

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		<input 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): @wraps(func) def wrapper(*args, **kwargs): return func(*args, **kwargs) return wrapper	<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(): try: name = name_entry.get().strip() age = int(age_entry.get().strip()) assert name and age >= 0	<input checked="" type="checkbox"/>	1,5

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

		Zastosowano zakresy klas	self.animals = {}	<input checked="" type="checkbox"/>		1,5
5	Moduły i pakiety	Projekt podzielony na moduły (import, _init_)	<pre>from data_manager import DataManager def __init__(self, root): self.root = root self.root.title("System Schroniska") self.root.geometry("1200x800")</pre>	<input checked="" type="checkbox"/>		2

Nr	Obszar	Wymaganie	KOD		Przyznane pkt	Pkt max
		Własne pakiety/funkcje pomocnicze w osobnych plikach .py	<pre>data_manager.py: import json import csv from datetime import datetime from animal_manager import Dog, Cat, Bird, Rabbit, Hamster, Turtle # Klasa zarządzająca danymi zwierząt i adopcji class DataManager: # Inicjalizacja menedżera danych z nazwami plików def __init__(self, animals_filename, adoptions_filename): self.animals_filename = animals_filename self.adoptions_filename = adoptions_filename self.animals = {} self.adoptions = {}</pre>	<input checked="" type="checkbox"/>		2
6	Obsługa błędów	Obsługa wyjątków (try, except, finally)	<pre>try: with open(self.animals_filename, 'r', encoding='utf-8') as f: data = json.load(f) except (FileNotFoundError, json.JSONDecodeError, KeyError):</pre>	<input checked="" type="checkbox"/>		2
		Użycie assert do testów i walidacji	<pre>assert name and age >= 0, "Imię puste lub wiek ujemny"</pre>	<input checked="" type="checkbox"/>		1,5
7	Łańcuchy znaków	Operacje na stringach (m.in. formatowanie, dzielenie, wyszukiwanie)	<pre>name_query = name_entry.get().strip().lower()</pre>	<input checked="" type="checkbox"/>		2
8	Obsługa plików	Odczyt z plików .txt, .csv, .json, .xml (min. 1)	<pre>with open(file_path, 'r', encoding='utf-8') as f: reader = csv.reader(f, delimiter=',')</pre>	<input checked="" type="checkbox"/>		2

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

		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): self.id = id self.name = name self.age = 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 self.root.title("System Schroniska") self.root.geometry("1200x800")	<input checked="" type="checkbox"/>		2
		Dziedziczenie	class Dog(Animal): #Dziedziczone def get_feeding_status(self): if not self.last_fed: return "Brak danych"	<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, not_vaccinated]	<input checked="" type="checkbox"/>		2
		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): animal = Dog("1", "Reksio", 5) animal.last_fed = datetime.now().strftime("%Y-%m- %d %H:%M:%S") self.assertTrue("godziny temu" in animal.get_feeding_status()) animal.last_fed = None self.assertEqual(animal.get_feedi ng_status(), "Brak danych")	<input checked="" type="checkbox"/>		1,5

		<pre> animal.last_fed = "invalid_date" self.assertEqual(animal.get_feedi ng_status(), "Nieprawidłowy format daty") </pre>			
	Testy funkcjonalne	<pre> def test_add_animal(self): self.app.animals = {} self.app.next_id = 1 animal = Dog("1", "Reksio", 5) animal.is_vaccinated = True animal.admission_date = "2025-06-20 02:00:00" self.app.animals["1"] = animal self.app.data_manager.save_anim als(self.app.animals, 2, 1) self.app.refresh_animals_tree() items = self.app.animals_tree.get_childre n() self.assertEqual(len(items), 1) values = self.app.animals_tree.item(items[0])["values"] self.assertEqual(values[1], "Reksio") self.assertEqual(values[2], 5) self.assertEqual(values[3], "Pies") </pre>	<input checked="" type="checkbox"/>		1,5
	Testy Integracyjne	<pre> def test_export_import_animals(self) : animal = Dog("1", "Reksio", 5) animal.is_vaccinated = True self.data_manager.animals = {"1": animal} self.data_manager.save_animals(self.data_manager.animals, 2, 1) self.data_manager.export_animal s_csv(self.test_csv) self.data_manager.animals = {} self.data_manager.import_animal s_csv(self.test_csv, replace=True) self.assertIn("1", self.data_manager.animals) self.assertEqual(self.data_manag er.animals["1"].name, "Reksio") </pre>	<input checked="" type="checkbox"/>		1,5

			<code>self.assertTrue(self.data_manager.animals["1"].is_vaccinated)</code>			
		Testy graniczne / błędne dane	<code>with open(self.test_csv, "w", encoding="utf-8") as f:</code> <code>f.write("ID;Imię;Wiek;Gatunek;Zaszczepione;Ostatnie karmienie;Data przyjęcia;Status\n")</code> <code>f.write("1;;-1;Nieznany;Tak;invalid_date;invalid_date;Adoptowane\n")</code> <code>errors =</code> <code>self.data_manager.import_animals_csv(self.test_csv, replace=True)</code> <code>self.assertIn("Imię nie może być puste", errors[0])</code>	<input checked="" type="checkbox"/>		1,5
		Testy wydajności (np. czas wykonania, timeit)	<code>@pytest.mark.performance</code> <code>def</code> <code>test_performance_save_animals(self):</code> <code> duration = timeit(lambda:</code> <code>self.dm.save_animals(self.animals, 1001, 1), number=100)</code> <code> self.assertLess(duration, 1.0)</code> <code># 100 zapisów < 1s</code>	<input checked="" type="checkbox"/>		1,5
		Testy pamięci memory_profiler	<code>@profile</code> <code>def</code> <code>test_memory_save_animals(self):</code> <code> animals = {str(i): Dog(str(i), f"Animal{i}", 5) for i in range(1000)}</code> <code>self.data_manager.save_animals(animals, 1001, 1)</code>	<input checked="" type="checkbox"/>		1,5
		Test jakości kodu (flake8, pylint)		<input type="checkbox"/>		1,5
13	Wersjonowanie	Repozytorium GIT	<code>.idea</code> Wgranie projektu na GitHub <code>last week</code> <code>Test</code> Wgranie projektu na GitHub <code>last week</code> <code>dist</code> Dodanie pliku wykonywalnego <code>last week</code> <code>Adopcje.json</code> Wgranie projektu na GitHub <code>last week</code> <code>Adopcje_przyk.csv</code> Przykładowe dane <code>yesterday</code> <code>README.md</code> Aktualizacja README <code>last week</code> <code>Zwierzeta.json</code> Wgranie projektu na GitHub	<input checked="" type="checkbox"/>		1

			last week Zwierzeta_przyk.csv Przykładowe dane last week animal_manager.py Wgranie projektu na GitHub last week data_manager.py Wgranie projektu na GitHub last week decorators.py Wgranie projektu na GitHub last week main.py Wgranie projektu na GitHub last week wykres.png Wgranie projektu na GitHub last week			
		Historia commitów	Dokumentacja i formularz KosaK27pushed 1 commit • 954bfab...4cec4e5 • yesterday Aktualizacja README KosaK27pushed 1 commit • 9ada48f...954bfab • 6 days ago Merge remote-tracking branch 'origin/master' KosaK27pushed 2 commits • c0930b6...9ada48f • 6 days ago Delete Aplikacja directory KosaK27pushed 1 commit • 6ed06ed...c0930b6 • 6 days ago Dodanie pliku wykonywalnego KosaK27pushed 1 commit • d2413eb...6ed06ed • 6 days ago Przykładowe dane KosaK27pushed 1 commit • 0a20cf8...d2413eb • 6 days ago Create README.md KosaK27pushed 1 commit • df1492d...0a20cf8 • 6 days ago Wgranie projektu na GitHub KosaK27created this branch • df1492d • 6 days ago	<input checked="" type="checkbox"/>		1

Nr	Obszar	Wymaganie	KOD		Przyznane pkt	Pkt max
		Link do GitHub	https://github.com/KosaK27/ProjektJS	<input checked="" type="checkbox"/>		1
		Opis commitów	Dokumentacja i formularz KosaK27pushed 1 commit to master • 954bfab...4cec4e5 • yesterday Aktualizacja README KosaK27pushed 1 commit to master • 9ada48f...954bfab • 6 days ago	<input checked="" type="checkbox"/>		1

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

14	Dokumentacja	Plik README.md (cel, autorzy, uruchamianie)	Cel projektu Aplikacja ułatwiająca prowadzenie opieki nad zwierzętami w schronisku Autorzy Jakub Kosatka, Oleksandr Stankevych Uruchamianie Uruchom plik 'Schronisko.exe' znajdujący się w katalogu 'dist'	<input checked="" type="checkbox"/>		1,5
		Przykładowe dane wejściowe i wyjściowe	ID;Imię;Wiek;Gatunek;Zaszczepi one;Ostatnie karmienie;Data przyjęcia;Status 1;Burek;5;Pies;Tak;2025-06-10 09:15:47;2025-01-20 12:30:11;Adoptowane 2;Goku;3;Kot;Nie;2025-06-05 16:22:19;2025-02-15 08:45:33;Adoptowane ID;ID zwierzęcia;Nazwisko;PESEL;Nu mer telefonu;Data adopcji 1;1;Kowalski;80051234567;1234 56789;2025-05-15 14:22:35 2;2;Kowalski;80051234567;1234 56789;2025-05-20 16:10:47	<input checked="" type="checkbox"/>		2
		Diagram klas lub struktura modułów	ProjektJS/ — main.py — animal_manager.py — data_manager.py — decorators.py — test/ — test.py	<input checked="" type="checkbox"/>		2
		SUMA				