

Dr inż. Dariusz Michalski. Formularz samooceny do projektu z języków skryptowych

Nr	Obszar	Wymaganie	KOD	Przyznane pkt
1	UI	Wprowadzanie danych	✓	<code>main.py</code> : 153-170 (dodawanie produktów)
		Wyświetlanie danych	✓	<code>main.py</code> : 108-138 (wyświetlanie produktów)
		Zmiana danych	✓	<code>main.py</code> : 172-185 (usuwanie z koszyka)
		Wyszukiwanie danych	✓	<code>main.py</code> : 108-138 (filtrowanie produktów)
		Przedstawienie wyników	✓	<code>cart.py</code> : 30-33 (metoda <code>show()</code>)
2	Podstawy	Zmienne	✓	W całym kodzie (np. <code>product.py</code> : 1-20)
		Typy danych	✓	<code>product.py</code> : 1-20 (definicja klasy)
		Komentarze	✓	<code>main.py</code> : 12-15 (dekorator)
		Operatory	✓	<code>utils.py</code> : 2 (<code>sorted</code> , <code>lambda</code>)
		Instrukcje warunkowe (<code>if</code> , <code>elif</code> , <code>else</code>)	✓	<code>main.py</code> : 108-138 (filtrowanie)
		Instrukcje iteracyjne <code>for</code>	✓	<code>cart.py</code> : 30-33 (pętla <code>for</code>)
		Instrukcje iteracyjne <code>while</code>	✓	<code>main.py</code> : 88-96 (pętla <code>while</code>)
		Operacje wejścia (<code>input</code>)	✓	<code>main.py</code> : 153 (<code>input</code>)
		Operacje wyjścia (<code>print</code>)	✓	<code>cart.py</code> : 32 (<code>print</code>)
		Funkcje z parametrami i wartościami zwracanymi	✓	<code>utils.py</code> : 1-2
		Funkcje rekurencyjne	✓	<code>main.py</code> : 51-63
		Funkcje przyjmujące inne funkcje jako argumenty	✓	<code>utils.py</code> : 1-2

Nr	Obszar	Wymaganie	KOD	Przyznane pkt
3	Kontenery	Dekoratory	✓	main.py : 12-21
		Użycie listy	✓	store.py : 5 (lista produktów)
		Użycie słownika	✓	sample_data.py : 4-17 (model_prefixes)
		Użycie zbioru	✓	main.py : 222 (set)
4	Przestrzenie nazw	Użycie krotki	✓	sample_data.py : 79 (products.append)
		Zastosowano zmienne lokalne	✓	main.py : 51-63 (cart_items)
		Zastosowano zmienne globalne	✓	main.py : 88 (stores)
		Zastosowano zakresy funkcji	✓	main.py : 12-21 (wrapper)
5	Moduły i pakiety	Zastosowano zakresy klas	✓	product.py : 1-20
		Projekt podzielony na moduły (import, init)	✓	main.py : 1-8 (importy)
		Własne pakiety/funkcje pomocnicze w osobnych plikach .py	✓	logic/utils.py
6	Obsługa błędów	Obsługa wyjątków (try, except, finally)	✓	cart.py : 38-42
		Użycie assert do testów i walidacji	✓	main.py : 60
7	Łańcuchy znaków	Operacje na stringach (m.in. formatowanie, dzielenie, wyszukiwanie)	✓	product.py : 19-20 (repr)
8	Obsługa plików	Odczyt z plików .txt, .csv, .json, .xml (min. 1)	✓	cart.py : 38-42 (JSON)
		Zapis do plików .txt, .csv, .json, .xml (min. 1)	✓	cart.py : 23-28 (JSON)
9	OOP	Klasy	✓	product.py : 1-20

Nr	Obszar	Wymaganie	KOD	Przyznane pkt
		Metody	<input checked="" type="checkbox"/>	<code>cart.py</code> : 8-10 (add)
		Konstruktory	<input checked="" type="checkbox"/>	<code>product.py</code> : 7-14
		Dziedziczenie	<input checked="" type="checkbox"/>	<code>limited_edition_product.py</code> : 1-12
10	Programowanie funkcyjne	map	<input checked="" type="checkbox"/>	<code>main.py</code> : 45
		filter	<input checked="" type="checkbox"/>	<code>main.py</code> : 118-121
		lambda	<input checked="" type="checkbox"/>	<code>main.py</code> : 45
		reduce	<input checked="" type="checkbox"/>	<code>main.py</code> : 45
11	Wizualizacja danych	Wygenerowano wykres (np. matplotlib, seaborn)	<input checked="" type="checkbox"/>	<code>main.py</code> : 65-136
		Zapisano wykres do pliku graficznego (.png lub .jpg)	<input checked="" type="checkbox"/>	<code>main.py</code> : 134 (plt.show())
12	Testowanie	Testy jednostkowe (assert, unittest, pytest)	<input checked="" type="checkbox"/>	<code>tests/test_product.py</code>
		Testy funkcjonalne	<input checked="" type="checkbox"/>	<code>tests/test_menu_flow_integration.py</code>
		Testy integracyjne	<input checked="" type="checkbox"/>	<code>tests/test_cart_integration.py</code>
		Testy graniczne / błędne dane	<input checked="" type="checkbox"/>	<code>tests/test_cart.py</code> : 38-42
		Testy wydajności (np. czas wykonania, timeit)	<input type="checkbox"/>	-
		Testy pamięci memory_profiler	<input type="checkbox"/>	-
		Test jakości kodu (flake8, pylint)	<input checked="" type="checkbox"/>	flake8 - domyślne ustawienia
13	Wersjonowanie	Repozytorium GIT	<input checked="" type="checkbox"/>	Istnieje - Na Github
		Historia commitów	<input checked="" type="checkbox"/>	Istnieje

Nr	Obszar	Wymaganie	KOD	Przyznane pkt
		Link do GitHub	<input checked="" type="checkbox"/>	https://github.com/jacekjarosz55/shopping-assistant
		Opis commitów	<input checked="" type="checkbox"/>	commity posiadają wiadomosci opisujące zmiany
14	Dokumentacja	Plik README.md (cel, autorzy, uruchamianie)	<input checked="" type="checkbox"/>	README.md
		Przykładowe dane wejściowe i wyjściowe	<input checked="" type="checkbox"/>	dokumentacja/przykładowe-uzycie.txt
		Diagram klas lub struktura modułów	<input checked="" type="checkbox"/>	dokumentacja/diagram.png
	SUMA			77