

Nr	Obszar	Wymaganie	KOD		Przyznane pkt	Pkt max
1	UI	JEST	import tkinter as tk	<input checked="" type="checkbox"/>		
		Wprowadzanie danych	def open_add_animal_window(self): name_entry = ttk.Entry(add_window, width=20)	<input checked="" type="checkbox"/>		2
		Wyświetlanie danych	def refresh_animals_tree(self): self.animals_tree.insert("", "end", values=( animal_id, animal.name, animal.age, ...))	<input checked="" type="checkbox"/>		2
		Zmiana danych	def open_edit_animal_window(self): def save_changes(): name = name_entry.get().strip()	<input checked="" type="checkbox"/>		2
		Wyszukiwanie danych	def open_animal_search_window(self) : ... def perform_search(): ... filtered = [(k, v) for k, v in filtered if id_query == k]	<input checked="" type="checkbox"/>		2
		Przedstawienie wyników	def open_animal_search_window(self) : ... def perform_search(): ... results_tree.insert("", "end", values=( animal_id, animal.name, animal.age, ...))	<input checked="" type="checkbox"/>		2
2	Podstawy	Zmienne	species = species_var.get() animal_id = str(self.next_id)	<input checked="" type="checkbox"/>		2
		typy danych	age = int(age_entry.get().strip()) animal_id = str(self.next_id)	<input checked="" type="checkbox"/>		2
		komentarze	class ShelterApp: # Inicjalizacja aplikacji z interfejsem i danymi def __init__(self, root):	<input checked="" type="checkbox"/>		1
		operatory	Tak	<input checked="" type="checkbox"/>		1,5
		Instrukcje warunkowe (if, elif, else)	def get_key(x): if self.animals_sort_default: return int(x[0]) if self.animals_sort_column == "ID": return int(x[0]) elif self.animals_sort_column == "Imię": ... else: return "Adoptowane" if x[1].is_adopted else "W schronisku"	<input checked="" type="checkbox"/>		3

		Instrukcje iteracyjne			
		for	for animal_id, animal in sorted_items:	<input checked="" type="checkbox"/>	2
		while	while row_index < len(list(reader)):	<input checked="" type="checkbox"/>	2
		Operacje wejścia (input)	def open_add_animal_window(self): name_entry = ttk.Entry(add_window, width=20)	<input checked="" type="checkbox"/>	1,5
		Operacje wyjścia (print)	except Exception as e: print(f"Błąd zapisu zwierząt: {e}")	<input checked="" type="checkbox"/>	1,5
		Funkcje z parametrami i wartościami zwracanymi	def get_feeding_status(self): last_fed_time = datetime.strptime(self.last_fed, "%Y-%m-%d %H:%M:%S") delta = datetime.now() - last_fed_time	<input checked="" type="checkbox"/>	2
		Funkcje rekurencyjne	def count_animals_by_type(self, animals, animal_class, index=0): if index >= len(animals): return 0 count = 1 if isinstance(animals[index][1], animal_class) and not animals[index][1].is_adopted else 0 return count + self.count_animals_by_type(animals, animal_class, index + 1)	<input checked="" type="checkbox"/>	3
		Funkcje przyjmujące inne funkcje jako argumenty	def log_action(func):	<input checked="" type="checkbox"/>	3
3	Kontenery	Dekoratory	def log_action(func): @wraps(func) def wrapper(*args, **kwargs): return func(*args, **kwargs) return wrapper	<input checked="" type="checkbox"/>	1,5
		Użycie listy	filtered = list(self.animals.items())	<input checked="" type="checkbox"/>	2
		Użycie słownika	self.animals = {str(k): self.species_map[v["species"]](str(k), v["name"], v["age"]) for k, v in data["animals"].items() }	<input checked="" type="checkbox"/>	2
		Użycie zbioru		<input type="checkbox"/>	1,5
4	Przestrzenie nazw	Użycie krotki	values=( animal_id, animal.name, animal.age	<input checked="" type="checkbox"/>	1,5
		Zastosowano zmienne lokalne	name = name_entry.get().strip()	<input checked="" type="checkbox"/>	1,5
		Zastosowano zmienne globalne	self.animals = self.data_manager.animals	<input checked="" type="checkbox"/>	1,5
		Zastosowano zakresy funkcji	def validate_and_add(): name = name_entry.get().strip()	<input checked="" type="checkbox"/>	1,5
5	Moduły i pakiety	Zastosowano zakresy klas	self.animals = {}	<input checked="" type="checkbox"/>	1,5
		Projekt podzielony na moduły (import, _init_)	from data_manager import DataManager	<input checked="" type="checkbox"/>	2

Nr	Obszar	Wymaganie	KOD		Przyznane pkt	Pkt max
		Własne pakiety/funkcje pomocnicze w osobnych plikach .py	from data_manager import DataManager	<input checked="" type="checkbox"/>		2
6	Obsługa błędów	Obsługa wyjątków (try, except, finally)	try: with open(self.animals_filename, 'r', encoding='utf-8') as f: data = json.load(f) except (FileNotFoundError, json.JSONDecodeError, KeyError):	<input checked="" type="checkbox"/>		2
		Użycie assert do testów i walidacji	assert name and age >= 0, "Imię puste lub wiek ujemny"	<input checked="" type="checkbox"/>		1,5
7	Łańcuchy znaków	Operacje na stringach (m.in. formatowanie, dzielenie, wyszukiwanie)	name_query = name_entry.get().strip().lower()	<input checked="" type="checkbox"/>		2
8	Obsługa plików	Odczyt z plików .txt, .csv, .json, .xml (min. 1)	with open(file_path, 'r', encoding='utf-8') as f: reader = csv.reader(f, delimiter=',')	<input checked="" type="checkbox"/>		2
		Zapis do plików .txt, .csv, .json, .xml (min. 1)	with open(file_path, 'w', newline='', encoding='utf-8') as f: writer = csv.writer(f, delimiter=',')	<input checked="" type="checkbox"/>		2
9	OOP	Klasy	class Animal: def __init__(self, id, name, age):	<input checked="" type="checkbox"/>		2
		Metody	def get_feeding_status(self): if not self.last_fed: return "Brak danych"	<input checked="" type="checkbox"/>		2
		Konstruktory	def __init__(self, root): self.root = root	<input checked="" type="checkbox"/>		2
		Dziedziczenie	class Dog(Animal):	<input checked="" type="checkbox"/>		2
10	Programowanie funkcyjne	map		<input type="checkbox"/>		1,5
		filter	vaccinated = len(list(filter(lambda x: x[1].is_vaccinated and not x[1].is_adopted, animals_list)))	<input checked="" type="checkbox"/>		1,5
		lambda	get_key = lambda x: int(x[0]) if col == "ID" else x[1].name	<input checked="" type="checkbox"/>		1,5
		reduce		<input type="checkbox"/>		1,5
11	Wizualizacja danych	Wygenerowano wykres (np. matplotlib, seaborn)	def generate_report(): ... if chart_data_var.get() == "Szczepienia": labels, data = ["Zaszczepione", "Niezaszczepione"], [vaccinated,	<input checked="" type="checkbox"/>		2

			not_vaccinated]			
		Zapisano wykres do pliku graficznego (.png lub .jpg)	def save_chart(): fig.savefig(file_path)	<input checked="" type="checkbox"/>		1,5
T12	Testowanie	Testy jednostkowe (assert, unittest, pytest)	def test_get_feeding_status(self): a = Dog("1", "Reksio", 3) a.last_fed = (datetime.now() - timedelta(hours=5)).strftime("%Y-%m-%d %H:%M:%S") self.assertIn("5", a.get_feeding_status())	<input checked="" type="checkbox"/>		1,5
		Testy funkcjonalne	def test_add_animal(self): self.app.animals = {} self.app.next_id = 1	<input checked="" type="checkbox"/>		1,5
		Testy Integracyjne	def test_export_import_animals(self): : animal = Dog("1", "Reksio", 5)	<input checked="" type="checkbox"/>		1,5
		Testy graniczne / błędne dane	with open(self.test_csv, "w", encoding="utf-8") as f:  f.write("ID;Imię;Wiek;Gatunek;Zaszczepione;Ostatnie karmienie;Data przyjęcia;Status\n")  f.write("1;;-1;Nieznany;Tak;invalid_date;invalid_date;Adoptowane\n") errors = self.data_manager.import_animals_csv(self.test_csv, replace=True) self.assertIn("Imię nie może być puste", errors[0])	<input checked="" type="checkbox"/>		1,5
		Testy wydajności (np. czas wykonania, timeit)	def test_performance_save_animals(self): duration = timeit(lambda: self.dm.save_animals(self.animals, 1001, 1), number=100) self.assertLess(duration, 1.0) # 100 zapisów < 1s	<input checked="" type="checkbox"/>		1,5
		Testy pamięci memory_profiler	def test_memory_save_animals(self): animals = {str(i): Dog(str(i), f"Animal{i}", 5) for i in range(1000)}  self.data_manager.save_animals(animals, 1001, 1)	<input checked="" type="checkbox"/>		1,5
		Test jakości kodu (flake8, pylint)		<input type="checkbox"/>		1,5
13	Wersjonowanie	Repozytorium GIT	Na github	<input checked="" type="checkbox"/>		1
		Historia commitów	Na github	<input checked="" type="checkbox"/>		1

Nr	Obszar	Wymaganie	KOD		Przyznane pkt	Pkt max
		Link do GitHub	<a href="https://github.com/KosaK27/ProjektJS">https://github.com/KosaK27/ProjektJS</a>	<input checked="" type="checkbox"/>		1
		Opis commitów	Tak	<input checked="" type="checkbox"/>		1
14	Dokumentacja	Plik README.md (cel, autorzy, uruchamianie)	Na github	<input checked="" type="checkbox"/>		1,5
		Przykładowe dane wejściowe i wyjściowe	Na github	<input checked="" type="checkbox"/>		2
		Diagram klas lub struktura modułów	Na github	<input checked="" type="checkbox"/>		2
SUMA						