Разные цены товаров для разных складов

Хотим выводить разные цены для разных складов. Делаем следующее:

- Создаем цены с такими же названиями, какие заданы для складов. Цены создаются в справочниках интернет-магазина (при создании цены устанавливаем флажок «Установить для всех товаров»).
- Добавляем наблюдатель для методов класса `Shop_Item_Controller`, в котором определеяется цена товара. Наблюдатель будет заменять цену товара в зависимости от выбранного основного склада. Для этого в файле `bootstrap.php` добавляем код:

```
<?php
// ID интернет-магазина
define('REGION_PRICES_SHOP_ID', 4);

// KAD: Подменяем цену товара в зависимости от текущего склада
Core_Event::attach('Shop_Item_Controller.onBeforeCalculatePrice',
array('Kad_Shop_Item_Observers_Regionprices',
'onBeforeCalculatePrice'));
Core_Event::attach('Shop_Item_Controller.onBeforeCalculatePriceInItemCu
rrency', array('Kad_Shop_Item_Observers_Regionprices',
'onBeforeCalculatePrice'));
```

  Константа `REGION_PRICES_SHOP_ID` задает идентификатор интернет-магазина, для которого производится подмена.

- В каталоге `/modules/kad/shop/item/observers/` создаем файл `regionprices.php` с таким содержимым:

```
<?php

/*
* @author Maxim Zasorin, KAD Systems (©) 2015
* @date 18-11-2015
*
* В bootstrap.php:
*
// KAD: Подменяем цену товара в зависимости от текущего склада
Core_Event::attach('Shop_Item_Controller.onBeforeCalculatePrice',
array('Kad_Shop_Item_Observers_Regionprices',
'onBeforeCalculatePrice'));
Core_Event::attach('Shop_Item_Controller.onBeforeCalculatePriceInItemCu
rrency', array('Kad_Shop_Item_Observers_Regionprices',
'onBeforeCalculatePrice'));

* Копирование и использование файлов модуля
* в коммерческих целях ЗАПРЕЩЕНО
*
*/
defined('HOSTCMS') || exit('HostCMS: access denied.');
```

```php
class Kad_Shop_Item_Observers_Regionprices
{
    static public function onBeforeCalculatePrice($object, $args)
    {
        if (
                defined('REGION_PRICES_SHOP_ID')
                &&
                REGION_PRICES_SHOP_ID == $args[]->shop_id
            )
        {
            // Товар
            $oShopItem = $args[];

            // Текущий склад
            $oShop = Core_Entity::factory('Shop',
REGION_PRICES_SHOP_ID);
            $oCurrentWarehouse = $oShop->Shop_Warehouses->getDefault();

            // Цена для текущего склада
            $oShopPrice = Core_Entity::factory('shop_price');
            $oShopPrice
                ->queryBuilder()
                    ->where('name', '=', $oCurrentWarehouse->name);

            // print $oCurrentWarehouse->name;

            $aoShopPrices = $oShopPrice->findAll();

            if (count($aoShopPrices) > )
            {
                $oPriceForWarehouse = $aoShopPrices[];

                // Цена товара для текущего склада
                $oShopItemPrices = $oShopItem->Shop_Item_Prices;
                $oShopItemPrices->queryBuilder()
                        ->where('shop_price_id', '=',
$oPriceForWarehouse->id);

                $aoShopItemPrice = $oShopItemPrices->findAll();

                // Если цена для товара не установлена, то возможно это
модификация,
                // поэтому просматриваем цены для родительского товара
                // if (count($aoShopItemPrice) == 0 &&
$oShopItem->modification_id != 0)
                // {
                //     $oParentShopItem =
Core_Entity::factory('shop_item', $oShopItem->modification_id);
                //     $oShopItemPrices =
$oParentShopItem->Shop_Item_Prices;
```

```
                //     $oShopItemPrices->queryBuilder()
                //         ->where('shop_price_id', '=',
$oPriceForWarehouse->id);

                //     $aoShopItemPrice = $oShopItemPrices->findAll();
                // }

                // Цена для склада все-таки есть
                if (count($aoShopItemPrice) > )
                {
                    $oShopItemPrice = $aoShopItemPrice[];
                    $price = $oShopItemPrice->value;

                    // Устанавливаем цену склада
                    $object->setAPrice(
                        array(
                            'tax' => ,
                            'rate' => ,
                            'price' => $price,
                            'price_discount' => $price,
                            'price_tax' => $price,
                            'discount' => ,
                            'discounts' => array()
                        )
                    );
                }
            }
        }
    }
}
```