

Na ocenę, będę pytał z programu.

Program działa w konsoli (nie ma wersji graficznej)

**Należy to zrobić samodzielnie wykorzystując swoją inteligencję (nie sztuczną). Lepiej napisać mniej a samodzielnie niż wygenerować gotowca
Jeżeli używasz technologii której nie znasz , to jej nie używaj
Zawsze możesz poprosić o pomoc**

Pracujemy na lekcji. Pod koniec każdej lekcji pokazujemy postępy pracy nad projektem , aby zaliczyć trzeba wykazać postęp prac na lekcji

Kryteria ocen

**na 2 -> dodanie metod, stworzenie obiektów , wymagane metody:
zarządzanie produktami (okrojone), obsługa sprzedaży (okrojona)**

na 3 -> to co na 2 tylko bardziej rozbudowane

na 4-> funkcjonalność sklepu z małymi uchybieniami

na 5-> pełna funkcjonalność

na 6 -> pełna funkcjonalność plus rozszerzenie

Założenia projektu: Program zarządzający małym sklepem

1. Cel programu:

Stworzenie aplikacji umożliwiającej zarządzanie asortymentem małego sklepu, sprzedażą produktów oraz obsługą klientów. Program powinien działać w trybie tekstowym, z wykorzystaniem menu w konsoli.

2. Zakres funkcjonalności:

Aplikacja powinna umożliwiać:

- Dodawanie, edycję i usuwanie produktów w sklepie.
- Wyświetlanie listy dostępnych produktów.
- Obsługę sprzedaży produktów (zmniejszanie stanu magazynowego po sprzedaży).
- ~~Obsługę klientów (rejestrowanie i wyświetlanie informacji o klientach).~~
- Zapisywanie i odczytywanie danych (produkty, ~~klienci~~, historia sprzedaży) z/do plików.
- Obsługę wyjątków, np. przy błędnym wprowadzaniu danych, braku produktu na magazynie itp.

3. Wymagania techniczne:

- Program ma być napisany w Pythonie.
- Zastosowanie klas w celu modelowania danych (np. produkt, klient, sklep).
- Wykorzystanie plików (tekstowych) do przechowywania danych.
- Obsługa wyjątków w sytuacjach takich jak błędne dane wejściowe, pliki niedostępne, brak produktów w magazynie.
- Menu użytkownika do zarządzania funkcjonalnościami programu.

Proponowane funkcjonalności:

1. Zarządzanie produktami:

- Dodawanie nowego produktu (nazwa, cena, ilość w magazynie, podatek VAT).
- Edycja danych produktu (np. zmiana ceny, aktualizacja stanu magazynowego).

- Usuwanie produktu z listy (jeżeli produkt jest na stanie, nie można go usunąć)
- Wyświetlanie wszystkich dostępnych produktów w sklepie.

2. Obsługa sprzedaży:

- Wybór produktu do sprzedaży (poprzez nazwę lub kod)
- Sprawdzanie dostępności produktu (czy jest wystarczająca ilość na magazynie).
- Obliczanie całkowitego kosztu transakcji.
- Aktualizacja stanu magazynowego po sprzedaży.
- ~~Zapisywanie historii sprzedaży do pliku (np. data, produkt, ilość, kwota).~~

3. ~~Zarządzanie klientami:~~

- ~~Dodawanie nowych klientów (imię, nazwisko, e-mail, telefon).~~
- ~~Wyświetlanie listy klientów.~~
- ~~Edycja danych klienta (np. zmiana numeru telefonu).~~
- ~~Usuwanie klientów.~~

4. Praca z plikami:

- Zapisywanie produktów, klientów i historii sprzedaży do plików.
- Odczyt danych z plików przy uruchamianiu programu.
- Obsługa błędów w przypadku braku pliku lub nieprawidłowego formatu danych.

5. Obsługa wyjątków:

- Obsługa błędów związanych z wprowadzaniem danych (np. wpisanie tekstu zamiast liczby).
- Obsługa braku plików lub ich nieprawidłowej zawartości.
- Obsługa sytuacji, w których użytkownik próbuje sprzedać więcej produktów, niż jest dostępnych.

Przykładowe klasy:

1. Produkt:

- Atrybuty: id, nazwa, cena, ilosc.
- Metody: dodaj/aktualizuj/usun produkt, wyświetl informacje o produkcie.

2. Klient:

- Atrybuty: id, imie, nazwisko, email, telefon.
- Metody: dodaj/edytuj/usun klienta, wyświetl informacje o kliencie.

3. Sklep:

- Atrybuty: lista produktów, lista klientów, historia sprzedaży.
- Metody: zarządzanie produktami, obsługa klientów, sprzedaż produktu, zapis/odczyt danych do/z plików.

Rozszerzenia (na ocenę celującą):

- Możliwość generowania raportów sprzedaży (np. za dany dzień/miesiąc).
- Filtrowanie produktów (np. wyświetlanie produktów z niskim stanem magazynowym).
- Proste logowanie użytkownika (np. administratora sklepu).