

САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ ГОРОДСКОГО ХОЗЯЙСТВА»

ЗАДАНИЕ

Группа ИП-21-3 четвертого курса очной формы обучения

Ф.И.О.

на учебную практику УП.02

по ПМ.02 Осуществление интеграции программных модулей

с «13» января 2025 г. по «15» февраля 2025 г.

Сессия 1:

Модуль 1. Разработка, администрирование и защита баз данных

На основе описания предметной области необходимо создать базу данных в выбранной СУБД для разрабатываемой системы. Обязательна 3 нормальная форма с обеспечением ссылочной целостности. При разработке базы данных обратить внимание на согласованную схему именования, создать необходимые первичные и внешние ключи. На данном этапе нет необходимости воспроизводить все сущности предметной области, достаточно создать таблицы, поля с подходящими типами данных и связи, непосредственно относящиеся к разрабатываемой подсистеме и ее функционалу.

Получить ER-диаграмму средствами СУБД: ER-диаграмма должна быть представлена в формате PDF и содержать таблицы, связи между ними, атрибуты и ключи (типами данных на данном этапе можно пренебречь).

Заказчик системы предоставил файлы с данными (с пометкой *import* в ресурсах) для переноса в новую систему. Необходимо подготовить данные файлов для импорта и загрузить в разработанную базу данных.

Сохранить полученные результаты: создать скрипт БД с данными.

Модуль 2. Разработка модулей программного обеспечения для компьютерных систем

Сформировать алгоритм подсистемы для работы с объектами системы.

В рамках выполнения задания демонстрационного экзамена необходимо разработать основные модули информационной системы заданной предметной

области. В предметной области должны быть определены следующие основные составляющие:

1. **Клиенты** (авторизованный): тип, наименование, адрес, ИНН, ФИО директора компании, контактные данные (телефон, email), рейтинг, история заказов. Рейтинг клиента должен быть целым неотрицательным числом. Клиенты покупают товары/материалы со скидками, которые зависят от общего объема их покупок в магазине. Скидки мотивируют клиентов к большему объему покупок. Взаиморасчеты с клиентами осуществляются в рублях.
Клиент (неавторизованный и авторизованный) может просматривать товары и формировать заказ;
Клиент может отменить заказ до момента внесения предоплаты.
2. **Администраторы**: добавление/редактирование/удаление товара.
3. **Менеджеры**:
 - просмотр товаров/материалов,
 - просмотр, редактирование и подтверждение заказа клиента;
 - поиск и регистрация клиентов, решение об изменении рейтинга клиента, прием заказов от клиентов.
4. **Заказ**: это информация, предоставленная клиентом о покупке товаров/материалов, которая требует подтверждения. Клиент оставляет новую заявку. Заказ может содержать данные о товаре (артикул, количество), дата заказа, дата доставки, пункт выдачи, ФИО клиента, код для получения, статус заказа.
Если предоплата не поступает в течение трех дней, то заказ отменяется автоматически, и статус заказа изменяется на «отменен». После поступления предоплаты, проверяется наличие нужного количества товара, и заказ выполняется.

Разработать алгоритм функции расчета индивидуальной скидки для авторизованного клиента системы.

Алгоритмы реализовать в виде кода программного продукта средствами любой среды разработки и языка программирования из доступных.

Создание настольного приложения: различных окон, таблиц, списков, форм для заполнения, работа с базой данных и пр.

- 1) Выбрать технологический стек для реализации программного продукта
- 2) **Главное окно приложения – список объектов**, из которого можно попасть на окно авторизации и содержащее другие элементы для перехода на интерфейсы, подсчет информации
- 3) Установить иконку приложения, если это реализуемо в рамках платформы, и логотип компании на главной форме, из ресурсов
- 4) Авторизация при необходимости (по заданию предметной области)
При вводе пароль должен быть скрыт маской ввода, но также должна быть реализована возможность просмотра введенного пароля.

При входе система выводит фамилию и имя пользователя, его роль.

- 5) Просмотр списка объектов (товаров, материалов, партнеров, пользователей и т.п.) и подробной информации об объекте (с изображением) по заданному макету; редактирование и добавление в новой форме с загрузкой информации в поля из БД, удаление объектов.

Пример: Список товаров/материалов

Получение актуальной информации об остатках товаров/материалов на складах, поэтому необходимо реализовать вывод товаров, которые хранятся в базе данных, согласно предоставленному макету. При отсутствии изображения необходимо вывести картинку-заглушку.

Вывод должен осуществляться списком:

<div><div>Фото</div></div>	<div>Наименование товара/материала Описание товара Производитель: Цена: Цена со скидкой Скидка:</div>	<div>Наличие на складе</div>
----------------------------	--	--------------------------------------

- 6) Выполнить отладку и тестирование модуля для проверки функциональности: приложение должно корректно работать и не должно происходить аварийного завершения работы.

Выполнить отладку программного обеспечения с использованием инструментальных средств. Сохранить и представить результаты в отчет в скриншотах.

Определить наборы входных данных и выполнить функциональное тестирование модуля по определенному сценарию. Провести тестирование для проверки функциональности программы (хотя бы 1 тест на 1 функцию). Использовать инструментальные средства для тестирования. Представить результаты тестирования в виде протокола тестирования, в соответствии со стандартами.

- 7) Разработать интерфейс программного модуля для работы с клиентами/товарами/материалами. Реализовать функции добавления/редактирования данных объекта (клиента, товара и т.п) в новом окне (странице) – форме для добавления/редактирования объекта. Переходы на эту форму должны быть реализованы из главной формы списка объектов: для редактирования – при нажатии на конкретный элемент, для добавления – при нажатии кнопки.

При открытии формы для редактирования все поля выбранного объекта должны быть подгружены в соответствующие поля из базы данных, а таблица заполнена актуальными значениями. После добавления/редактирования данные в окне списка объектов должны быть обновлены.

- 8) Обновление и сохранение данных в форме и БД. При вводе данных использование масок. Вычисляемые поля.

При разработке приложения выдерживать **стандарты разработки ПО**:

- Компоненты системы должны иметь единый согласованный внешний вид, соответствующий руководству по стилю
- Название приложения
- Организация файловой структуры проекта
- Структура проекта
- Макет и технические характеристики
- Обратная связь системы с пользователем
- Обработка ошибок (стабильная работа всех разработанных программ)
- Оформление кода (стиль кода на протяжении разработки всей системы, соблюдение культуры кодирования)
- Комментарии по коду

Модуль № 3: Сопровождение и обслуживание программного обеспечения компьютерных систем

Разработать интерфейс программного модуля для работы с заказами. Добавить функционал в систему для работы с объектами согласно требованиям заказчика. Необходимо осуществить вывод истории заказов клиентов: ФИО и рейтинг клиента, дата заказа, сумма заказа, пункт выдачи, статус заказа. Выполнить интеграцию модуля вывода истории заказов клиентом: обеспечить соответствие стилю приложения, единый для системы согласованный внешний вид. Реализовать переход на данное окно (страницу), получить историю заказов для конкретного клиента (установить фильтр по ФИО или кнопку перехода в окно).

Реализовать обработку исключительных ситуаций в приложении. Необходимо уведомлять пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях, предупреждать о неотвратимых операциях. 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.

Сессия 2:

Модуль 4. Осуществление интеграции программных модулей

С целью обеспечить одинаковый расчет скидки клиенту на товар, необходимо разработать метод.

***Примечание:** Если в вашем варианте не заданы критерии величины скидки клиента, то выполните вычисление по следующему условию:*

Величина скидки для клиента рассчитывается как действующая скидка на товар + дополнительная скидка клиента. Дополнительная скидка зависит от общей суммы покупок за весь период и составляет: до 50000 – 0%, от 50000 – до 100000 – 1%, от 100000 – до 300000 – 3%, более 300000 – 5%.

Необходимо загрузить исходный код метода расчета индивидуальной скидки в отдельный репозиторий с названием, совпадающим с названием проекта.

Разработка библиотеки (отдельным проектом в соответствии с вариантом)

При отсутствии требования к библиотеке в варианте выполните следующее:

Необходимо решить какие виды расчетов будут важны для Вашей предметной области, и реализовать этот расчет в виде библиотеки. Данная библиотека классов должна подключаться к основному проекту и быть представлена в виде .dll файла. Чтобы система правильно интегрировалась необходимо обязательно следовать правилам именования библиотек, классов и методов в них. Классы и методы должны содержать модификатор public, чтобы внешние приложения могли получить к ним доступ.

Документирование программных решений

Разработать руководство пользователя разработанного программного решения.

Тестовая документация

Разработка тестовых сценариев Test-cases

- 1) Для выполнения процедуры тестирования «Авторизации» Вам нужно описать два сценария. Авторизация может быть выполнима, а может быть отклонена согласно требованиям предметной области. Необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма. Для описания тестовых сценариев в ресурсах предоставлен шаблон *testing-template.docx*.
- 2) Для выполнения процедуры тестирования добавления/удаления товара (пользователя и т.п. объекта) администратором Вам нужно описать три сценария. 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*.

Представление результатов работы (Git /сервис Gogs)

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

- исходный код приложения (в виде коммита текущей версии проекта (структура с файлами, не архив)),
- скрипт БД с данными,
- исполняемые файлы,
- прочие графические / текстовые файлы (документация).

Результаты работы каждой сессии должны быть загружены в отдельный репозиторий с названием «*Сессия X*» (X – номер сессии).

Создание заметок в электронном виде (readme.md)

Руководитель практики Солопова Е.В.

(подпись)

«_____» _____ 20__ г.

Материалы в помощь для работы:

В качестве примера полное поэтапное описание выполнения задания Образец_Школа Great Britain.

ПРОГРАММНЫЕ РЕШЕНИЯ ДЛЯ БИЗНЕСА

Информационные и коммуникационные технологии

УЧЕБНО-МЕТОДИЧЕСКИЕ КУРСЫ

<https://nationalteam.worldskills.ru/skills/programmnye-resheniya-dlya-biznesa/>

ссылка для скачивания видео (без мобилок)

<https://disk.yandex.ru/d/BKAsdCLD2iGeEg>

Импорт и подготовка данных

Импорт csv: https://professorweb.ru/my/it/blog/net/csv_sql_server.php

Импорт xml: <https://blogadmin.day.ru/import-xml-fayla-v-bazu-dannyh-mssql/>

Подготовка файлов к импорту: <https://nationalteam.worldskills.ru/skills/rabota-s-nestrukturovannymi-dannymi-obrabotka-i-import-v-bazu-dannykh/>

Обработка поля с датой в виде длинного числа (разные формулы):

<https://www.extendoffice.com/documents/excel/2473-excel-timestamp-to-date.html>

Приведение дат к одному виду

1) Выделить все даты - **Ячейка -Формат -Дата**

2) Далее для удаления пробелов в начале дат тип 2015-02-04 выбрать **Заменить** в первом поле нажать пробел, во втором ничего и нажать **Заменить**