

Сессия 2

Модуль 4: Разработка desktop-приложений

Формирование заказа (терминал)

Реализуйте возможность формирования заказа из окна просмотра списка товаров для авторизованных пользователей и для гостя ООО «Обувь». Этот модуль информационной системы предназначен для терминала в торговых центрах города.

При просмотре списка товаров пользователь может выделить товар, по нажатию правой клавиши мыши вызвать контекстное меню с командой «Добавить к заказу». При нажатии на команду происходит формирование заказа и добавление в заказ выбранной позиции в количестве 1 единицы.

Если хотя бы один товар добавлен к заказу в интерфейсе пользователя появляется кнопка для просмотра заказа. Если ни один товар не добавлен к заказу кнопка просмотра заказа не отображается в интерфейсе. При просмотре товаров все добавления производятся в один заказ.

Просмотр заказа должен быть реализован в отдельном модальном окне. При просмотре заказа отображается вся информация от товаре, в том числе изображение. Реализована возможность удалить товар, указав количество - 0 или нажатием на элемент интерфейса.

При формировании заказа:

- в реальном времени ведется подсчет суммы заказа, подсчет скидки заказа, если в нем есть товары со скидкой;
- заказу автоматически присваивается номер (+1 к существующему в БД);
- информация о заказе хранится в БД;
- при создании статус заказа новый;
- реализована возможность выбрать пункт выдачи.

Для получения заказа формируется талон с возможностью сохранения в pdf.

Талон содержит: дата заказа, номер заказа, состав заказа, сумму заказа, сумму скидки, пункт выдачи и код получения. Код для получения должен быть выделен размером шрифта и типом начертания – полужирный. Код состоит из 3 цифр, которые генерируются случайным образом при создании заказа. Если все заказы в наличии на складе в количестве более 3 позиций, то в талоне указывается срок доставки – 3 дня. Если менее 3 позиций или отсутствует, то 6 дней.

Если заказ формирует авторизованный клиент в окне просмотра заказа отображается ФИО клиента.

Работа с заказами

Реализуйте интерфейс для работы с заказами, который доступен только менеджеру или администратору.

Должна быть реализована возможность просмотра всех данных по заказу: номер заказа, состав заказа, дата заказа, дата доставки, общая сумма заказа, общий размер скидки заказа, ФИО клиента (если заказ был сформирован авторизованным клиентом).

Если все товары в заказе есть на складе в наличии более 3 позиций, то строка с заказом должна быть выделена цветом #20b2aa.

Если хотя бы одного товара нет на складе, то строка должна быть выделена цветом #ff8c00.

Реализуйте возможность изменения статуса заказа и даты доставки, сортировку заказа по убыванию и возрастанию стоимости. Пользователь должен иметь возможность отфильтровать данные о заказах по размеру суммарной скидки в диапазонах: 0-10%, 11-14%, 15% и более.

Разработка библиотеки

Разработайте библиотеку (отдельным проектом), которая позволит вернуть список свободных временных интервалов (заданного размера) в графике сотрудника ООО «Обувь» для формирования оптимального графика работы сотрудников.

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.

Вход:

- список занятых промежутков времени (в двух массивах: `startTimes` - начало, `durations` - длительность),
- минимальное необходимое время для работы менеджера (`consultationTime`),
- рабочий день сотрудника (начало - `beginWorkingTime` и завершение - `endWorkingTime`)

Выход:

- список подходящих свободных временных промежутков (в массив строк формата `HH:mm-HH:mm`)

Требования к именованиям и форматам:

	C#	Java	Python
Библиотека классов	SF2022User{NN}Lib.dll	SF2022User{NN}Lib.jar	SF2022User{NN}Lib
Название класса	Calculations	Calculations	Calculations
Название метода	AvailablePeriods()	availablePeriods()	available periods()
Входящие обязательные параметры	TimeSpan[] startTimes, int[] durations, TimeSpan beginWorkingTime, TimeSpan endWorkingTime, int consultationTime	LocalTime[] startTimes, int[] durations, LocalTime beginWorkingTime, LocalTime endWorkingTime, int consultationTime	time[] start times, int[] durations, time begin working time, time end working time, int consultation time
Возвращаемые параметры	string[]	string[]	string[]

Пример:

Вход	Выход
startTime duration 10:00 60 11:00 30 15:00 10 15:30 10 16:50 40 Working Times 08:00-18:00 Consultation Time 30	08:00-08:30 08:30-09:00 09:00-09:30 09:30-10:00 11:30-12:00 12:00-12:30 12:30-13:00 13:00-13:30 13:30-14:00 14:00-14:30 14:30-15:00 15:40-16:10 16:10-16:40 17:30-18:00

Модуль 7: Тестирование программных решений

Тестирование

Модульные тесты

Реализуйте 10 unit-тестов на основе технологии TDD для библиотеки, функционал которой описан ранее. Важно, чтобы тестовые данные предусматривали различные ситуации. Например, недостаточное время в промежутках между ранее созданными интервалами, либо в начале рабочего дня, либо в конце рабочего дня; различная длительность работы и т.д.

Тестовая документация

Для выполнения процедуры тестирования формирования заказа администратором Вам нужно описать пять сценариев. Добавление может быть выполнено успешно, а может быть отклонено согласно требованиям предметной области. Необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма. Для описания тестовых сценариев в ресурсах предоставлен шаблон `testing-template.docx`.

Модуль 6: Создание инсталляторов

Разработка инсталлятора

Разработайте установочный файл для desktop-приложения под ОС Windows.

Модуль 8: Документирование программных решений

Документация API

На данный момент разрабатываемая Вами информационная система взаимодействует напрямую с базой данных. Руководство ООО «Обувь» приняло решение об изменениях архитектуры информационной системы. В дальнейшем планируется организовать работу ИС ООО «Обувь» с помощью API. Разработайте документацию по API, которая будет полезна для реализации взаимодействия с базой данных.

Убедитесь, что ваша документация API содержит следующую информацию:

- путь для вызова каждой конечной точки;
- какие методы HTTP можно использовать с каждой конечной точкой;
- описание и назначение каждого метода;
- поля данных запроса и куда они идут, например путь, строка запроса или тело;
- объяснение того, какие данные запроса требуются, а какие являются необязательными.

Руководство пользователя

Вам необходимо разработать руководство пользователя для вашего настольного приложения, которое описывает последовательность действий для выполнения всех функций вашей системы.

При подготовке документации старайтесь использовать живые примеры и скриншоты вашей системы для более наглядного пояснения шагов работы с различным функционалом.

Обратите внимание на оформление документа: оформите титульный лист, используйте автоматическую нумерацию страниц, разделите руководство на подразделы и сформируйте оглавление, используйте ссылки на рисунки, нумерованные и маркированные списки для описания шагов и т. д.

Сохраните итоговый документ с руководством пользователя в формате Word, используя в качестве названия следующий шаблон: Руководство пользователя XX, где XX - номер вашего рабочего места.