

N r	Obszar	Wymaganie	KOD		Przyzna ne pkt	Pkt max
1	UI	JEST		<input type="checkbox"/>		
		Wprowadzanie danych	<pre>product_id = int(input("Podaj ID produktu do dodania do koszyka: "))</pre>	<input type="checkbox"/>		2
		Wyświetlanie danych	<pre>print("\nLista produktów:") display_products(p roducts)</pre>	<input type="checkbox"/>		2
		Zmiana danych		<input type="checkbox"/>		2
		Wyszukiwanie danych	<pre>category = input("Podaj nazwę kategorii do filtrowania: ") filtered = filter_products_by _category(products , category)</pre>	<input type="checkbox"/>		2
		Przedstawienie wyników	<pre>print(f"\nProdukty w kategorii '{category}':") display_products(f iltered)</pre>	<input type="checkbox"/>		2
2	Podstawy	Zmienne	<pre>product = Product(id=item['id'], name=item['name'], category=item['cat egory'], price=float(item[' price']), store_name=item['s tore_name'])</pre>	<input type="checkbox"/>		2

		typy danych	<pre>price=float(item['price']),</pre>	<input type="checkbox"/>	2
		komentarze	<pre># Wywołanie funkcji wizualizacyjnych - nie sprawdzają się w testach automatycznych, # ale można je wywołać, aby zobaczyć wykres</pre>	<input type="checkbox"/>	1
		operatory	<pre>if __name__ == '__main__': unittest.main()</pre>	<input type="checkbox"/>	1,5
		Instrukcje warunkowe (if, elif, else)	<pre>while True: print("\n--- MENU ---") print("1. Wyświetl produkty") print("2. Posortuj produkty") print("3. Filtruj produkty po kategorii") print("4. Dodaj produkt do koszyka") print("5. Wyświetl koszyk") print("6. Wykonaj wizualizacje") print("0. Wyjście") choice = input("Wybierz opcję: ") if choice == '1':</pre>	<input type="checkbox"/>	3

			<pre>print("\nLista produktów:") display_products(p roducts)</pre>			
	Instrukcje iteracyjne		<pre>for item in data:</pre>			
	for		<pre>for item in data:</pre>	<input type="checkbox"/>		2
	while		<pre>while True: print("\n--- MENU ---") print("1. Wyświetl produkty") print("2. Posortuj produkty") print("3. Filtruj produkty po kategorii") print("4. Dodaj produkt do koszyka") print("5. Wyświetl koszyk") print("6. Wykonaj wizualizacje") print("0. Wyjście") choice = input("Wybierz opcję: ")</pre>	<input type="checkbox"/>		2
	Operacje wejścia (input)		<pre>order = input("Wybierz kolejność (asc/desc): ").lower()</pre>	<input type="checkbox"/>		1,5
	Operacje wyjścia (print)		<pre>print(f"\nProdukty w kategorii '{category}':")</pre>	<input type="checkbox"/>		1,5

		Funkcje z parametrami i wartościami zwracanymi	<pre>def save_products_to_j son(products, filename):</pre>	<input type="checkbox"/>		2
		Funkcje rekurencyjne		<input type="checkbox"/>		3
		Funkcje przyjmujące inne funkcje jako argumenty		<input type="checkbox"/>		3
		Dekoratory		<input type="checkbox"/>		1,5
3	Kontenery	Użycie listy		<input type="checkbox"/>		2
		Użycie słownika	<pre>'id': p.id, 'name': p.name, 'category': p.category, 'price': p.price, 'store_name': p.store_name</pre>	<input type="checkbox"/>		2
		Użycie zbioru		<input type="checkbox"/>		1,5
		Użycie krotki		<input type="checkbox"/>		1,5
4	Przestrzenie nazw	Zastosowano zmienne lokalne	<pre>choice = input("Wybierz opcję: ")</pre>	<input type="checkbox"/>		1,5
		Zastosowano zmienne globalne	<pre>filename = 'products.json'</pre>	<input type="checkbox"/>		1,5
		Zastosowano zakresy funkcji		<input type="checkbox"/>		1,5
		Zastosowano zakresy klas		<input type="checkbox"/>		1,5
5	Moduły i pakiety	Projekt podzielony na moduły (import, __init__)		<input type="checkbox"/>		2

N r	Obszar	Wymaganie	KOD		Przyzna ne pkt	Pkt max
		Własne pakiety/funkcje pomocnicze w osobnych plikach .py		<input type="checkbox"/>		2
6	Obsługa błędów	Obsługa wyjątków (try, except, finally)	<pre>try: product_id = int(input("Podaj ID produktu do dodania do koszyka: ")) product = next((p for p in products if p.id == product_id), None) if product: cart.add_product(p roduct) print(f"Produkt {product.name} dodany do koszyka.") else: print("Nie znaleziono produktu o podanym ID.") except ValueError: print("Niepoprawne ID.")</pre>	<input type="checkbox"/>		2
		Użycie assert do testów i walidacji	<pre>self.assertEqual(s orted_products[0]. name, "Książka ABC")</pre>	<input type="checkbox"/>		1,5
7	Łańcuchy znaków	Operacje na stringach (m.in. formatowanie, dzielenie,	<pre>sorted_products = sort_products(prod ucts, reverse=reverse)</pre>	<input type="checkbox"/>		2

		wyszukiwanie)				
8	Obsługa plików	Odczyt z plików .txt, .csv, .json, .xml (min. 1)	<pre>products = [Product(1, "Laptop ABC", "Elektronika", 1200.0, "SklepX"), Product(2, "Książka DEF", "Książki", 45.0, "KsięgarniaY"), Product(3, "Odkurzacze", "Dom i ogród", 300.0, "SklepZ"), Product(4, "Smartphone", "Elektronika", 800.0, "SklepX"), Product(5, "Mata do jogi", "Sport", 50.0, "SklepSport")] save_products_to_j son(products, filename)</pre>	<input type="checkbox"/>		2
		Zapis do plików .txt, .csv, .json, .xml (min. 1)		<input type="checkbox"/>		2
9	OOP	Klasy		<input type="checkbox"/>		2
		Metody		<input type="checkbox"/>		2
		Konstruktory		<input type="checkbox"/>		2
		Dziedziczenie		<input type="checkbox"/>		2
10	Programowa nie funkcyjne	map		<input type="checkbox"/>		1,5
		filter		<input type="checkbox"/>		1,5
		lambda		<input type="checkbox"/>		1,5
		reduce		<input type="checkbox"/>		1,5
11	Wizualizacja danych	Wygenerowano wykres (np. matplotlib, seaborn)	<pre>plot_products_by_c ategory(products)</pre>	<input type="checkbox"/>		2

			<pre>plot_total_price_per_category(products)</pre>			
		Zapisano wykres do pliku graficznego (.png lub .jpg)		<input type="checkbox"/>		1,5
T1 2	Testowanie	Testy jednostkowe (assert, unittest, pytest)	<pre>def test_sort_products_ascending(self): sorted_products = sort_products(self.products) self.assertEqual(sorted_products[0].name, "Książka ABC") self.assertEqual(sorted_products[-1].name, "Telefon XYZ")</pre>	<input type="checkbox"/>		1,5
		Testy funkcjonalne		<input type="checkbox"/>		1,5
		Testy Integracyjne		<input type="checkbox"/>		1,5
		Testy graniczne / błędne dane		<input type="checkbox"/>		1,5
		Testy wydajności (np. czas wykonania, timeit)		<input type="checkbox"/>		1,5
		Testy pamięci memory_profiler		<input type="checkbox"/>		1,5
		Test jakości kodu (flake8, pylint)		<input type="checkbox"/>		1,5
13	Wersjonowanie	Repozytorium GIT		<input type="checkbox"/>		1
		Historia commitów		<input type="checkbox"/>		1

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

N r	Obszar	Wymaganie	KOD		Przyzna ne pkt	Pkt max
		Link do GitHub		<input type="checkbox"/>		1
		Opis commitów		<input type="checkbox"/>		1
14	Dokumentacja	Plik README.md (cel, autorzy, uruchamianie)		<input type="checkbox"/>		1,5
		Przykładowe dane wejściowe i wyjściowe		<input type="checkbox"/>		2
		Diagram klas lub struktura modułów		<input type="checkbox"/>		2
		SUMA				