



## Сессия 2

### Разработка

#### Список товаров

Необходимо отобразить список товаров (в виде плитки), который должен соответствовать макету, предоставленному в ресурсах к заданию (пример в файле `products_grid.png`). Однако это не означает, что необходимо следовать макету в точности до пикселя - это всего лишь схема расположения элементов, на которую нужно ориентироваться при разработке пользовательского интерфейса.

Каждый товар должен содержать следующую информацию: главное изображение, наименование, стоимость и отметку об активности (является товар на данный момент актуальным или нет).

Если товар не активен, то фон элемента должен быть серого цвета, в остальных случаях - белого.

Около названия товара в круглых скобках должно отображаться количество дополнительных товаров, которые могут идти вместе с этим товаром.

При наведении на карточку товара необходимо реализовать эффект изменения изображений как на примере (`product_image_change_example.mov`). Область с изображением товара делится на равные части с таким же количеством, как и количество изображений у товара. При перемещении мыши по горизонтали изображения должны меняться, а также иконки точек должны показывать текущую позицию в рамках списка изображений.

На странице со списком товаров необходимо добавить возможность поиска товаров по названию и описанию. Поиск должен работать в реальном времени (то есть без необходимости нажатия кнопки “найти”).

На странице со списком товаров необходимо добавить возможность фильтрации списка по производителю. Все производители из базы данных должны быть выведены в выпадающий список для фильтрации. Первым элементом в выпадающем списке должен быть “Все элементы”, при выборе которого настройки фильтра сбрасываются. Фильтрация должна работать в реальном времени (то есть без необходимости нажатия кнопки “найти”).

Функции фильтрации и поиска должны применяться совместно к итоговой выборке.

Добавьте возможность отсортировать список по убыванию и возрастанию стоимости товара. Сортировка должна быть реализована отдельно (не с помощью стандартных функций элементов управления), потому что в дальнейшем планируется усложнение этого функционала путем добавления дополнительных условий.

В нижней части окна необходимо показывать количество выведенных данных и общее количество записей в базе. Например, 230 из 450. В случае если данные в таблицу выводятся после фильтрации или поиска, количество выведенных данных необходимо обновить исходя из размера выборки.





Должна быть реализована возможность удаления выбранного товара. При этом должны соблюдаться следующие условия. Если у товара есть прикрепленные дополнительные товары, то информация по этим прикрепленным товарам может быть удалена вместе с основным. Но если у товара есть информация о его продажах, то удаление товара из базы данных должно быть запрещено.

### Добавление/редактирование товара

Необходимо добавить возможность редактирования данных существующего товара, а также добавление нового товара в новом окне - форме для добавления/редактирования товара.

На форме должны быть предусмотрены следующие поля: идентификатор, наименование товара, стоимость, описание товара, главное изображение, производитель товара.

При открытии формы для редактирования все поля выбранного объекта должны быть подгружены в соответствующие поля из базы данных.

При добавлении нового товара идентификатор должен автоматически генерироваться, а поле для идентификатора отсутствует. В случае если пользователь редактирует товар, то поле с идентификатором доступно только для чтения.

Стоимость товара может включать сотые части, а также не может быть отрицательной.

При изменении данных о товаре у пользователя должна быть возможность сделать товар неактивным или наоборот (поставить/снять соответствующую галочку). Это означает, что в предложениях дополнительных товаров он не будет показан администратору, но в списке товаров он должен быть.

Пользователь может добавить/заменить главное изображение у товара. На экране должна показываться миниатюра выбранной фотографии. Размер фотографии не должен превышать 2 мегабайта.

Для того чтобы администратор случайно не изменял несколько товаров (тем более связанных), предусмотрите невозможность открытия более одного окна редактирования товара.

После редактирования/добавления товара данные в окне списка товаров должны быть обновлены.

### Предложение дополнительных товаров

В окне добавления/редактирования товара необходимо указывать связанные товары, чтобы администратор при работе с клиентом мог предложить товары, которые так или иначе связаны с покупкой. Для примера вспомните, как в одном популярном ресторане для детей, менеджер вам предлагает дополнительный пирожок или картошку фри к вашему заказу? Так вот они это делают, так как у них на экране появляется надпись (например, “предложить клиенту пирожок”) в зависимости от списка товаров, которые вы заказали.

При открытии формы для редактирования ранее прикрепленные товары должны подгружаться из базы данных.

Для каждого прикрепленного товара должна выводиться миниатюра (главное изображение). При наведении на миниатюру в подсказке должны отображаться название и стоимость данного товара. По клику можно перейти в окно добавления/редактирования товара со всей подробной информацией.





Необходимо предусмотреть функционал для добавления определенных дополнительных товаров к выбранному (причем не только один, но и несколько). Имейте в виду, что нельзя добавить товар в качестве дополнительного к самому себе (не пытайтесь нарушить покой вселенной), а также должен быть реализован запрет на прикрепление неактивных товаров.

Предусмотрите также возможность удаления прикрепленных товаров из списка.

Прикрепленные товары должны сохраняться в базе данных для каждого выбранного товара.

### История продаж товаров

В окне списка товаров необходимо предусмотреть возможность перейти на окно для получения списка всех продаж этого товара.

На данном окне должна быть возможность посмотреть информацию о конкретном товаре, дате и времени продажи, а также количестве проданного товара.

Записи должны быть по умолчанию отсортированы по дате и времени по убыванию.

Необходимо предусмотреть фильтр по товарам. Данные для фильтрации в выпадающий список должны быть подгружены из БД. По умолчанию в выпадающем списке должен быть выбран товар, переход по которому был осуществлен на данное окно.

## Разработка библиотеки классов

### Общие требования

В связи со стремительным развитием нашей системы было решено вынести некоторый важный функционал за рамки основного проекта и сделать библиотеку классов, которую мы сможем подключать к любому нашему проекту в случае расширения. Данная библиотека будет подключаться к основному проекту и должна быть представлена в виде .dll/.jar файла или папки с файлом .ру.

Чтобы система правильно интегрировалась вам необходимо обязательно следовать правилам именования библиотек, классов и методов в них. В случае ошибок в рамках именования ваша работа не может быть проверена и ваш результат не будет зачтен. Классы и методы должны содержать модификатор public (если это реализуемо в рамках платформы), чтобы внешние приложения могли получить к ним доступ.

В качестве названия для библиотеки необходимо использовать: CompanyCoreLib. Вам необходимо загрузить исходный код проекта с библиотекой в отдельный репозиторий с названием, совпадающим с названием приложения.

### Класс аналитики

Вам необходимо создать класс с названием Analytics, который будет позволять проводить аналитику различных процессов в рамках компании.

Реализуйте метод, который принимает в себя список объектов даты и времени по совершенным покупкам/заказам в рамках нашей компании, а возвращает список дат (без времени), отсортированный в порядке уменьшения частоты заказов. Это необходимо, чтобы наша компания могла прогнозировать наиболее высокий спрос на следующий год для обеспечения более качественного оказания услуг.





Возвращаемые данные должны содержать только даты для первого числа каждого месяца и 00:00 минут. Например, вам поступили следующие данные: 2019-12-12 14:43, 2019-12-01 15:05, 2019-11-04 09:01, а, значит, самый популярный месяц - декабрь. Вам необходимо вернуть следующие данные: 2019-12-01 00:00, 2019-11-01 00:00. В случае совпадения характеристик популярности сперва нужно вывести более ранние месяцы.

Прогноз строится на основе предыдущего года. Так что данные Вам будут выдаваться строго за предыдущий год.

Спецификация метода представлена в отдельном файле в ресурсах.

### Разработка модульных тестов (Unit-tests)

Для выполнения процедуры тестирования созданного вами метода библиотеки CompanyCoreLib, возвращающего упорядоченный список популярных месяцев, вам необходимо создать отдельный проект модульных тестов.

В рамках проекта разработайте тесты, максимально полно покрывающие функционал метода. Ничего страшного, если ваш метод работает не совсем идеально и тесты могут быть не пройдены в связи с этим - в данном модуле это не так важно.

Обратите внимание, что имена тестов должны отражать их суть, т.е. вместо TestMethod1() тест следует назвать, например, PopularMonths\_NullList() для тестирования случая передачи пустой коллекции дат.

Необходимо разработать модульные тесты, которые на основании исходных данных можно условно разделить на 2 группы следующим образом: 10 методов низкой сложности и 5 методов высокой сложности.

