

Лабораторная работа №5

Передача данных представлению.

1. Цель работы.

Дальнейшее изучение механизма передачи данных представлению.

Время выполнения работы: 4 часа

2. Общие сведения.

3. Выполнение работы

3.1. Исходные данные

Используйте проект из лабораторной работы №4.

3.2. Задание №1

Выберите любую предметную область. Для одной сущности из выбранной предметной области создайте В папке Entities проекта создайте класс, содержащий следующие свойства:

- ID – уникальный номер;
- Название – короткое название конкретного объекта;
- Описание – дополнительное описание конкретного объекта;
- Категория – свойство для объединения объектов в группы;
- Цена/Вес/Расстояние – выберите любой параметр, который можно в дальнейшем обработать математически, например, просуммировать;
- Изображение – имя файла изображения объекта
- Mime тип изображения

В той же папке создайте класс, описывающий категорию объекта. Отношение должно быть один-ко-многим: одна категория описывает много объектов (см. Рисунок)

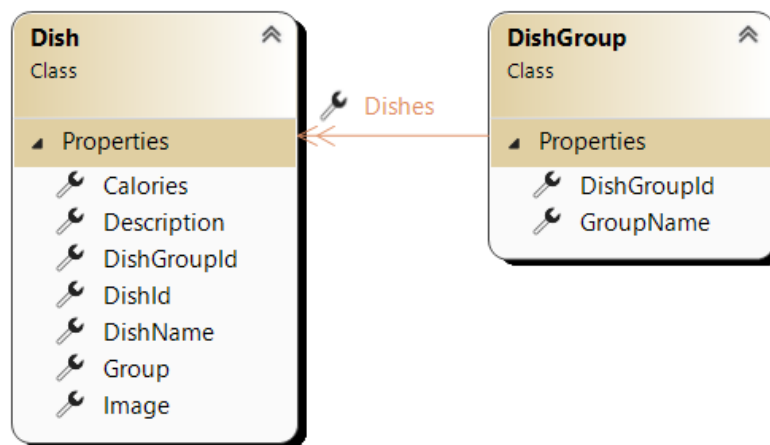


Рисунок. Пример сущностных классов

3.3. Задание №2

Создайте контроллер с именем Product. Данный контроллер будет отвечать за отображение информации об объектах из задания №1.

Примечание:

Вы можете выбрать другое имя для контроллера, чтобы название соответствовало имени класса объекта из задания 1. В этом случае также измените название контроллера в компоненте меню (см. лабораторную работу №3)

Представление Index контроллера должно выводить список объектов.

3.3.1. Рекомендации к заданию №2

В классе контроллера создайте списки объектов и групп объектов типа List<>.

В отдельном методе выполните заполнение списков. Количество элементов в списке объектов должно быть 4-5. Количество элементов в списке групп значения не имеет. Вызовите метод в конструкторе контроллера.

Список объектов используйте в качестве модели представления Index.

Добавьте в папку «wwwroot/images» несколько изображений объектов вашей предметной области. Для удобства отображения желательно, чтобы изображения были одного размера или, хотя бы, имели одинаковое соотношение сторон.

Создание представления с помощью scaffolding:

Add MVC View

View name:

Template: List

Model class: Dish (WebLabsV05.DAL.Entities)

Data context class:

Options:

☐ Create as a partial view

☐ Reference script libraries

☒ Use a layout page:

...

(Leave empty if it is set in a Razor _viewstart file)

Add Cancel

В представлении описана модель:

```
@model IEnumerable<XXX.Entities.Dish>
```

В файл _ViewImports добавьте:

```
@using XXX.Entities
```

Измените модель в представлении Index:

```
@model IEnumerable<Dish>
```

Запустите проект. Выберите пункт меню «Каталог». Убедитесь, что выводится список объектов:

WebLabsV05 Lab 2 Каталог Администрирование 00,0 руб.(0) [User Icon]

Index

[Create New](#)

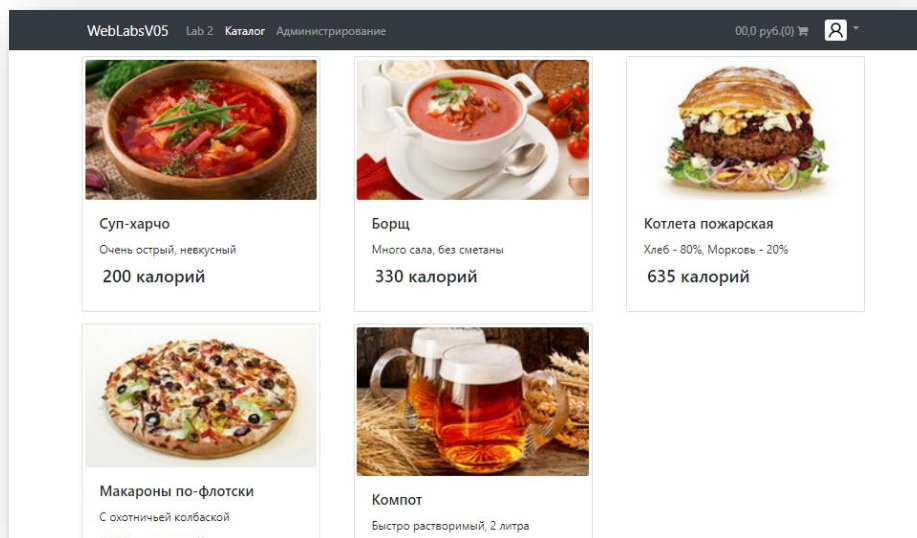
DishId	DishName	Description	Calories	Image	DishGroupId	
1	Суп-харчо	Очень острый, невкусный	200	Суп.jpg	3	Edit Details Delete
2	Борщ	Много сала, без сметаны	330	Борщ.jpg	3	Edit Details Delete
3	Котлета пожарская	Хлеб - 80%, Морковь - 20%	635	Котлета.jpg	4	Edit Details Delete
4	Макароны по-флотски	С охотничьей колбаской	524	Макароны.jpg	4	Edit Details Delete
5	Компот	Быстро растворимый, 2 литра	180	Компот.jpg	5	Edit Details Delete

3.4. Задание №3

Оформите список с помощью класса bootstrap «card» (см. <https://getbootstrap.com/docs/4.3/components/card/#using-grid-markup>):

- информация об одном объекте должна занимать 1/3 контейнера экрана
- в верхней части расположить изображение объекта
- название объекта оформить стилем «card-title»
- текст описания объекта оформить стилем «card-text»
- числовую характеристику оформите стилем «card-subtitle badge»

Пример оформления списка:



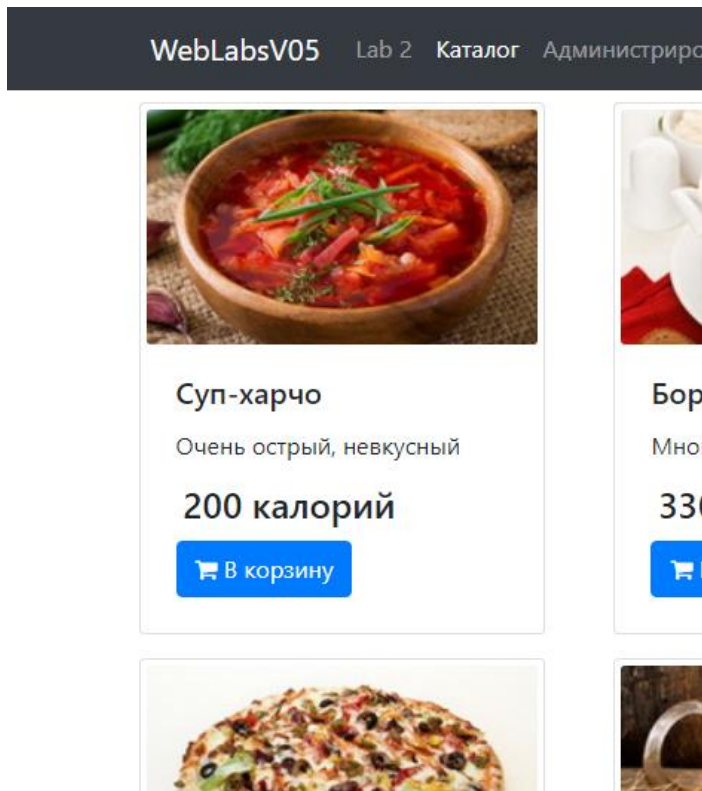
3.5. Задание №4

Для каждого объекта списка добавьте ссылку (тэг <a>) «В корзину».

- ссылка должна адресовать к методу Add контроллера Cart (будут созданы в дальнейшем).
- ссылка должна передавать id добавляемого элемента и адрес текущей страницы для возврата (можно использовать имя returnUrl)
- ссылку оформить стилем «btn btn-primary»
- рядом с текстом «В корзину» должна быть иконка shopping-cart из библиотеки font-awesome (см.

<https://fontawesome.com/icons/shopping-cart?style=solid>)

Пример оформления:



3.5.1. Рекомендации к заданию №4

Адрес текущей страницы можно получить так:

```
@{ // Получение текущего адреса
var request = ViewContext.HttpContext.Request;
var returnUrl = request.Path + request.QueryString.ToUriComponent();
}
```

Пример оформления кнопки добавления в корзину:

```
<p class="mt-2">
    <a asp-action="Add"
        asp-controller="Cart"
        asp-route-id="@item.DishId"
        asp-route-returnUrl="@returnUrl"
        class="btn btn-primary">
        <i class="fa fa-shopping-cart"></i> В корзину
    </a>
</p>
```

3.6. Задание №5

Реализуйте разбиение списка товаров на страницы (по 3 товара на странице). При обращении к нужной странице (например ко 2-й) адрес должен иметь вид /Product/Index?pageNo=2.

3.6.1. Рекомендации к заданию №5

В классе контроллера Product создайте свойство `_pageSize` – количество объектов на странице.

В метод `Index` контроллера Product передайте номер текущей страницы.

Для выборки данных страницы используйте запрос LINQ.

Например:

```
var items = list
    .Where(filter)
    .Skip((current - 1) * itemsPerPage)
    .Take(itemsPerPage)
    .ToList();
```

3.7. Задание №6

В папке Models основного проекта создайте модель представления - класс `ListViewModel`, - имеющий свойства:

- номер текущей страницы;
- общее количество страниц;
- список объектов для отображения на текущей странице.

Класс должен сам делать выборку объектов из общего списка для текущей страницы.

Для создания класса удобно воспользоваться шаблоном проектирования «декоратор»: **унаследуйте** ваш класс от класса `List<T>`, добавив функционал для выборки данных и хранения информации о разбиении на страницы.

Сделайте конструктор `private`, а для создания объекта класса `ListViewModel` опишите **статический** фабричный метод `GetModel`, который принимает исходный список объектов, номер текущей страницы и количество объектов на странице. Метод должен вернуть объект класса `ListViewModel` с

уже сформированным списком объектов и значениями номера текущей страницы и общего количества страниц.

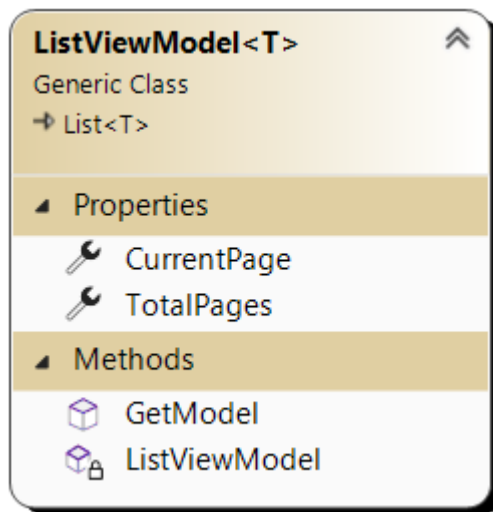


Рисунок. Класс ListViewModel<T>

Вариант сигнатуры метода GetModel:

```
public static ListViewModel<T> GetModel(IQueryable<T> list,  
    int current,  
    int itemsPerPage,  
    Expression<Func<T, bool>> filter)
```

list – исходная коллекция,

current – номер текущей страницы,

itemsPerPage – количество объектов на страницу,

filter – условие отбора элементов коллекции

3.8. Задание №7

Используйте класс ListViewModel в действии Index контроллера Product.

Измените соответственно модель в представлении Index.

3.9. Задание №8

Оформите разметку одного элемента списка в виде частичного представления _ListItemPartial.cshtml.

Пример использования частичного представления для вывода списка:

```
<div class="row">  
    <div class="card-deck col">
```

```

    @foreach (var item in Model)
    {
        <partial name="_ListItemPartial" model="@item" />
    }
</div>
</div>

```

3.10. Задание №9

Добавьте на страницу index возможность выбора категории объектов. При выборе категории в метод Index должен передаваться id выбранной категории.

Выбор категории разместить на странице Index слева от списка объектов.

3.10.1. Рекомендации к заданию №9

Список категорий представляет собой набор ссылок (тэг <a>) на метод Index контроллера Product с передачей параметра «group» - id выбранной группы.

Список категорий можно оформить классом bootstrap «dropdown» (см. <https://getbootstrap.com/docs/4.3/components/dropdowns/>)

Список объектов категорий можно передать через ViewData. Также для выделения текущего выбора в представление нужно передать id текущей выбранной группы или «0», если выбраны все группы.

id выбранной группы можно получить из запроса:

```
HttpContext.Request.Query["group"]
```

Для размещения выбора группы разбейте страницу на две колонки, например, «col-2» и «col-10».

Если категория не выбрана, то в списке категорий должно отображаться «Все».

Пример кода контроллера:

```

public IActionResult Index(int? group, int pageNo=1)
{

```



```

        // Поместить список групп во ViewData
        ViewData["Groups"] = _dishGroups;

        // Получить id текущей группы и поместить в TempData
        ViewData["CurrentGroup"] = group ?? 0;

var model = ListViewModel<Dish>.GetModel(list,
    pageNo,
    _pageSize,
    d=> !group.HasValue || d.DishGroupId == group.Value
    );
    . . .
}

```

Пример получения списка категорий (групп) в представлении:

```

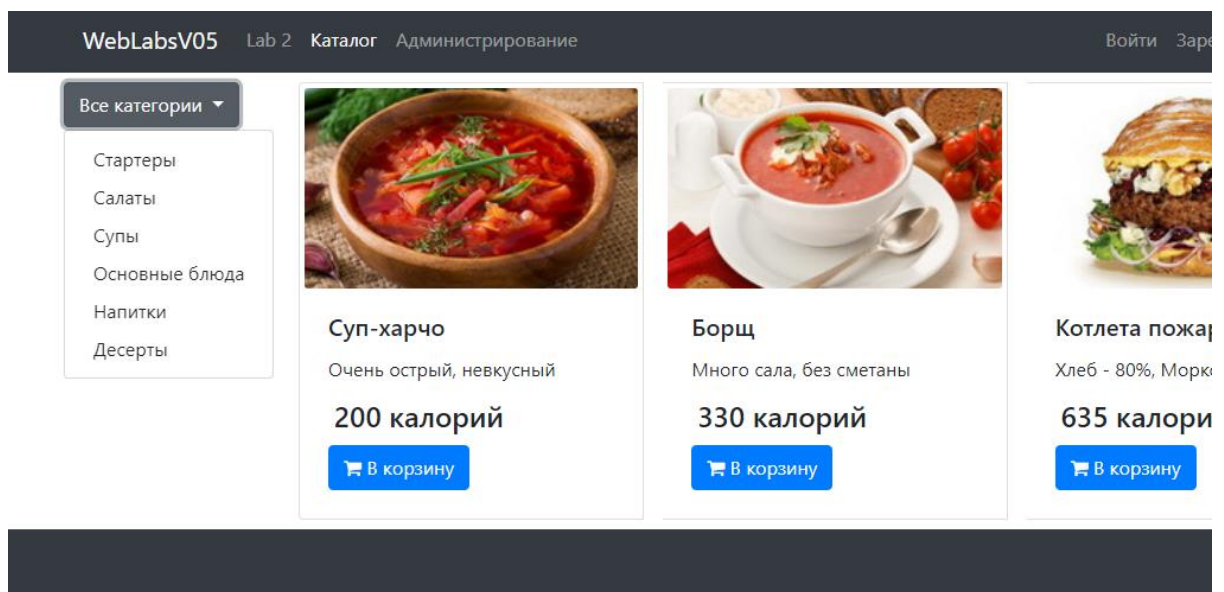
@{
    ViewData["Title"] = "Меню";

    var categories = ViewData["Groups"] as IEnumerable<DishGroup>;
    int currentGroup = (int)ViewData["CurrentGroup"];

    var text = currentGroup != 0
        ? categories
            .FirstOrDefault(g => g.DishGroupId == currentGroup)?
            .GroupName
        : "Все";
}

```

Пример окончательного вида страницы:



4. Контрольные вопросы

Как осуществляется привязка данных запроса к методу контроллера?

Для чего используются объекты ViewData и ViewBag?

Что такое модель представления?

Как передать несколько объектов в модель представления?