

Nr	Obszar	Wymaganie	KOD		Przyznane pkt	Pkt max
1	UI	JEST		<input checked="" type="checkbox"/>		
		Wprowadzanie danych	self.amount_entry.get()	<input checked="" type="checkbox"/>	2	2
		Wyświetlanie danych	self.tree.insert(...)	<input checked="" type="checkbox"/>	2	2
		Zmiana danych	self.edit_transaction()	<input checked="" type="checkbox"/>	2	2
		Wyszukiwanie danych	self.search_transactions()	<input checked="" type="checkbox"/>	2	2
		Przedstawienie wyników	pokaz_saldo_miesieczne()	<input checked="" type="checkbox"/>	2	2
2	Podstawy	Zmienne	self.user, self.budget, username, amount, description, transactions	<input checked="" type="checkbox"/>	2	2
		typy danych	float, str, list, dict, set, tuple, bool, datetime.date	<input checked="" type="checkbox"/>	2	2
		komentarze	#	<input checked="" type="checkbox"/>	1	1
		operatory	+, -, ==, !=, >, <, >=, <=, and, or, in	<input checked="" type="checkbox"/>	1,5	1,5
		Instrukcje warunkowe (if, elif, else)	if amount > 0:	<input checked="" type="checkbox"/>	3	3
		Instrukcje iteracyjne				
		for	for t in self.budget.transactions	<input checked="" type="checkbox"/>	2	2
		while	while True:	<input checked="" type="checkbox"/>	2	2
		Operacje wejścia (input)	self.username_entry.get()	<input checked="" type="checkbox"/>	1,5	1,5
		Operacje wyjścia (print)	log_action	<input checked="" type="checkbox"/>	1,5	1,5
		Funkcje z parametrami i wartościami zwracanymi	def add_transaction(t):	<input checked="" type="checkbox"/>	2	2
		Funkcje rekurencyjne	recursive_sum()	<input checked="" type="checkbox"/>	3	3
		Funkcje przyjmujące inne funkcje jako argumenty	filter_transactions(transactions, condition_func)	<input checked="" type="checkbox"/>	3	3
		Dekoratory	@log_action	<input checked="" type="checkbox"/>	1,5	1,5
3	Kontenery	Użycie listy	transactions = []	<input checked="" type="checkbox"/>	2	2
		Użycie słownika	{"Jedzenie": 200}	<input checked="" type="checkbox"/>	2	2
		Użycie zbioru	set([t.category for t in transactions])	<input checked="" type="checkbox"/>	1,5	1,5
		Użycie krotki	(kwota, kategoria)	<input checked="" type="checkbox"/>	1,5	1,5
4	Przestrzenie nazw	Zastosowano zmienne lokalne	def f(): x = 1	<input checked="" type="checkbox"/>	1,5	1,5
		Zastosowano zmienne globalne	global logged_user	<input checked="" type="checkbox"/>	1,5	1,5

		Zastosowano zakresy funkcji	nonlocal counter	<input checked="" type="checkbox"/>	1,5	1,5
		Zastosowano zakresy klas	self.amount	<input checked="" type="checkbox"/>	1,5	1,5
5	Moduły i pakiety	Projekt podzielony na moduły (import, __init__)	import charts	<input checked="" type="checkbox"/>	2	2
Nr	Obszar	Wymaganie	KOD		Przyznane pkt	Pkt max
		Własne pakiety/funkcje pomocnicze w osobnych plikach .py	utils.py	<input checked="" type="checkbox"/>	2	2
6	Obsługa błędów	Obsługa wyjątków (try, except, finally)	try: ... except ValueError:	<input checked="" type="checkbox"/>	2	2
		Użycie assert do testów i walidacji	assert isinstance(transaction, Transaction)	<input checked="" type="checkbox"/>	1,5	1,5
7	Łańcuchy znaków	Operacje na stringach (m.in. formatowanie, dzielenie, wyszukiwanie)	desc.upper()desc.upper(), f'{amount:.2f} zł"	<input checked="" type="checkbox"/>	2	2
8	Obsługa plików	Odczyt z plików .txt, .csv, .json, .xml (min. 1)	json.load(...)	<input checked="" type="checkbox"/>	2	2
		Zapis do plików .txt, .csv, .json, .xml (min. 1)	json.dump(...)	<input checked="" type="checkbox"/>	2	2
9	OOP	Klasy	class Budget:	<input checked="" type="checkbox"/>	2	2
		Metody	def add_transaction(...):	<input checked="" type="checkbox"/>	2	2
		Konstruktory	__init__	<input checked="" type="checkbox"/>	2	2
		Dziedziczenie	class IncomeTransaction(Transaction)	<input checked="" type="checkbox"/>	2	2
10	Programowanie funkcyjne	map	map(lambda x: x.amount, transactions)	<input checked="" type="checkbox"/>	1,5	1,5
		filter	filter(lambda x: x.amount > 0, ...)	<input checked="" type="checkbox"/>	1,5	1,5
		lambda	lambda x: x.category	<input checked="" type="checkbox"/>	1,5	1,5
		reduce	reduce(lambda a, b: a + b, lista)	<input checked="" type="checkbox"/>	1,5	1,5
11	Wizualizacja danych	Wygenerowano wykres (np. matplotlib, seaborn)	plt.bar(...)	<input checked="" type="checkbox"/>	2	2
		Zapisano wykres do pliku graficznego (.png lub .jpg)	fig.savefig(...)	<input checked="" type="checkbox"/>	1,5	1,5

12	Testowanie	Testy jednostkowe (assert, unittest, pytest)	unittest.TestCase	<input checked="" type="checkbox"/>	1,5	1,5
		Testy funkcjonalne	test_login_valid()	<input checked="" type="checkbox"/>	1,5	1,5
		Testy Integracyjne	test_gui_login_flow()	<input checked="" type="checkbox"/>	1,5	1,5
		Testy graniczne / błędne dane	test_invalid_date()	<input checked="" type="checkbox"/>	1,5	1,5
		Testy wydajności (np. czas wykonania, timeit)	Brak	<input type="checkbox"/>	0	1,5
		Testy pamięci memory_profiler	Brak	<input type="checkbox"/>	0	1,5
		Test jakości kodu (flake8, pylint)	Brak	<input type="checkbox"/>	0	1,5
		13	Wersjonowanie	Repozytorium GIT	git init	<input checked="" type="checkbox"/>
Historia commitów	git log			<input checked="" type="checkbox"/>	1	1
Nr.	Obszar	Wymaganie	KOD		Przyznane pkt	Pkt max
		Link do GitHub	https://github.com/BartlomiejKonopka03/BudzetOsobisty.git	<input checked="" type="checkbox"/>	1	1
		Opis commitów	git commit -m 'Poprawy raport.pdf'	<input checked="" type="checkbox"/>	1	1
14	Dokumentacja	Plik README.md (cel, autorzy, uruchamianie)	README.md	<input checked="" type="checkbox"/>	1,5	1,5
		Przykładowe dane wejściowe i wyjściowe	input: kwota → zapis do JSON	<input checked="" type="checkbox"/>	2	2
		Diagram klas lub struktura modułów	diagram.png	<input checked="" type="checkbox"/>	2	2
SUMA					95,5	100