

Замена размера товаров прямо в корзине

Для подключения используем [cart.php](#)

```
<?php

defined('HOSTCMS') || exit('HostCMS: access denied.');
```

*/**
* Замена размера товаров прямо в корзине
*
* Версия для HostCMS v.6x
* @author KAD Systems
*
* Копирование и использование файлов модуля
* в коммерческих целях ЗАПРЕЩЕНО
*
/

```
class Kad_Cart_Changesize
{
    // Имя передаваемого массива замены через POST
    public $input_name = 'mod';

    public function __construct()
    {

    }

    /*      3
    *   Получить информацию о доп. свойстве модификаций переданного товара или
    модификации
    *   Применяется в корзине для реализации смены размера прямо в корзине
    *   Возвращает:
    *   массив объектов Property_Value
    */
    static function getModificationsPropertyValues($shop_item_id,
$property_id)
    {
        $aPropertyValues = array();

        $oShopItem = Core_Entity::factory('shop_item', $shop_item_id);
        if ($oShopItem->modification_id != )
        {
            $oShopItem = Core_Entity::factory('shop_item',
$oShopItem->modification_id);
        }

        $oModifications =
```

```

Core_Entity::factory('shop_item')->getAllByModification_id($oShopItem->id);

    foreach($oModifications as $oModification)
    {
        $oProperty = Core_Entity::factory('Property', $property_id);
        $aModificationPropertyValues =
$oProperty->getValues($oModification->id);

        foreach ($aModificationPropertyValues as $oPropertyValue)
        {
            $aPropertyValues[$oModification->id] = $oPropertyValue;
        }
    }

    return $aPropertyValues;
}

/*      2
*   Генерация xml для списка доп. свойств товара
*   Применяется в корзине для реализации смены размера прямо в корзине
*   Возвращает:
*   объект Core_Xml_Entity в случае если значения есть
*   иначе false
*/
static function getXmlModPropertyValues($shop_item_id, $property_id)
{
    $aPropertyValues =
self::getModificationsPropertyValues($shop_item_id, $property_id);

    if (count($aPropertyValues))
    {

        $oXml =
Core::factory('Core_Xml_Entity')->name('shop_item')->addAttribute('id',
$shop_item_id);

        foreach ($aPropertyValues as $id => $oPropertyValue)
        {
            $oXml->addEntity($oPropertyValue);
        }

        if (isset($aPropertyValues[$shop_item_id]))
        {
            $oXml->addEntity(
Core::factory('Core_Xml_Entity')->name('current')->value($aPropertyValues[$s
hop_item_id]->id));
        }

        return $oXml;
    } else
    {

```

```

        return false;
    }
}

/*      1
*   Генерация xml для списка доп. свойств товаров корзины
*   Применяется в корзине для реализации смены размера прямо в корзине
*   Возвращает:
*   объект Core_Xml_Entity
*/
public function getXmlPropertyValuesCartItems($oShop, $property_id)
{
    $oShop_Cart_Controller = Shop_Cart_Controller::instance();
    $oItemsInCart = $oShop_Cart_Controller->getAll($oShop);

    $oXml = Core::factory('Core_Xml_Entity')->name('property_values');
    foreach ($oItemsInCart as $oItemInCart)
    {
        $oXmlItemPropertyValues =
self::getXmlModPropertyValues($oItemInCart->shop_item_id, $property_id);
        if ($oXmlItemPropertyValues)
        {
            $oXml->addEntity($oXmlItemPropertyValues);
        }
    }

    return $oXml;
}

/*      2
*   Замена товара в корзине
*   Применяется в корзине для реализации смены размера прямо в корзине
*   Возвращает:
*   true при успешной замене
*/
static function changeCartItem($shop_item_id, $shop_item_change_id)
{
    $oShop_Cart_Controller = Shop_Cart_Controller::instance();
    $oShopCartItem =
$oShop_Cart_Controller->shop_item_id($shop_item_id)->get();

    $quantity = $oShopCartItem->quantity;
    $oShopCartItem->delete();

    $oShop_Cart_Controller
        ->shop_item_id($shop_item_change_id)
        ->quantity($quantity)
        ->add();
}

```

```

/*      1
*   Проверка и смена всех товаров, для которых был изменен размер
*   Применяется в корзине для реализации смены размера прямо в корзине
*   Возвращает:
*       true при успешной замене
*/
public function checkCartItems($oShop)
{
    $oShop_Cart_Controller = Shop_Cart_Controller::instance();
    $oItemsInCart = $oShop_Cart_Controller->getAll($oShop);
    $aMod = Core_Array::getPost('mod');

    foreach ($oItemsInCart as $oItemInCart)
    {
        $oShopCartItem =
$oShop_Cart_Controller->shop_item_id($oItemInCart->shop_item_id);

        if (isset($aMod[$oItemInCart->shop_item_id]) &&
            $oItemInCart->shop_item_id !=
$aMod[$oItemInCart->shop_item_id] &&
            $newId = $aMod[$oItemInCart->shop_item_id])
        {
            self::changeCartItem($oItemInCart->shop_item_id, $newId);
        }
    }
}
}

```