котеров Д., Костарев А. PHP в подлиннике. 2-е издание. – СПб.: «ВНУ-СПб», 2013.- 1104 с.
- Кузнецов М.В. PHP5/6 в подлиннике. – СПб.: «ВНУ-СПб», 2010.- 1024 с.
- Веллинг Л., Томпсон Л. Разработка веб-приложений с помощью PHP и MySQL. 4-е издание.- М.: «Вильямс», 2012.- 848 с.
- Кузнецов М.В., Симдянов И.В. Самоучитель PHP5/6. 3-е издание.- СПб.: «БХВ-Петербург», 2011.- 672 с.
- Суэринг С., Конверс Т., Парк Д. PHP и MySQL. Библия программиста. 2-е издание.- М.: «Вильямс», 2010.- 912 с.
- Дронов В.А. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов.- СПб.: «БХВ-Петербург», 2009.- 544 с

Литература (продолжение)

- Хорстманн К., Корнелл Г. Java 2. Библиотека профессионала. Том 1. Основы. 8-е издание. - М.: “Диалектика-Вильямс”, 2012.- 816 с.
- Эккель Б. Философия Java.- 4-е издание.- СПб, «Питер», 2013.- 640 с.
- Шилдт Г. Java: Руководство для начинающих.- М.: “Диалектика-Вильямс”, 2012.- 624 с.
- Курняван Б. Программирование Web-приложений на языке Java. - М.: “Лори”, 2009.- 900 с.
- Монахов В. Язык программирования Java и среда NetBeans. 3-е издание.- СПб.: «ВНУ-СПб», 2012.-704 с.
- Машнин Т. Eclipse: разработка RCP-, Web-, Ajax- и Android-приложений на Java. СПб.: «ВНУ-СПб», 2013.- 384 с.

Курсовая работа

Тематика курсовых работ:

Разработка элементов программного обеспечения информационной системы

- Средства: PHP, Java
- Оболочки: Денвер, Eclipse
- Этапы:
 - согласование перечня входных и выходных сообщений;
 - согласование структуры БД;
 - разработка диаграммы классов;
 - программирование;
 - оформление пояснительной записки.

Назначение языка PHP

PHP — *personal home page* (личная домашняя страница). Это полноценный язык программирования, с помощью которого можно создавать развитые Web-узлы, обменивающиеся информацией с современными базами данных

- PHP - это язык программирования, используемый на стороне сервера, конструкции которого вставляются в HTML-текст.
- В отличие от обычного HTML-текста Web-страницы, программа на PHP не передается браузеру, но обрабатывается препроцессором PHP.

Свойства языка PHP

- ✓ PHP - динамично развивающийся язык программирования
- ✓ Язык PHP является универсальным
- ✓ Язык PHP является интерпретируемым языком программирования
- ✓ Язык PHP является скриптовым языком

История языка PHP

Первая версия PHP – 1994 г., автор
Rasmus Lerdorf

В 1995 г. автор выпустил вторую версию
языка, назвав ее PHP/FI.

1998 г. - версия PHP3, включавшая в себя
новый лексический анализатор Zend
авторов Zeev Surasky, Andi Gutmans.

PHP 4.0 вышел в 2000 г.

История языка PHP

5-я версия PHP была выпущена в 2004 г. Обновлено ядро (Zend Engine 2), введена поддержка XML. Доработаны функции ООП

План по созданию 6-й версии PHP был принят в 2006 г.

Изменения было решено вводить, постепенно добавляя их в новые выпуски PHP 5.1, 5.2, 5.3

Характеристики PHP

- ✓ Поддержка платформ Win32 (NT/2000/XP), UNIX, OS/2, QNX, MacOS
- ✓ совместимость с серверами: Apache (Win32, UNIX), phttpd, fhttpd, ISAPI, NSAPI
- ✓ поддержка технологий COM, XML, Java, Flash
- ✓ развитая функциональность для работы с сетевыми соединениями

Характеристики РНР

- ✓ поддержка свыше 20 БД и развитая функциональность для работы с ними
- ✓ возможность создания полноценных объектно-ориентированных приложений
- ✓ сравнительно простой синтаксис и удобство в практическом использовании
- ✓ бесплатность
- ✓ высокая производительность
- ✓ открытость кода

Размещение PHP-кода

<u>Вид тэгов</u>	<u>Открывающий тэг</u>	<u>Закрывающий тэг</u>
• Стандартные	<?php	?>
• Короткие	<?	?>
• ASP	<%	%>
• Программные	<SCRIPT LANGUAGE="php">	</SCRIPT>
• Специальные	<?=	?>

- Чтобы короткие тэги распознавались правильно, в файле `php.ini` должна быть включена директива `short_open_tag`.

```
short_open_tag = on;
```

- Для того чтобы разрешить использование тэгов в стиле ASP, нужно включить директиву `asp_tags`.

```
asp_tags = on;
```

Переменные и типы данных

- Имена переменных PHP начинаются со знака доллара, за которым следуют буквенно-цифровые символы и символы подчеркивания
- Язык PHP является слаботипизированным, т.е. переменные не требуют строгого задания типа при их объявлении
- В ходе выполнения программы тип переменной может быть почти всегда изменен неявным образом, без специальных преобразований

Перечень типов данных

Тип данных	Описание
boolean	Логический тип
integer	Целое число
double, float	Вещественное число
string	Строковый тип
array	Массив
object	Объект
resource	Дескриптор, позволяющий оперировать ресурсом
NULL	Специальный тип

Объявление переменных

- Пример.

```
$num = 25;
```

```
$num = 025; // восьмеричное число
```

```
$num = 0x25; // 16-ричное число
```

```
$f = 0.012;
```

```
$f = 1.2E-2;
```

```
$b=true;
```

Уничтожение переменной

- Пример вызова функции unset()
- `$user = "Alex";`
- `$num = 25;`
- `unset ($user, $num);`

Проверка существования переменной

- Для проверки существования переменной используется функция `isset()`
- Для проверки, является ли строка пустой или нет, используется функция `empty()`

Пример.

```
$str = '';
```

```
if(isset ($str)) // true
```

```
echo 'Переменная str существует';
```

```
if(empty ($str)) // true
```

```
echo 'Переменная str пустая';
```

Определение типа переменных

Имя функции	Описание
<code>gettype(\$var)</code>	Возвращает тип переменной <code>\$var</code>
<code>is_array(\$var)</code>	Возвращает <code>true</code> , если <code>\$var</code> является массивом
<code>is_double(\$var)</code>	Возвращает <code>true</code> , если <code>\$var</code> является переменной типа double
<code>is_int(\$var)</code>	Возвращает <code>true</code> , если <code>\$var</code> является переменной типа int
<code>is_object(\$var)</code>	Возвращает <code>true</code> , если <code>\$var</code> является объектом
<code>is_string(\$var)</code>	Возвращает <code>true</code> , если <code>\$var</code> является переменной типа string

Неявное приведение типов

- Использование переменной в контексте, где ожидается переменная другого типа, приводит к неявному преобразованию переменной к нужному типу.
- Если строка содержит число и используется в арифметическом выражении, то она автоматически будет приведена к числовому типу.
- **Пример.**

```
$str = "5.5";  
$n = $str + 2;  
echo $n; // 7.5
```

Явное приведение типов

- Можно явно преобразовать переменную к некоторому типу. Для этого существует несколько способов.
- Первый способ заключается в использовании круглых скобок.
- Второй способ – использование специальных функций
- **Пример.**

```
$f = 5.75;
```

```
$n = (int)f;
```

```
echo $n; // 5
```

Использование круглых скобок для приведения типов

Имя функции	Описание
<code>\$var = (int)var;</code>	Приведение к целому типу int
<code>\$var = (integer)var;</code>	Приведение к целому типу int
<code>\$var = (float)var;</code>	Приведение к вещественному типу double
<code>\$var = (double)var;</code>	Приведение к вещественному типу double
<code>\$var = (real)var;</code>	Приведение к вещественному типу double
<code>\$var = (string)var;</code>	Приведение к строковому типу string
<code>\$var = (object)var;</code>	Приведение к объекту
<code>\$var = (array)var;</code>	Приведение к массиву

Использование специальных функций для приведения типов

Имя функции	Описание
<code>settype(\$var, \$type)</code>	Преобразует переменную <code>\$var</code> к типу, указанному в параметре <code>\$type</code> , который может принимать значение “boolean,” “integer”, “int” и другие.
<code>floatval(\$var)</code>	Преобразует переменную <code>\$var</code> к вещественному типу
<code>intval(\$var)</code>	Преобразует переменную <code>\$var</code> к целому типу
<code>strval(\$var)</code>	Преобразует переменную <code>\$var</code> к строковому типу <code>string</code>

Константы

- Объявление константы осуществляется при помощи функции `define()`:

```
define($name, $value);
```

- Значением константы может быть логическое, целое, вещественное или строковое значение
- **Пример.**

```
define("num", 5);  
define("val", "КН-11-2");  
echo num;  
echo val;
```



Константы (продолжение)

- Функция `constant()` возвращает значение константы.
`echo constant("val");`
- Проверку существования константы выполняет функция `defined()`
`if(defined("val")) echo val;`

- Предопределенные константы.
`_LINE_` текущая строка в файле
`_FILE_` полный путь и имя текущего файла
`_FUNCTION_` имя функции
`_CLASS_` имя класса
`_METHOD_` имя метода класса

- **Пример.**
`echo "Имя файла " . _FILE_ . "
";`

Функция echo ()

- Синтаксис:
echo(\$str1 [, \$str2...]) ;
- Пример.
\$kol = 15;
echo "Присутствуют", \$kol, "студентов";
- То же с использованием оператора «точка»:
echo "Присутствуют". \$kol . "студентов";
- Использование интерполяции переменной:
echo "Присутствуют \$kol студентов";
- Вместо функции echo() можно использовать функцию **print()**. Однако она принимает только один аргумент.