

Politechnika Świętokrzyska w Kielcach Wydział Elektrotechniki, Automatyki i Informatyki Katedra Informatyki, Elektroniki i Elektrotechniki		
Kierunek Informatyka	Projekt Języki Skryptowe	
Grupa dziekańska 2ID12B	Tema: Asystent zakupowy z porównywarką cen	Wykonał Maciej Kmita
Data wykonania 27.06.2025	Data oddania 27.06.2025	Ocena i podpis

1. Krótki opis problemu – przyjęte założenia

1. Cel projektu

Celem projektu było stworzenie aplikacji w języku Python pełniącej funkcję asystenta zakupowego, który umożliwia:

- Przeglądanie i filtrowanie produktów według kategorii,
- Sortowanie produktów według ceny,
- Dodawanie wybranych produktów do koszyka,
- Zapis koszyka do pliku w formacie JSON,
- Obsługę błędnych danych wejściowych,
- Automatyczne przywracanie koszyka po uruchomieniu programu.

Projekt został zaprojektowany zgodnie z wymaganiami przedstawionymi w formularzu samooceny i podzielony na moduły.

2. Struktura projektu

Pliki projektu:

- `main.py` – główny moduł programu, zawiera menu oraz sterowanie aplikacją,
 - `models.py` – definicje klas: `Product`, `Store`, `Cart`,
 - `utils.py` – funkcje pomocnicze, w tym generowanie przykładowych danych,
 - `data.json` – plik zawierający dane koszyka (tworzony automatycznie),
 - `test.py` – testy jednostkowe.
-

3. Opis działania

Po uruchomieniu programu wczytywane są dane produktów oraz – jeśli istnieje – zapisany wcześniej koszyk. Użytkownik ma do dyspozycji menu, które pozwala:

1. Wyświetlić wszystkie produkty,
2. Filtrować produkty po kategorii,
3. Sortować produkty po cenie,
4. Dodać wybrany produkt do koszyka,
5. Zapisać koszyk do pliku `data.json`,
6. Wyjść z programu.

Przy wyjściu dane koszyka można zachować, co pozwala na jego późniejsze przywrócenie.

4. Wykorzystane elementy języka Python (zgodne z formularzem)

- **Podstawy języka:** zmienne, operatory, instrukcje warunkowe i pętle (`for`, `while`), funkcje z parametrami i wartością zwracaną,
- **Kontenery:** listy, słowniki, krotki,

- **Programowanie obiektowe:** klasy, metody, konstruktory, dziedziczenie,
 - **Obsługa wyjątków:** `try`, `except`, `finally`, `assert`,
 - **Modularność:** podział na moduły i import własnych plików,
 - **Obsługa plików:** zapis i odczyt w formacie `.json`,
 - **Testowanie:** testy jednostkowe przy użyciu `unittest`, testy błędów,
 - **Formatowanie stringów i operacje na łańcuchach znaków.**
-

5. Testowanie

Zaimplementowano testy jednostkowe dla:

- Klasy `Product` – poprawność tworzenia obiektu,
- Klasy `Store` – filtrowanie i sortowanie produktów,
- Klasy `Cart` – dodawanie produktów i zapis do pliku.

Plik testowy: `test_asystent_zakupowy.py`

6. Przykładowe dane wejściowe i wyjściowe

Przykład wejścia:

Wybierz opcję: 2

Podaj kategorię: Elektronika

Przykład wyjścia:

Sklep A:

- Laptop (Elektronika): 3000.00 zł
- Mysz (Elektronika): 100.00 zł

Sklep B:

- Laptop (Elektronika): 2800.00 zł
- Mysz (Elektronika): 120.00 zł

Zawartość zapisanego pliku data.json:

```
[  
  {"name": "Laptop", "category": "Elektronika", "price": 2800},  
  {"name": "Książka", "category": "Edukacja", "price": 40}  
]
```

7. Podsumowanie

Projekt zrealizowano zgodnie z wymaganiami. Aplikacja jest modularna, testowalna i odporna na błędy użytkownika. Dzięki podziałowi na moduły kod jest czytelny i łatwy do rozbudowy. Możliwe rozszerzenia w przyszłości obejmują integrację z API porównywarek cen lub rozbudowę interfejsu o wersję graficzną.