и хранение предметов.

7. Настройте класс shoppingCart и его зависимости в файле config/web.php:

<?php

use app\cart\storage\SessionStorage;

Yii::$container->setSingleton('app\cart\ShoppingCart');

Yii: :$container->set('app\cart\storage\StorageInterface', function() {

return new SessionStorage(Yii::$app->session, 'primary-cart');

});

$params = require( DIR . '/params.php');

//...

8. Создайте контроллер cart с расширенным конструктором:

<?php

namespace app\controllers;

use app\cart\ShoppingCart;

use app\models\CartAddForm;

use Yii;

use yii\data\ArrayDataProvider;

use yii\filters\VerbFilter;

use yii\web\Controller;

class CartController extends Controller {

private $cart;

public function construct($id, $module, ShoppingCart $cart, $config = [])

{

$this->cart = $cart;

parent::construct($id, $module, $config);

}

public function behaviors()

{

return [

'verbs' => [

'class' => VerbFilter::className(),

'actions' => [

'delete' => [' post' ],

],

],

];

}

public function actionIndex()

{

$dataProvider = new ArrayDataProvider([

'allModels' => $this->cart->getItems(),

]);

return $this->render('index', [

'dataProvider' => $dataProvider,

]);

}

public function actionAdd()

{

$form = new CartAddForm();