Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

ПРАКТИЧЕСКАЯ РАБОТА

ПО «ОАИП Python»

Разработка информационной системы

|  |  |
| --- | --- |
| Выполнил(а) студент(ка)  Группы П50-6-22  Мурлаева Екатерина Олеговна  «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2023 года | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_Т.Д. Артамонова  «\_\_\_\_» \_\_\_\_\_\_\_2023 года |

Москва 202

Цель работы: Разработать информационную систему по своей предметной области, используя инкапсуляция, наследование и полиморфизм

Ход работы:

* + 1. Создание кода

Код представляет собой простую систему управления базой данных для магазина телефонов.

В нем реализованы четыре класса: Database, User, Employee, и Tovar

Класс Database:

Данный класс обеспечивает взаимодействие с базой данных, создание таблиц, добавление данных и другие операции.

Основные функции класса Database:

* Создание таблиц для пользователей (users), сотрудников (employees), товаров (tovars), и заказов (orders)
* Методы для добавления пользователей, сотрудников, товаров и заказов
* Методы для удаления таблицы товаров и получения всех товаров из базы данных
* Методы для создания начальных записей о товарах и получения списка всех товаров

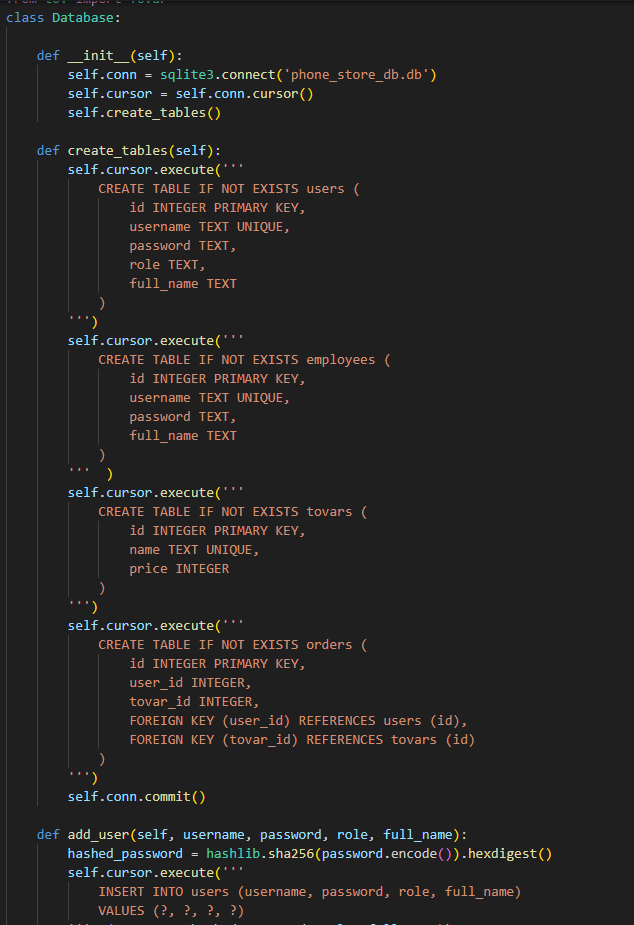


Рисунок 1 - Класс Database



Рисунок 2 – Класс Database

Классы User и Employee:

Данные классы предоставляют атрибуты для хранения информации (ID, логин, пароль, полное имя) о пользователе и сотруднике.

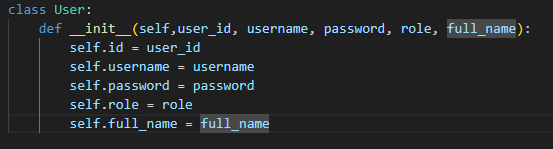


Рисунок 3 – Класс User

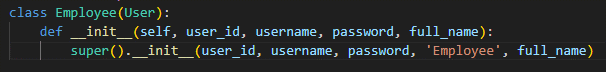


Рисунок 4 – Класс Employee

Класс Tovar:

Данный класс схож с двумя предыдущими, но предоставляет атрибуты для хранения информации (название, цена) о товаре в магазине.

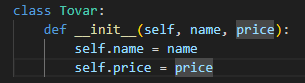


Рисунок 5 - Класс Tovar

* + 1. Функционал кода:
       - Регистрация и Вход

Пользователи могут зарегистрироваться, указав логин, пароль, роль (клиент, сотрудник, админ) и полное имя.

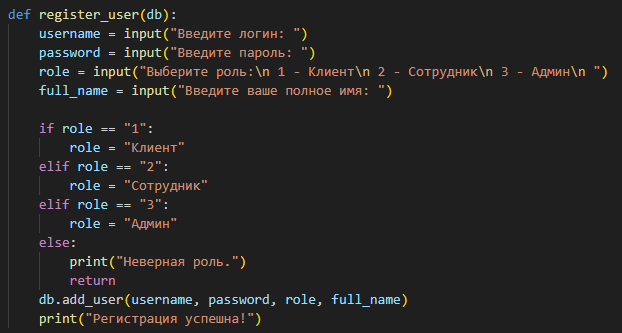


Рисунок 6 – Регистрация

* + - * + Пользователи могут войти в систему, указав свой логин и пароль.

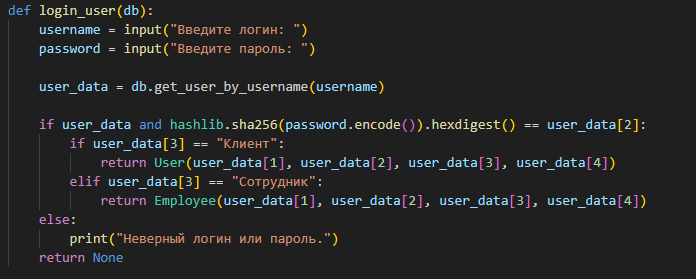


Рисунок 7 – Вход в систему

* + - * + Меню для клиента, сотрудника и админа

У каждого пользователя, в зависимости от того, какую роль он выбрал, будет появляться свое меню с разными выборами действий

* Клиент может просматривать товары, добавлять заказы,

просматривать свои заказы, изменять свои данные и выйти в меню регистрации, входа.

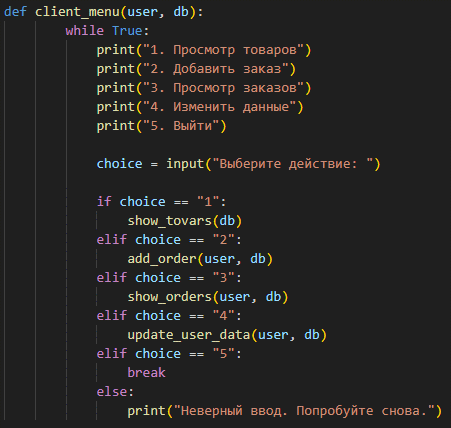


Рисунок 8 – Меню клиента

* Сотрудник может просматривать товары, добавлять товары,

удалять товары, изменять свои данные и выйти в меню регистрации, входа.

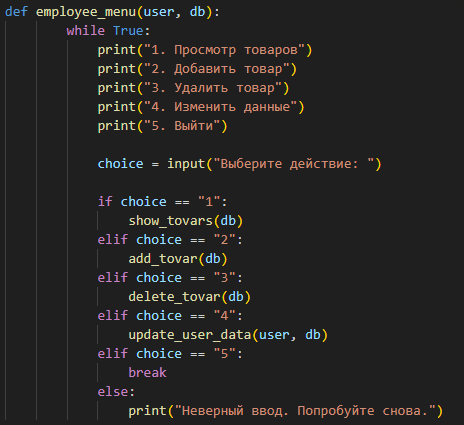


Рисунок 9 – меню сотрудника

* Администратор может просматривать сотрудников, добавлять

сотрудников, изменять свои данные и выйти в меню регистрации, входа.

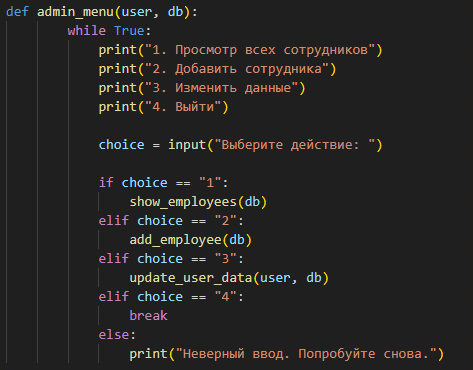


Рисунок 10 – меню админа

* + 1. Дополнительные функции

В зависимости от роли у каждого свое меню, где разный выбор действий, но у всех них есть одинаковая функция – обновление данных

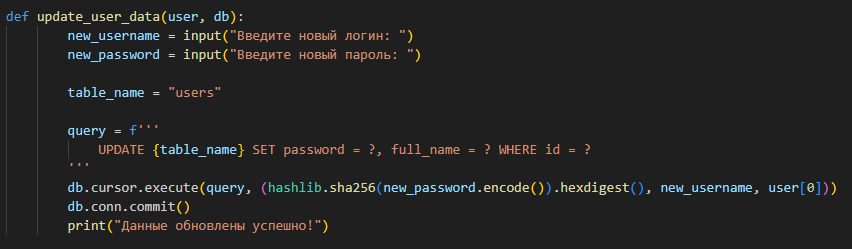


Рисунок 11 – Обновление данных

* Клиенты

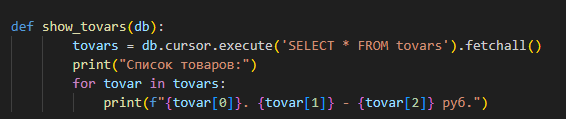


Рисунок 12 – Просмотр товаров

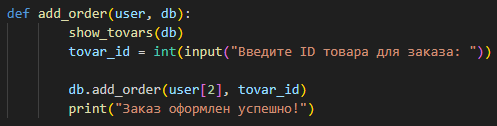


Рисунок 13 – Создание заказа, добавление туда товара

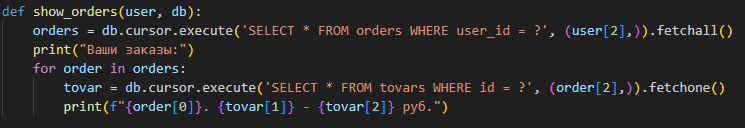


Рисунок 14 – Просмотр товаров в заказе

* Сотрудники

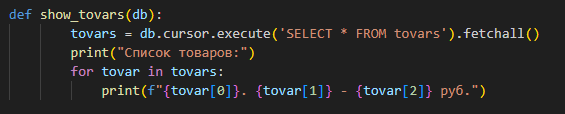


Рисунок 15 – Просмотр товаров

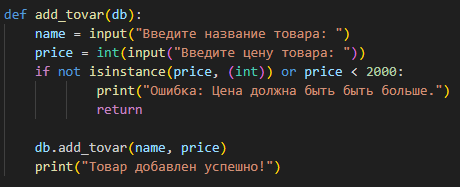


Рисунок 16 – Добавление товаров в список

В данной функции(add\_tovar) также идет проверка на то, чтоб цена добавленного товара было не менее 2000руб.

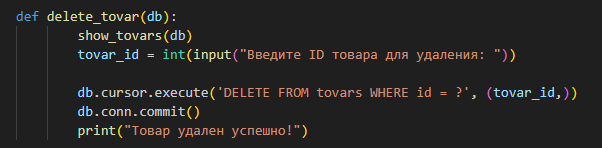


Рисунок 17 - Удаление товара из списка

* Админ

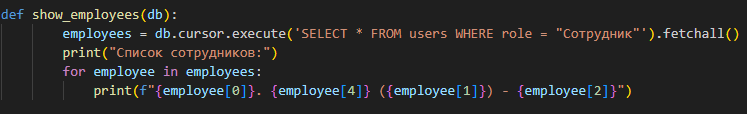


Рисунок 18 – Просмотр списка сотрудников

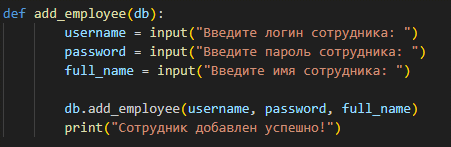


Рисунок 19 – Добавление нового сотрудника

* + 1. Функция main

В данном функции прописывается интерфейс в консоли, там вызывается функции для регистрации и авторизации, после в зависимости от роли у вас выполняются те или иные функции.



Рисунок 20 – Функция main

* + 1. Инкапсуляция, наследование и полиморфизм
* Инкапсуляция
* Класс Database

В данном примере conn и cursor являются приватными атрибутами класса, так как они не являются доступными извне класса. Они предоставляют интерфейс для взаимодействия с базой данных, но детали реализации скрыты.

* Метод add\_user в классе Database

Здесь hashed\_password - это локальная переменная, инкапсулированная внутри метода. Она не доступна извне класса и используется только внутри метода add\_user.

* Атрибуты классов User, Employee и Tovar

В каждом из этих классов атрибуты, такие как id, username, password, role, и full\_name, инкапсулированы внутри экземпляров класса и могут быть доступны только через методы и свойства классов.

* Наследование
* Класс Employee наследует от класса User

В этом примере класс Employee является подклассом класса User. Он наследует все атрибуты и методы класса User, добавляя при этом свой собственный функционал. Использование super().\_\_init\_\_ в конструкторе Employee также обеспечивает вызов конструктора базового класса User.

Таким образом, объекты класса Employee могут использовать функционал класса User и в то же время иметь свои уникальные атрибуты и методы.

* Полиморфизм
* Методы меню для клиента, сотрудника и админа

Они все обладают одинаковым интерфейсом для работы с разными ролями (клиент, сотрудник, админ). Каждый из этих методов реализует свою уникальную логику в соответствии с ролью пользователя.

* Методы show\_tovars, add\_tovar, delete\_tovar

В этих методах используются общие интерфейсы для работы с товарами, независимо от роли пользователя.

* Классы User и Employee

Класс Employee наследует от класса User и переопределяет метод \_\_init\_\_, т.е объект класса Employee может использоваться в тех местах, где ожидается объект класса User.

Вывод: в ходе работы была разработать информационная систему по своей предметной области, используя инкапсуляция, наследование и полиморфизм