



## Сессия 2

### Разработка desktop-приложений

#### Список продукции

Необходимо реализовать вывод продукции, которая хранится в базе данных, согласно предоставленному макету. При отсутствии изображения необходимо вывести картинку-заглушку из ресурсов (picture.png).

В случае если в базе более 20 продуктов, то вывод должен осуществляться постранично (по 20 продуктов на страницу). Для удобства навигации по страницам необходимо вывести список их номеров (как на макете) с возможностью перехода к выбранной странице, а также предусмотреть переходы к предыдущей и следующей страницам.

Стоимость продукта должна быть рассчитана исходя из используемых материалов.

Пользователь должен иметь возможность отсортировать продукцию (по возрастанию и убыванию) по следующим параметрам: наименование, номер производственного цеха и минимальная стоимость для агента. Выбор сортировки должен быть реализован с помощью выпадающего списка.

Кроме этого, пользователь должен иметь возможность отфильтровать данные по типу продукта. Все типы из базы данных должны быть выведены в выпадающий список для фильтрации. Первым элементом в выпадающем списке должен быть “Все типы”, при выборе которого настройки фильтра сбрасываются.

Пользователь должен иметь возможность искать конкретную продукцию, используя поисковую строку. Поиск должен осуществляться по наименованию и описанию продукта.

Поиск, сортировка и фильтрация должны происходить в реальном времени, без необходимости нажатия кнопки “найти”/”отфильтровать” и т.п. Фильтрация и поиск должны применяться совместно. Параметры сортировки, выбранные ранее пользователем, должны сохраняться и во время фильтрации с поиском.

В списке продукции необходимо подсвечивать светло-красным цветом те продукты, которые не продавались агентами в последний месяц.

Необходимо добавить возможность изменения минимальной стоимости продукции для агента сразу для нескольких выбранных продуктов. Для этой цели реализуйте возможность выделения сразу нескольких элементов в списке продукции, после чего должна появиться кнопка “Изменить стоимость на ...”. При нажатии на кнопку необходимо отобразить модальное окно с возможностью ввода числового значения, на которое и будет увеличена стоимость выбранных продуктов. По умолчанию в поле должно быть введено среднее значение цены на продукцию для агента. После нажатия кнопки “Изменить” стоимость выделенных продуктов должна быть изменена в базе данных, а также обновлена в интерфейсе.

#### Добавление/редактирование продукции

Необходимо добавить возможность редактирования данных существующей продукции, а также добавление новой продукции в новом окне - форме для добавления/редактирования продукции.





Переходы на данное окно должны быть реализованы из главной формы списка: для редактирования - при нажатии на конкретный элемент, для добавления - при нажатии кнопки “Добавить продукцию”.

На форме должны быть предусмотрены следующие поля: артикул, наименование, тип продукта (выпадающий список), изображение, количество человек для производства, номер производственного цеха, минимальная стоимость для агента и подробное описание (с возможностью многострочного ввода).

Также необходимо реализовать вывод списка материалов, используемых при производстве продукции, с указанием количества. В список можно добавлять новые позиции и удалять существующие. При добавлении материалы должны выбираться из выпадающего списка с возможностью поиска по наименованию.

При открытии формы для редактирования все поля выбранного объекта должны быть подгружены в соответствующие поля из базы данных, а таблица заполнена актуальными значениями.

Стоимость продукции может включать сотые части, а также не может быть отрицательной. Система должна проверять существование продукта с введенным артикулом и не давать использовать один артикул для нескольких продуктов.

Пользователь может добавить/заменить изображение у продукции.

Для того чтобы администратор случайно не изменял несколько продуктов, предусмотрите невозможность открытия более одного окна редактирования.

В окне редактирования продукта должна присутствовать кнопка “Удалить”, которая удаляет продукт из базы данных. При этом должны соблюдаться следующие условия. Если у продукта есть информация о материалах, используемых при его производстве, или история изменения цен, то эта информация должна быть удалена вместе с продуктом. Но если у продукта есть информация о его продажах агентами, то удаление продукта из базы данных должно быть запрещено. После удаления продукта система должна сразу вернуть пользователя обратно к списку продукции.

После редактирования/добавления/удаления продукции данные в окне списка продукции должны быть обновлены.

## Разработка библиотеки классов

### Общие требования

Для того чтобы в производстве могли быстро и одинаково рассчитывать количество необходимого сырья для производства той или иной продукции, необходимо разработать библиотеку классов.

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: `WSUniversalLib`. Вам необходимо загрузить исходный код проекта с библиотекой в отдельный репозиторий с названием, совпадающим с названием проекта.

### Класс расчета материалов

Метод должен рассчитывать целое количество сырья, необходимого для производства определенного количества (`count`) продукции, учитывая возможный брак материалов. Для упрощения расчетов будем считать всю продукцию прямоугольного размера с известными значениями ширины (`width`) и длины (`length`).

Количество необходимого качественного сырья на одну единицу продукции рассчитывается как площадь продукции, умноженная на коэффициент типа продукции.

Коэффициенты типа продукции (`product_type`):

Тип продукции 1 - 1.1,

Тип продукции 2 - 2.5,

Тип продукции 3 - 8.43.

При этом нужно учитывать процент брака материала в зависимости от его типа (`material_type`):

Тип материала 1 - 0.3%,

Тип материала 2 - 0.12%.

При этом если в качестве параметров метода будут приходить несуществующие типы продукции/материалов или другие неподходящие данные, то метод должен вернуть -1.

Например, необходимо изготовить 15 единиц продукции 3 типа шириной 20 и длиной 45 из материала 1 типа. Количество качественного сырья (без учета брака) будет равно 113 805. Однако с учетом возможного брака материалов общее необходимое количество сырья должно быть увеличено до 114 147,442. Округлив полученное значение до ближайшего большего целого, получим 114 148 единиц необходимого сырья.

Спецификация метода представлена в отдельном файле в ресурсах.

### Разработка модульных тестов (Unit-tests)

Для выполнения процедуры тестирования созданного вами метода библиотеки `WSUniversalLib`, возвращающего целое количество сырья для производства, 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, `GetQuantityForProduct_NonExistentProductType()` для тестирования случая передачи несуществующего типа продукции.

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.

