**Задание для подготовки**

*Описание предметной области*

Вашей задачей является разработка информационной системы для Отдела технического контроля.

ОТК отвечает за некачественную продукцию, за ее возврат на производство.

Вам необходимо разработать информационную систему для ОТК, с которой будут работать сотрудники отдела, а также клиенты – сторонние предприятия для контрольных проверок или частные лица.

|  |
| --- |
| Модуль 1: Разработка программных модулей программного обеспечения для компьютерных систем |
| Задание модуля 1:  **Разработка desktop-приложений**  **Окно входа**  Разработайте окно входа, в котором пользователю предлагается ввести свой логин и пароль. Только после удачной авторизации пользователь получает доступ к остальным модулям системы.  При вводе пароль должен быть скрыт маской ввода, но также должна быть реализована возможность просмотра введенного пароля.  Реализуйте необходимые интерфейсы для всех пользователей системы. После входа в любую учетную запись должна быть реализована возможность выхода на главный экран – окно входа.  **Формирование заказа**  Прием материала и формирование заявки выполняется менеджером по работе с клиентами, лаборантом или контролером ОТК.  Реализуйте интерфейс для формирования заказа:   1. поле ввода номера заказа; 2. поле ввода ФИО заказчика; 3. поле выбора услуги; 4. поле стоимость - считается автоматически, но можно указать размер скидки, стоимость будет пересчитана.   *Требования к разработке*  *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 содержатся 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 отдается 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 базы данных); * 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 MainWindow, Form1 и тп).   *Обратная связь с пользователем*  Уведомляйте пользователя о совершаемых им ошибках или о запрещенных в 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 элементов управления (например, не должно быть значений по умолчанию типа Form1, button3).  Идентификаторы должны соответствовать соглашению об именовании (Code Convention) и стилю CamelCase (для C# 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.) |
| Модуль 2: Разработка и администрирование баз данных |
| Задание модуля 2:  **Проектирование базы данных (ERD)**  Изучите описание предметной области. На основе описания предметной области и задания экзамена Вам необходимо спроектировать ER-диаграмму для информационной системы. Обязательна 3 нормальная форма с обеспечением ссылочной целостности.  **База данных**  Для работы приложения вам необходимо создать базу данных. Создайте базу данных, используя предпочтительную платформу, на сервере баз данных, который вам предоставлен.  Создайте таблицы основных сущностей, атрибуты, отношения и необходимые ограничения. В любом случае созданные таблицы должны содержать начальные тестовые данные. Заполните базу данных тестовыми данными.  **Data Dictionary**  Для диаграммы ER разработайте словарь данных – набор информации, описывающий, какой тип данных хранится в базе данных, их формат, структуру и способы использования данных. Обратите внимание на соответствие вашей диаграммы и словаря данных. Используйте подходящие типы данных, ограничения и форматы. Отразите в документе ограничения, определенные в таблицах, включая первичные ключи, отношения внешнего ключа с другими таблицами и ненулевые ограничения. Не забудьте сделать необходимые пояснения и комментарии к неоднозначным полям.  Обеспечьте хранение в базе данных:   * услуги (наименование, стоимость); * данные клиентов; * заказ; * данные других сотрудников ОТК (логин, пароль).   При организации хранения данных вам необходимо учесть запрет на полное удаление данных, реализовав возможность отправки данных в архив. |
| Модуль 3: Участие в интеграции программных модулей |
| Задание модуля 3:  **Диаграмма прецедентов**  Для согласования процесса разработки с заказчиком Вам необходимо ознакомиться с описанием предметной области и заданием экзамена, сделать диаграмму прецедентов (Use Case) для основных пользователей системы. Сохраните файл с диаграммой в форматах .vsdx и .pdf.  **Тестовая документация**  Для выполнения процедуры тестирования прецедента «Авторизация» Вам нужно описать два сценария. Добавление заказа может быть выполнено успешно, а может быть отклонено согласно требованиям предметной области. Необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма.  **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. |