**ЗАДАНИЕ ДЕМОНСТРАЦИОННОГО ЭКЗАМЕНА**

**ОПИСАНИЕ ЗАДАНИЯ**

Описание модуля 2: «Разработка программного обеспечения»

Вы можете выбрать любую среду разработки и язык программирования из доступных, но должны следить это обдуманно, придерживаться при их использовании профессиональных стандартов.

Обязательными требованиями является обеспечение ограниченного доступа к продукту, возможности ввода и хранения данных.

Программный продукт должен быть готовым решением. Пользователи не должны устанавливать или настраивать СУБД, вручную переносить хранящиеся данные и т.п. Учтите: компьютер при проверке будет сконфигурирован точно также, как и ваш перед началом знакомства с ним.

При выполнении модуля 2 ставятся следующие цели:

1. Разработка программного продукта.

При выполнении данного модуля 2 ставятся следующие задачи:

1. Выбрать технологический стек для реализации программного продукта.
2. Разработать объекты баз данных, заполнить таблицы текстовыми данными.
3. Разработать программный продукт на основании предоставленных функциональных требований

Описание модуля 3: «Стандарты разработки программного обеспечения»

Модуль отражает общий профессионализм решения: обратная связь системы с пользователем, стабильная работа всех разработанных программ, стиль кода на протяжении разработки всей системы, организация файловой структуры проекта, соблюдение культуры кодирования, комментарии к коду, умение работать с системой контроля версий.

При выполнении модуля 3 ставятся следующие цели:

1. Разработка кода программного продукта в соответствии с отраслевыми стандартами.

При выполнении данного модуля 3 ставятся следующие задачи:

1. Реализовать обратную связь системы с пользователем.
2. Соблюдать культуру кодирования.
3. Результаты работы предоставить в системе контроля версий.

**Название приложения**

Используйте соответствующие названия для ваших приложений и файлов. Так, например, наименование настольного приложения должно обязательно включать название компании-заказчика.

**Файловая структура**

Файловая структура проекта должна отражать логику, заложенную в приложение. Например, все формы содержаться в одной директории, пользовательские визуальные компоненты – в другой, классы сущностей – в третьей.

**Структура проекта**

Каждая сущность должна быть представлена в программе как минимум одним отдельным классом. Классы должны быть небольшими, понятными и выполнять одну единственную функцию (Single responsibility principle).

Для работы с разными сущностями используйте разные формы, где это уместно.

**Макет и технические характеристики**

Все компоненты системы должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

* разметка и дизайн (предпочтение отдаётся масштабируемой компоновке;
* должно присутствовать ограничение на минимальный размер окна;
* должна присутствовать возможность изменения размеров окна, где это необходимо;
* увеличение размеров окна должно увеличивать размер контентной части, например, таблицы с данными из БД);
* группировка элементов (в логические категории);
* использование соответствующих элементов управления (например, выпадающих списков для отображения подстановочных значений из БД);
* расположение и выравнивание элементов (метки, поля для ввода и т.д.);
* последовательный переход фокуса по элементам интерфейса (по нажатию клавиши Tab);
* общая компоновка логична, понятна и проста в использовании;
* последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (в том числе обратно, например с помощью кнопки «Назад»);
* соответствующий заголовок на каждом окне приложения (не должно быть значений по умолчанию типа Main Window, Form1 и тп).

**Обратная связь с пользователем**

Уведомляйте пользователя о совершаемых им ошибках или о запрещенных рамках задания действиях, запрашивайте подтверждение перед удалением, предупреждайте о неотвратимых операциях, информируйте об отсутствии результатов поиска и т.п. Окна сообщений соответствующих типов (например, ошибка, предупреждение, информация) должны отображаться с соответствующим заголовком и пиктограммой. Текст сообщения должен быть полезным и информативным, содержать полную информацию о совершенных ошибках пользователя и порядок действий для их исправления. Также можно использовать визуальные подсказки для пользователя при вводе данных.

**Обработка ошибок**

Не позволяйте пользователю вводить некорректные значения в текстовые поля сущностей. Например, в случае несоответствия типа данных или размера поля введенному значению. Оповестите пользователя о совершенной им ошибке.

При возникновении непредвиденной ошибки приложение не должно аварийно завершать работу.

**Оформление кода**

Идентификаторы переменных, методов и классов должны отражать суть и/или цель их использования, в том числе и наименования этих элементов управления (например, не должно быть значений по умолчанию типа Form1.Button3).

Идентификаторы должны соответствовать соглашению об именовании (Code Convention) и стилю CamelCase (для С# и Java) и snake\_case (для Python).

Допустимо использование не более одной команды в строке.

**Комментарии**

Используйте комментарии для пояснения неочевидных фрагментов кода. Запрещено комментирование кода. Хороший код воспринимается как обычный текст. Не используйте комментарии для пояснения очевидных действий. Комментарии должны присутствовать только в местах, которые требуют дополнительного пояснения.

Используйте тип комментариев, который в дальнейшем позволит сгенерировать XML-документацию, с соответствующими тегами (например, param, return(s), summary и др.)

**Требования к оформлению письменных материалов**

Все письменные материалы в ходе выполнения экзаменационного задания должны быть оформлены аккуратно, руководствуясь ГОСТ для оформления документов при необходимости.

**Представление результатов работы**

Все практические результаты должны быть переданы заказчику путем загрузки файлов на предоставленный вам репозиторий системы контроля версий git. Практическими результатами являются:

* исходный код приложения (в виде коммита текущей версии проекта, не архивом),
* исполняемые файлы,
* прочие текстовые файлы.

Результаты работы каждой сессии должны быть загружены в отдельный репозиторий с названием «Сессия Х» (Х-номер сессии).

Для оценки работы будет учитываться только содержимое репозитория. При оценке рассматриваются заметки только в электронном виде (readme.md). Рукописные примечания не будут использоваться для оценки.

**Необходимые приложения**

Необходимые приложения смотреть в папке «КОД 1.1 Приложения к вариантам»

Приложение 1. Описание предметной области.pdf

Приложение 2. Сессия 1.pdf

Описание предметной области

ООО «Книжный клуб» - магазин по продаже книг и канцелярских товаров.

В рамках выполнения задания демонстрационного экзамена необходимо разработать основные модули информационной системы для ООО «Книжный клуб»:

* неавторизованный клиент и авторизованный клиент может просматривать товары и формировать заказ;
* менеджер может просматривать товары, формировать и редактировать заказы;
* администратор может добавлять/редактировать/удалять товары, просматривать и редактировать заказы.

Кроме того, разрабатываемая Вами информационная система предполагает установку на терминалах при входе в торговые центры города. На терминале клиент (авторизованный и неавторизованный) может просмотреть товары, сформировать заказ и выбрать удобный для него пункт выдачи.

Описание предметной области

ООО «Спорт» - магазин по продаже товаров для спорта.

В рамках выполнения задания демонстрационного экзамена необходимо разработать основные модули информационной системы для ООО «Спорт»:

* неавторизованный клиент и авторизованный клиент может просматривать товары и формировать заказ;
* менеджер может просматривать товары, формировать и редактировать заказы;
* администратор может добавлять/редактировать/удалять товары, просматривать и редактировать заказы.

Кроме того, разрабатываемая Вами информационная система предполагает установку на терминалах при входе в торговые центры города. На терминале клиент (авторизованный и неавторизованный) может просмотреть товары, сформировать заказ и выбрать удобный для него пункт выдачи.

Описание предметной области

ООО «Спорт» - магазин по продаже товаров для спорта.

В рамках выполнения задания демонстрационного экзамена необходимо разработать основные модули информационной системы для ООО «Спорт»:

* неавторизованный клиент и авторизованный клиент может просматривать товары и формировать заказ;
* менеджер может просматривать товары, формировать и редактировать заказы;
* администратор может добавлять/редактировать/удалять товары, просматривать и редактировать заказы.

Кроме того, разрабатываемая Вами информационная система предполагает установку на терминалах при входе в торговые центры города. На терминале клиент (авторизованный и неавторизованный) может просмотреть товары, сформировать заказ и выбрать удобный для него пункт выдачи.

Сессия 1

**База данных**

На основе описания предметной области и задания демонстрационного экзамена Вам необходимо разработать систему хранения данных. При разработке объектов базы данных обратите внимание па согласованную осмысленную схему именования, создайте необходимые первичные и внешние ключи, определите ограничения внешних ключей, отражающие характер предметной области.

Заполните все таблицы тестовыми данными. Для изображений используйте

системные изображения ОС.

**Разработка** **desktop-приложений**

**Список товаров**

Очень важно постоянно получать актуальную информацию об остатках товаров па складах, поэтому необходимо реализовать вывод товаров, которые хранятся в базе данных: фото, наименование товара, описание товара, производитель, цена, размер скидки (не обязательное поле).

Вывод должен осуществляться списком.

**Формирование заказа**

Реализуйте возможность формирования заказа из окна просмотра списка товаров для гостя. 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 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 о 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 сеть товары со скидкой;
* заказу автоматически присваивается помер (+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 отсутствуют, то 6 дней.