

Несмотря на все преимущества, которые дают нам типы товара, в VirtueMart заложена только основа для работы с ними, прорабатывать эту функцию разработчики не стали. С одной стороны это правильно - каждый делает себе так как ему нравится, с другой - некоторые базовые возможности стоило бы добавить сразу.

1. Для начала выведем параметры типа товара в шаблон категории. Для этого достаточно внести небольшие изменения в файл **administrator/components/com_virtuemart/html/shop.browse.php**

```
require_once( CLASSPATH.'ps_product_type.php' ); $ps_product_type = new
ps_product_type(); if ($product_parent_id!=0 &&
!$ps_product_type->product_in_product_type($db_browse->f("product_id"))) { $product_type
= $ps_product_type->list_product_type($product_parent_id); } else { $product_type =
$ps_product_type->list_product_type($db_browse->f("product_id")); }
$products[$i]['product_type'] = $product_type;
```

После вставки этого кода, мы можем заходить в свой шаблон browse_N.php и в нужном месте выводить параметры для типа:

2. Следующая проблема заключается в том, что при выводе параметров, выводятся все параметры данного типа, даже пустые. Это нужно исправить.

Для этого будем давать параметру значение null, при необходимости введя его в список допустимых значений. А в файле **components/com_virtuemart/themes/default/templates/common/product_type.tpl.php** добавим условие:

```
if($product_type_params["parameter_value"] != "null") { выводим параметр; }
```

3. Теперь, собственно, сделаем так, что бы можно было выбрать - какие параметры выводить в карточке товара, а какие в шаблоне категории. Для этого вставим чекбокс в админпанель редактирования параметра с названием: "Выводить параметр в шаблоне категории?". Если чекбокс будет выбран, то этот параметр будет выводиться как в карточке, так и в категории. Если нет - только в карточке.

Сначала внесем изменения в файл **administrator/components/com_virtuemart/html/product.product_type_parameter_form.php**. Ищем участок кода:

Ниже, вставим следующий код:

```
: />
```

Добавим языковую переменную в файл **administrator/components/com_virtuemart/languages/product/russian.php** :

```
'PHPSHOP_PRODUCT_TYPE_PARAMETER_INCATEGORY' => 'Выводить параметр в шаблоне категории?'
```

Теперь нам нужно создать новое поле в таблице `_vm_product_type_parameter` (не забываем про префикс). Идем в PHPMyAdmin и выполняем следующий запрос:

```
ALTER TABLE `jos_vm_product_type_parameter` ADD `parameter_incATEGORY`  
varchar(50) NOT NULL default 'N'
```

Следующий шаг - правка файла **administrator/components/com_virtuemart/classes/ps_product_type_parameter.php**. Тут необходимо дополнить функции `add_parameter` и `update_parameter` следующей строчкой:

```
'parameter_incATEGORY' => vmGet($d, 'parameter_incATEGORY'),
```

И дальше передача значения нашего поля в файл `product_type.tpl.php`. Осуществляется она в файле **administrator/components/com_virtuemart/classes/ps_product_type.php**. Ищем строку:

```
$product_type_param[$i]["parameter_unit"] = $dba->f("parameter_unit");
```

И под ней сразу пишем:

```
$product_type_param[$i]["parameter_incATEGORY"] = $dba->f("parameter_incATEGORY");
```

Вот, в общем то несложные действия, которые решают нашу проблему. Осталось только правильно поставить условие в шаблоне вывода параметров. Открываем файл **components/com_virtuemart/themes/default/templates/common/product_type.tpl.php** и где-то сверху пишем:

```
$Call = debug_backtrace(); $VMpage = JRequest::getVar('page', NULL);
```

Ниже выводим параметры товара. Включая наши условия должно получиться как то так:

```
if($product_type_params["parameter_value"] != "null") {    if($VMpage ==  
"shop.product_details" && $Call[3]["function"] == "include") {    ?>
```