

УДК 519.95

РАЗРАБОТКА КОМПОНЕНТА SOLDTOGETHER ДЛЯ СОЗДАНИЯ СКИДОЧНЫХ КОМПЛЕКТОВ ТОВАРОВ С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ MAGENTO

© О.В. Крючин, К.Р. Хабирова, Д.О. Квашенкин

Ключевые слова: информационная система; интернет-магазин; система управления.

Описан процесс создания компонентов товаров, предоставляющих скидку. В качестве базового движка описываемой системы выбран Magento. Приведен пошаговый алгоритм реализации модуля и интерфейса для него.

ВВЕДЕНИЕ

Цели и задачи. *Magento* – это система управления интернет-магазинами, являющаяся на сегодняшний день самой популярной в мире. Изначально разрабатываемая на Украине, в июне 2011 г. она была приобретена компанией *eBay Inc* [1, 2].

Интернет-магазины, базирующиеся на движке *Magento*, обладают большим количеством функций, однако, для некоторых задач ее возможностей недостаточно. В частности, весьма актуальной для России является возможность предоставления скидки на какой-либо товар при покупке вместе с ним одного или нескольких других, т. е. формирование т. н. скидочных комплектов товаров.

Целью данной работы является создание компонента *SoldTogether*, позволяющего добавлять в интернет-магазин, созданный на основе движка *Magento*, комплекты товаров, в рамках которых на первый из них предоставляется скидка.

Общее описание компонента *SoldTogether*. Каждый продукт может содержать произвольное число комплектов, в рамках которых его цена существенно ниже обычной. Например, обычная цена принтера *HP LaserJet 1022* составляет 5100 руб., а его цена в комплекте – 4700 руб. Данный продукт имеет два комплекта. Первый состоит из картриджа (стоящего 1050 руб.) и мышки (250 руб.), а второй – из *USB*-привода (350 руб.), наушников (200 руб.) и микрофона (120 руб.). Таким образом, покупая отдельно принтер, человек платит 5100, а покупая совместно с ним мышку и картридж – 6450 (4700+1500+250) руб., экономя тем самым 400 руб.

Для реализации описанной выше модели были разработаны несколько классов – *SoldComplexItem*, *SoldComplex* и *SoldProduct*, чья архитектура будет подробно рассмотрена ниже. Класс *SoldProduct* описывает сам продукт (в описанном выше примере – принтер *HP*), *SoldComplex* – каждый комплект продукта (т. е. объект принтера содержит для члена – экземпляра этого класса), а *SoldComplexItem* – каждый элемент комплекта (т. е. первый состоит из двух: мыши и картриджа, а второй – из трех: клавиатуры, микрофона, наушников).

Структура компонента состоит из нескольких частей: фронтенда (пользовательского интерфейса), интерфейса администратора, серверного модуля и таблицы в базе данных, которая хранит SKU товаров, принадлежащих скидочным комплектам.

SKU (Stock Keeping Unit) – идентификатор товарной позиции (учетная единица / складской номер), который в торговле является идентификатором товара, применяется для учета и отслеживания статистики по реализованным товарам. Понятие пришло в отечественный мерчандайзинг вместе с сетевым ритейлом, до этого использовался термин «складская единица». Каждой принимаемой на склад и продаваемой через магазин позиции, будь то товар, иной вариант упаковки товара, комплект товаров (продаваемых по акции вместе), назначается свой SKU. Для предприятия, которое продает товары, количество SKU – это количество отличающихся друг от друга по своим свойствам обособленных товаров.

Для поставщика магазина каждая SKU (номенклатурная позиция) является объектом управления запасами: анализируются, восполняются, возвращаются, реализуются товарные запасы, анализируется оборачиваемость складских запасов по группам товаров и по ассортиментам, по каждой позиции SKU [3].

СОЗДАНИЕ НОВОГО АТТРИБУТА

Для создания возможности назначения отдельной цены товара при покупке комплекта ему необходимо добавить новый атрибут. Сделать это можно двумя способами – через управление атрибутами в панели управления и через вызов инсталлятора в исходном коде. Ниже будут рассмотрены оба варианта.

Как известно, атрибуты в *Magento* – описания или показатели, характеризующие товар, от цвета до производителя или артикула – делятся на два типа: простые и системные. По умолчанию *Magento* включает все необходимые системные атрибуты, которые не могут быть удалены и должны быть определены у каждого товара. Таким образом, каждый набор должен включать все системные атрибуты, которые содержат имя (*Name*), цену (*Price*) и номер (*SKU*), без которых товар не может быть использован, а простые атрибуты могут быть созданы владельцем магазина и обычно

Рис. 1. Снимок экрана с набором свойств добавляемого атрибута

являются характерными для конкретных товаров свойствами и потому не обязаны быть включены в каждый набор.

Создание атрибута средствами панели управления. Чтобы создать атрибут через панель управления, необходимо перейти в «Управление Атрибутами» (Catalog->Attributes->Manage Attributes) и щелкнуть по кнопке «Add New Attribute» в правом верхнем углу страницы.

Как известно, каждый атрибут имеет ряд свойств (Attribute Properties):

1) идентификатор атрибута (Attribute Identifier) – имя, используемое в системе – отображается в списке атрибутов в колонке «Attribute Code» и используется при управлении наборами атрибутов, но не отображается во фронтэнде; каждый созданный атрибут должен иметь уникальный идентификатор, не содержащий пробелов; идентификатор создаваемого атрибута назван *complect_price*;

2) область действия (Scope) – свойство, определяющее уровни, на которых значения этого атрибута будут одинаковым для данного вида товара. Он может быть трех видов: глобальный (Global – значение данного атрибута должно быть одинаково для всех товаров данного вида, вне зависимости от того, на каком сайте или в каком виде магазина он находится), уровень сайта (Website – позволяет менять значения атрибута в зависимости от того, на каком сайте он расположен) или уровень вида магазина (Store View – у одного товара на данном сайте); для атрибута *complect_price* выбрано свойство Website;

3) тип данных (Catalog Input Type) – описание типа данных, которые будут храниться в атрибуте – определяет метод ввода данных (варианты можно редактировать на вкладке «Manage Label / Options tab»); в качестве типа атрибута *complect_price* выбран Price;

4) значение по умолчанию (Default Value) – если

установить его, то каждый новый товар с данным атрибутом будет иметь предустановленное значение, определяемое данным свойством; данное свойство не требуется для атрибута *complect_price*;

5) уникальное значение (Unique Value) – если его установить, то значения (выбранные из списка или введенные вручную) для каждого товара должны быть различными;

6) непустой (Values Required) – если атрибут устанавливается как непустой, то необходимо устанавливать его значение для каждого созданного товара; товар может не иметь комплектов, поэтому данное свойство не установлено для атрибута *complect_price*;

7) проверка значения (Input Validation for Store Owner) – Magento будет проверять соответствие введенных данных типу, на который указывает данное свойство, если выбрать ответ «None», то в значение данного атрибута можно будет ввести информацию любого вида; для атрибута *complect_price* установлена валидация на соответствие формату цены;

8) применить к (Apply To) – определяет, какие товарные типы будут включать данный атрибут, в Magento существует 3 товарных типа: простые (Simple), группированные (Grouped) и изменяемые товары (Configurable Products); создаваемый атрибут будет применяться ко всем простым товарам;

9) использовать для создания изменяемых продуктов (Use To Create Configurable Product) – данное свойство станет доступным, только если область действия атрибута установлена как глобальная, поэтому в данном случае оно недоступно.

Для взаимодействия с фронтендом также имеется ряд свойств:

1) использовать в быстром поиске (Use in Quick Search) – если выбрано «Yes», то при использовании строки поиска в шапке страницы Magento будет искать в т. ч. и в значениях этого атрибута; данное свойство не установлено для создаваемого атрибута;

2) использовать в расширенном поиске (*Use in Advanced Search*) – позволяет искать среди значений атрибута расширенным поиском (в расширенном поиске для каждого атрибута будет иметься свое собственное поле ввода, вместо одного для всех, и если установить значение «Yes», то для данного атрибута на странице расширенного поиска будет создано поле ввода поискового запроса); для создаваемого атрибута данное свойство установлено;

3) возможность сравнения во фронт-энде (*Comparable on Front-end*) – при выборе значения «Yes» в окне сравнения товаров будет создан ряд для этого атрибута; данное свойство включено для атрибута *complex_price*;

4) использовать в навигации по фильтрам (*Use in Layered Navigation*) – позволяет покупателю перейти лишь к определенной части товаров используя любой атрибут, который может служить фильтром (меню навигации по фильтрам располагается слева на странице категории и содержит все «фильтруемые» атрибуты, под каждым из них отображены принимаемые им значения и при нажатии на какое-либо из них все товары данной категории будут отфильтрованы так, что будут отображаться лишь те товары, у которых значение данного атрибута равно фильтру – существуют два типа

фильтруемых атрибутов, которые можно выбрать в соответствующем выпадающем окне: фильтруемый с результатами и фильтруемый без результатов; данное свойство включено для атрибута *complex_price*;

5) позиция (*Position*) – определяет положение этого атрибута в меню навигации по фильтрам по отношению к другим;

6) видимость на странице каталога во фронт-энде (*Visible on Catalog Pages on Front-end*) – свойство доступно только для простых атрибутов (на странице о товаре, под описанием товара находится секция дополнительной информации, которая не содержит имени всех видимых атрибутов и их значения, если у товара нет ни одного видимого простого атрибута, секция дополнительной информации отображаться не будет); данное свойство включено для атрибута *complex_price*.

Создание атрибута средствами инсталлятора.

Для создания атрибута средствами инсталлятора в него добавлена приватная функция *createNewAttribute()*, создающая новый атрибут. В результате в товар будет добавлена новая группа «Complex Price» – рис. 1.

```
function createNewAttribute() {
    $attrCode = 'complexprice';
    $attrGroupName = 'Complex Price';
    $attrLabel = 'Complex Price';
    $attrNote = 'Complex Price';

    $objCatalogEavSetup = Mage::getResourceModel('catalog/eav_mysql4_setup', 'core_setup');
    $attrIdTest =
        $objCatalogEavSetup->getAttributeId(Mage_Catalog_Model_Product::ENTITY, $attrCode);

    if ($attrIdTest === false) {
        $objCatalogEavSetup->addAttribute(Mage_Catalog_Model_Product::ENTITY, $attrCode, array(
            'group' => $attrGroupName,
            'sort_order' => 7,
            'type' => 'varchar',
            'backend' => '',
            'frontend' => '',
            'label' => $attrLabel,
            'note' => $attrNote,
            'input' => 'text',
            'class' => '',
            'source' => '',
            'global' => Mage_Catalog_Model_Resource_Eav_Attribute::SCOPE_GLOBAL,
            'visible' => true,
            'required' => false,
            'user_defined' => true,
            'default' => '0',
            'visible_on_front' => false,
            'unique' => false,
            'is_configurable' => false,
            'used_for_promo_rules' => true
        ));
    }
}
```

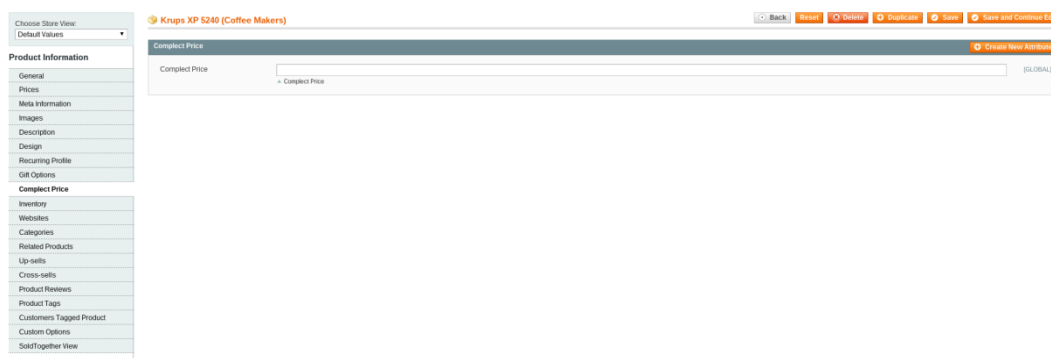


Рис. 2. Снимок экрана с ценой товара в комплекте

ПАНЕЛЬ АДМИНИСТРИРОВАНИЯ

Интерфейс панели администрирования. Рабочее пространство *SoldTogether* состоит из двух частей – списка комплектов и таблицы продуктов. В первой из них отображается набор комплектов продукта (рис. 3) над которым можно производить операции трех видов: добавлять, удалять и редактировать.

Sold together

Delete	Test_Sold	Brother 9320CW	Brother MFC-8370DN	Submit
131	3464536747564754	4977766672177	4977766672072	
Edit		Edit	Edit	

Delete	Pentax X90	Brother MFC-8370DN	Brother 9320CW	Submit
132	27075164512	4977766670272	4977766672177	
Edit	Edit	Edit	Edit	

Delete	Pentax X90	Test_Sold	Brother MFC-9120CN	Submit
134	27075164512	3464536747564754	4977766672047	
Edit	Edit	Edit	Edit	

Delete	Brother MFC-9970CDW	Pentax X90	Submit
142	4977766696393	27075164512	
Edit	Edit		

Delete	Brother MFC-9970CDW	Krups XP 5240	Submit
139	2112121212121	0010942208266	
Edit	Edit		

Add complect

Рис. 3. Снимок экрана с набором комплектов продукта

Delete	Test_Sold	Brother 9320CW	Brother MFC-8370DN	Submit
131	3464536747564754	4977766672177	4977766672072	
Edit		Edit	Edit	

Delete	Pentax X90	Brother MFC-8370DN	Brother 9320CW	Submit
132	27075164512	4977766670272	4977766672177	
Edit	Edit	Edit	Edit	

Delete	Pentax X90	Test_Sold	Brother MFC-9120CN	Submit
134	27075164512	3464536747564754	4977766672047	
Edit	Edit	Edit	Edit	

Delete	Brother MFC-9970CDW	Pentax X90	Submit
142	4977766696393	27075164512	
Edit	Edit		

Delete	Brother MFC-9970CDW	Krups XP 5240	Submit
139	2112121212121	0010942208266	
Edit	Edit		

Select	October Bundle	Select	Submit
Select	289182837412	Select	

Add complect

Рис. 4. Снимок экрана с добавлением нового комплекта в набор

Delete	Test_Sold	Brother 9320CW	Brother MFC-8370DN	Submit
131	3464536747564754	4977766672177	4977766672072	
Edit		Edit	Edit	

Delete	Pentax X90	Brother MFC-8370DN	Brother 9320CW	Submit
132	27075164512	4977766670272	4977766672177	
Edit	Edit	Edit	Edit	

Delete	Pentax X90	Brother MFC-9120CN	Brother MFC-9120CN	Submit
134	27075164512	4977766672047	4977766672047	
Edit	Edit	Edit	Edit	

Delete	Brother MFC-9970CDW	Pentax X90	Submit
142	4977766696393	27075164512	
Edit	Edit		

Delete	October Bundle	Submit
143	289182837412	
Edit		

Delete	Brother MFC-9970CDW	Krups XP 5240	Submit
139	2112121212121	0010942208266	
Edit	Edit		

Рис. 4. Снимок экрана с редактированием комплекта

Для добавления нового комплекта необходимо нажать на кнопку «Add complect», расположенную под набором комплектов, после чего появляется пустой комплект, ячейки которого содержат только кнопку «Select». При нажатии на эту кнопку открывается таблица продуктов, из которой необходимо выбрать один, щелкнув на нем. После этого продукт заносится в ячейку, как это показано на рис. 4. После того, как выбраны все товары, необходимо нажать на кнопку «Submit», расположенную в правой части набора, которая отвечает за сохранение. После нажатия на эту кнопку комплект заносится в базу данных и его можно редактировать.

Для удаления комплекта необходимо нажать на его кнопку «Delete», расположенную в левой части.

Для редактирования элемента комплекта нужно нажать на кнопку «Edit», расположенную в левой части этого элемента, после чего откроется таблица выбора товаров. После выбора нового товара он заносится в редактируемую ячейку, а комплект изменяет цвет – фон становится темным (рис. 4), а кнопка «Submit» – активной. После нажатия на эту кнопку происходит занесение изменений в базу данных.

Техническая реализация панели управления на стороне клиента. Реализация функций на стороне клиента осуществлена с использованием библиотеки *jQuery* (для решения проблемы конфликта вызывается функция *jQuery.noConflict()*).

Как известно, *jQuery* – это библиотека *JavaScript*, фокусирующаяся на взаимодействии *JavaScript* и *HTML* и помогающая легко получать доступ к любому элементу *DOM*, обращаться к атрибутам или содержимому элементов *DOM*, а также манипулировать ими. Кроме того, библиотека *jQuery* предоставляет удобный *API* для работы с *AJAX*.

Точно так же, как *CSS* отделяет визуализацию от структуры *HTML*, *jQuery* отделяет поведение от струк-

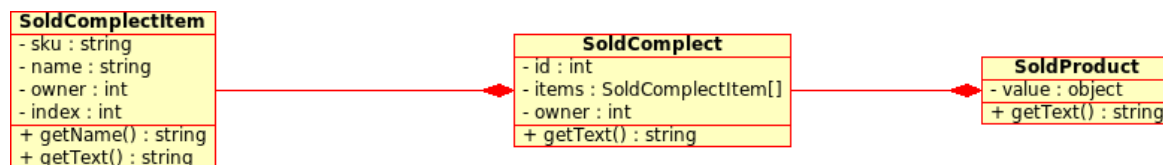


Рис. 5. Диаграмма классов панели управления комплектами товара

туры *HTML*. Например, вместо прямого указания на обработчик события нажатия кнопки управление передается *jQuery*, которая идентифицирует кнопки и затем преобразовывает его в обработчик события клика. Такое разделение поведения и структуры также называется принципом ненавязчивого *JavaScript*.

Библиотека *jQuery* содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения в *jQuery* функций, которые подошли бы всюду, поскольку это привело бы к большому коду, большая часть которого не востребована, поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов, что позволяет собрать для ресурса именно ту *JavaScript*-функциональность, которая на нем была бы востребована [4].

Значение компонента *soldOperation* (*hidden*) определяет выполняемую операцию – *add/remove/update*.

При добавлении нового комплекта (нажатия кнопки «Add complect») вызывается функция *addComplect()*, которая, в свою очередь, вызывает открытие таблицы товаров. При нажатии на кнопку «Select» для выбора товара в новый комплект вызывается функция *selectProduct(index)*, которая в качестве параметра принимает индекс элемента (1, 2 или 3). Для отправки данных на сервер предназначена функция *createSold()*, которая вызывает функцию *Magento productForm.submit()*. Компонент *soldOperation* имеет значение *add* по умолчанию, поэтому нет необходимости устанавливать его.

Для удаления элемента вызывается функция *removeSold(id)*, принимающая в качестве параметра идентификатор удаляемого комплекта. Эта функция устанавливает значение *remove* в компонент *soldOperation* и присваивает передаваемый идентификатор компоненту *u*.

Для редактирования комплекта предназначена функция *editSoldItem(complect, item)*, принимающая в качестве параметров идентификатор комплекта и номер редактируемого элемента. Данная функция присваивает эти значения компонентам *u* и *x* и открывает список товаров. После редактирования товаров для подтверждения изменений предназначена функция *submitSold(complect)*, принимающая в качестве параметра идентификатор редактируемого комплекта и присваивающая его компоненту *u* и значение *update* компоненту *soldOperation*.

Для обработки выбора товара из таблицы предназначена функция *checkElement*, которая вызывается при нажатии на товар (активацию флажка компонента *checkbox*) и ищет первый отмеченный товар. Найдя его, функция передает соответствующее имя товара в заданный элемент *name_* и значение SKU в элементы *sku_* и *skuValue_* (для обозначения индексов этих элементов служат значения компонентов *u* и *x*). После

этого изменяется внешний вид редактируемого комплекта и происходит переключение вкладок (таблица товаров закрывается, а список комплектов открывается).

Функция *buildCheckers(checkers)* отвечает за перестройку флажков *checkbox*. В качестве параметра она принимает список компонентов, которые нужно переформировать. К каждому из них применяются три действия – изменение имени на *checkerSoldTogether*, снятие активации (если она была) и присвоение событию *onclick* описанной выше функции *checkElement*.

РЕАЛИЗАЦИЯ СЕРВЕРНОЙ ЧАСТИ

Серверная часть панели администрирования. Серверная часть панели администрирования реализована в классе *Rosavto_SoldViewMod_Model_Observer* (файл *app/code/local/Rosavto/SoldViewMod/Model/Observer.php*). При передаче данных на сервер вызывается *saveProductTabData(Varien_Event_Observer \$observer)*.

```

public function saveProductTabData(
    Varien_Event_Observer $observer) {
    if (!self::$_singletonFlag) {
        self::$_singletonFlag = true;
        $product = $observer->getEvent()-
            >getProduct();
        if ($_REQUEST['soldOperation'] = 'add') {
            $this->newSoldTogether($product);
            return;
        }
        if ($_REQUEST['soldOperation'] = 'remove') {
            $this->deleteSoldTogether($_REQUEST,
                $_REQUEST['y']);
            return;
        }
        if ($_REQUEST['soldOperation'] = 'update') {
            $this->deleteSoldTogether($_REQUEST,
                $_REQUEST['y']);
            $this->newSoldTogether($product);
            return;
        }
    }
}
  
```

Добавление товара в корзину. При добавлении товара в корзину вызывается метод *addToCartComplete(\$observer)* класса *Rosavto_SoldViewMod_Model_Observer_Addtocart* (*app/code/local/Rosavto/SoldViewMod/Model/Observer/Addtocart.php*). При добавлении товара необходимо проверить лежащие в корзине товары на соответствие комплектам *SoldTogether*, причем такую проверку необходимо осуществлять при каждом добавлении (или удалении) товара в корзину. Для этого при каждом вызове метода *addToCartComplete* создается экземпляр класса *CartChecker*, содержащий, в свою очередь, экземпляры классов *CartProducts* и *ComplectCollection*, первый из которых содержит товары, лежащие в корзине, а вто-

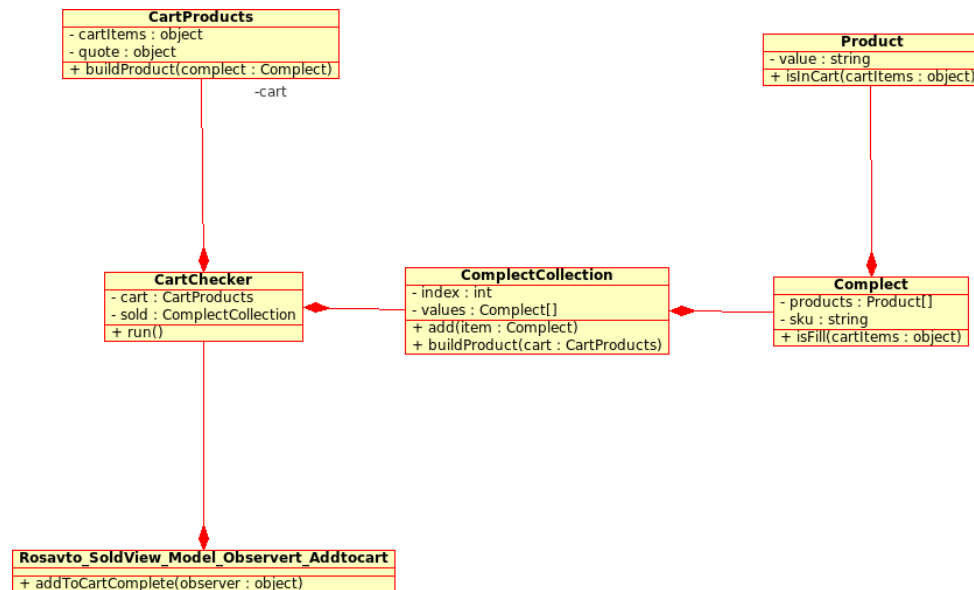


Рис. 6. Диаграмма классов сервера добавления товара в корзину

рой – комплекты *SoldTogether*. Экземпляры обоих этих классов заполняются самостоятельно (используя методы *Magento Mage::getModel* и т. п.). Объект класса *ComplectCollection* содержит набор комплектов – массив экземпляров класса *Complect*, каждый из которых, в свою очередь, содержит SKU главного товара и массив дополнительных товаров – экземпляров класса *Product*.

Для проверки соответствия товаров корзины комплектам *Soldtogether* из метода *run* класса *CartChecker* вызывается метод *buildProduct* класса *ComplectCollection* (*\$this->sold->buildProduct(\$this->cart)*), получающий в качестве параметра экземпляр класса *CartChecker*. В этом методе осуществляется пробег по всем элементам (комплектam *Soldtogether* из базы данных), и для каждого происходит вызов метода *buildProduct* из экземпляра класса *CartChecker* (*foreach (\$this->values as \$id => \$value) \$cart->buildProduct(\$value)*), в котором происходит пробег по всем товарам из корзины и для каждого – сравнение SKU со SKU переданного в метод комплекта. При совпадении SKU вызывается метод *isFull* переданного комплекта, в который передаются товары корзины (*\$complect->isFull(\$this->cartItems)*). В этом методе происходит проверка заполненности комплекта (содержатся ли все товары комплекта в корзине), если комплект заполнен, то происходит изменение его цены:

```
$this->quote->getItemById($item->getId())->setCustomPrice($price);
```

```
$this->quote->getItemById($item->getId())->setOriginalCustomPrice($price);
```

```
$this->quote->save();
```

ВЫВОД

Таким образом, в рамках данной работы был создан компонент *SoldTogether*, позволяющий добавлять в интернет-магазин, созданный на основе движка *Magento*, комплекты товаров, в рамках которых на первый из них предоставляется скидка.

ЛИТЕРАТУРА

1. Magento extends lead in ecommerce survey – february 2011 URL: <http://tomrobertshaw.net/2011/02/magento-extends-lead-in-ecommerce-survey-february-2011/>. Загл. с экрана.
2. Magento – Blog – eBay Agrees to Acquire Magento – eCommerce Software for Growth. URL: <http://www.magentocommerce.com/blog/comments/ebay-agrees-to-acquire-magento/> Загл. с экрана.
3. Что такое SKU? SKU – идентификатор торговой позиции. URL: http://www.marketch.ru/marketing_dictionary/marketing_terms_s/sku/. Загл. с экрана.
4. Бер Бибо, Иегуда Кац. 1.3.5 Расширение jQuery // jQuery. Подробное руководство по продвинутому JavaScript = jQuery in Action. СПб.: Символ-Плюс, 2009. 384 с.
5. Ecommerce Software & Ecommerce Platform Solutions | Magento / URL: <http://www.magentocommerce.com/>. Загл. с экрана.
6. Российская сборка Magento. URL: <http://magento-forum.ru/>. Загл. с экрана.

Поступила в редакцию 28 октября 2013 г.

Kryuchin O.V., Khabirova K.R., Kvashenkin D.O. SOLD-TOGETHER COMPONENT DEVELOPMENT FOR CREATION OF DISCOUNT SERIES OF GOODS WITH SYSTEM MAGENTO USE

In this paper the process creating components for products allowing the discount is described. In the frame of the base for this system Magento is selected. The step by step algorithm realigning module and interface for it is given.

Key words: information system; internet-store; management system.