



Karta przedmiotu

Nazwa i kod przedmiotu	Programowanie w językach wysokiego poziomu, PG_00067327						
Kierunek studiów	Automatyka, cybernetyka i robotyka						
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu			2026/2027		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			polski		
Semestr studiów	2	Liczba punktów ECTS			2.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Systemów Decyzyjnych i Robotyki						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr Piotr Dalka					
	Prowadzący zajęcia z przedmiotu	dr hab. inż. Michał Czubenko mgr inż. Tymoteusz Byrwa mgr inż. Jakub Kłopotek Głowczewski dr Piotr Dalka mgr inż. Michał Kopczyński					
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	15.0	0.0	0.0	30
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach		Praca własna studenta		RAZEM
	Liczba godzin pracy studenta	30	3.0		17.0		50
Cel przedmiotu	Celem przedmiotu jest kompleksowe przygotowanie studenta do rozwiązywania złożonych problemów inżynierskich poprzez naukę programowania w nowoczesnym, wszechstronnym języku wysokiego poziomu. W trakcie kursu student zdobędzie umiejętność tworzenia czytelnego i efektywnego kodu, przechodząc od podstawowej składni, typów danych i struktur kontrolnych, do zaawansowanych paradygmatów funkcyjnego i obiektowego. Program obejmuje kluczowe aspekty inżynierii oprogramowania, takie jak modularność kodu, zarządzanie zależnościami i środowiskami wirtualnymi, a także obsługę wyjątków i operacje wejścia/wyjścia, w tym serializację danych. Student pozna zaawansowane techniki, takie jak dekoratory, generatory, metaprogramowanie oraz podstawy programowania asynchronicznego, co pozwoli na tworzenie wydajnych i skalowalnych rozwiązań. Uzupełnieniem wiedzy będzie przegląd bibliotek standardowych i zewnętrznych, podstawy tworzenia aplikacji webowych z wykorzystaniem architektury REST oraz umiejętność pisania testów jednostkowych.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	<p>[K6_U12] potrafi analizować działanie elementów, układów i systemów związanych z kierunkiem studiów oraz mierzyć ich parametry i badać charakterystyki techniczne, a także planować i przeprowadzać eksperymenty związane z kierunkiem studiów, w tym pomiary i symulacje komputerowe, oraz interpretować uzyskane wyniki i wyciągać wnioski</p>	<p>Student potrafi analizować działanie komponentów programistycznych elementów, układów i systemów charakterystycznych dla automatyki i robotyki, planować i przeprowadzać symulacje komputerowe oraz testy funkcjonalne, a także interpretować uzyskane wyniki i wyciągać technicznie uzasadnione wnioski. Umie obsługiwać środowiska programistyczne i debuggera, identyfikować i analizować błędy kompilatora lub interpretera, a także projektować oraz implementować testy jednostkowe i funkcjonalne w celu zapewnienia poprawności i niezawodności tworzonego oprogramowania.</p>	<p>[SU4] Ocena umiejętności korzystania z metod i narzędzi</p>
	<p>[K6_U09] potrafi dokonać krytycznej analizy sposobu funkcjonowania istniejących rozwiązań technicznych związanych z kierunkiem studiów i ocenić te rozwiązania, a także wykorzystać zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską doświadczenie związane z utrzymaniem urządzeń, obiektów i systemów technicznych typowych dla kierunku studiów</p>	<p>Student potrafi dokonać krytycznej analizy działania i struktury istniejących rozwiązań programistycznych i technicznych stosowanych w systemach automatyki i robotyki. Umie ocenić ich efektywność, czytelność, niezawodność oraz skalowalność, a także wskazać ograniczenia i możliwości usprawnień. Wykorzystuje zdobyte doświadczenie w zakresie utrzymania i testowania systemów programowalnych – zarówno na poziomie aplikacyjnym, jak i sprzętowym – w celu formułowania rekomendacji oraz projektowania nowych rozwiązań zgodnych z wymaganiami praktyki inżynierskiej.</p>	<p>[SU3] Ocena umiejętności wykorzystania wiedzy uzyskanej w ramach przedmiotu</p>
	<p>[K6_W04] zna i rozumie w zaawansowanym stopniu zasady, metody i techniki programowania oraz zasady tworzenia oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, specyficznych dla kierunku studiów, a także organizację pracy systemów wykorzystujących komputery lub te urządzenia</p>	<p>Student zna i rozumie w zaawansowanym stopniu zasady projektowania i implementacji oprogramowania w językach wysokiego poziomu, ze szczególnym uwzględnieniem metod i technik stosowanych w automatyce i robotyce. Posiada wiedzę na temat paradygmatów programowania (proceduralnego, obiektowego, funkcyjnego), modularności kodu, zarządzania zależnościami, obsługi błędów i operacji wejścia-wyjścia, a także tworzenia kodu dla systemów opartych na mikroprocesorach, mikrokontrolerach i układach programowalnych. Rozumie organizację pracy systemów komputerowych oraz architekturę środowisk wykonawczych używanych w systemach wbudowanych i robotycznych.</p>	<p>[SW1] Ocena wiedzy faktograficznej</p>

	<table border="1"> <tr> <th>Efekt kierunkowy</th> <th>Efekt z przedmiotu</th> <th>Sposób weryfikacji i oceny efektu</th> </tr> <tr> <td>[K6_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów</td> <td>Student potrafi wykorzystywać zdobytą wiedzę z zakresu technik programowania do tworzenia przejrzystego i efektywnego kodu, dobierając odpowiednie narzędzia oraz paradygmaty programistyczne (proceduralne, obiektowe, funkcyjne) w zależności od charakterystyki problemu. Umie zastosować nowoczesne środowiska programistyczne oraz biblioteki wspierające rozwój oprogramowania dla systemów komputerowych wykorzystywanych w automatyce i robotyce, a także analizować i implementować algorytmy zgodnie z wymaganiami funkcjonalnymi danego systemu.</td> <td>[SU1] Ocena realizacji zadania</td> </tr> </table>	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu	[K6_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów	Student potrafi wykorzystywać zdobytą wiedzę z zakresu technik programowania do tworzenia przejrzystego i efektywnego kodu, dobierając odpowiednie narzędzia oraz paradygmaty programistyczne (proceduralne, obiektowe, funkcyjne) w zależności od charakterystyki problemu. Umie zastosować nowoczesne środowiska programistyczne oraz biblioteki wspierające rozwój oprogramowania dla systemów komputerowych wykorzystywanych w automatyce i robotyce, a także analizować i implementować algorytmy zgodnie z wymaganiami funkcjonalnymi danego systemu.	[SU1] Ocena realizacji zadania				
Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu									
[K6_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów	Student potrafi wykorzystywać zdobytą wiedzę z zakresu technik programowania do tworzenia przejrzystego i efektywnego kodu, dobierając odpowiednie narzędzia oraz paradygmaty programistyczne (proceduralne, obiektowe, funkcyjne) w zależności od charakterystyki problemu. Umie zastosować nowoczesne środowiska programistyczne oraz biblioteki wspierające rozwój oprogramowania dla systemów komputerowych wykorzystywanych w automatyce i robotyce, a także analizować i implementować algorytmy zgodnie z wymaganiami funkcjonalnymi danego systemu.	[SU1] Ocena realizacji zadania									
Treści przedmiotu	<p>Treści przedmiotu - wykład</p> <p>Wykład</p> <p>W ramach przedmiotu prowadzonych będzie 15 wykładów powiązanych z następującymi tematami (lista tematów nie jest zamknięta):</p> <ul style="list-style-type: none"> <li>• cechy szczególne języka</li> <li>• składnia języka, podstawowe typy danych oraz kolekcje</li> <li>• kontrola przepływu</li> <li>• programowanie funkcyjne</li> <li>• organizacja kodu, w tym moduły i pakiety</li> <li>• zarządzanie środowiskiem wirtualnym i menadżery pakietów</li> <li>• specyficzne elementy języka: dekoratory, generatory, menadżer kontekstu i obsługa wyjątków</li> <li>• operacje wejścia/wyjścia oraz serializacja i deserializacja danych do/z popularnych formatów takich jak CSV, JSON, XML oraz YAML</li> <li>• programowanie obiektowe</li> <li>• implementacja wybranych wzorców projektowych</li> <li>• przetwarzanie współbieżne i równoległe oraz optymalne wykonywanie operacji ograniczonych wydajnością procesora oraz operacji wejścia/wyjścia</li> <li>• programowanie asynchroniczne</li> <li>• zaawansowane elementy języka: deskryptory, protokoły, metody magiczne, metaklasy</li> <li>• przegląd biblioteki standardowej oraz popularnych bibliotek zewnętrznych</li> <li>• frameworki webowe oraz podstawy komunikacji RESTowej</li> <li>• tworzenie testów jednostkowych</li> </ul> <p>Laboratorium:</p> <ul style="list-style-type: none"> <li>• Lista ćwiczeń nie jest zamknięta:</li> <li>• Podstawy języka Python typy danych (liczby, listy, słowniki, krotki), operacje logiczne i arytmetyczne, instrukcje warunkowe i pętle, wprowadzenie do pracy z danymi z sensorów.</li> <li>• Funkcje i modularność definiowanie funkcji, parametry, wartości domyślne, zwracanie danych, przetwarzanie i filtracja sygnałów z czujników.</li> <li>• Programowanie obiektowe tworzenie klas i obiektów, dziedziczenie, metody specjalne, modelowanie komponentów robotycznych (np. czujnik, robot mobilny, przeszkoda).</li> <li>• Operacje na plikach i strumieniach, logger zapis i odczyt danych (.csv, .txt), przechowywanie logów i konfiguracji systemów.</li> <li>• Generatory i iteratory przetwarzanie danych w czasie rzeczywistym z wykorzystaniem yield, analiza strumieni pomiarowych przy użyciu map, filter, zip.</li> <li>• Programowanie asynchroniczne i środowiska asyncio, symulacja równoległych odczytów sensorów, zarządzanie środowiskiem (venv, pip, poetry, uv, requirements.txt).</li> <li>• Symulacja i wizualizacja ruchu robota obliczenia trajektorii w 2D i 3D (numpy) wizualizacja ruchu z użyciem matplotlib, analiza pozycji i orientacji robota w przestrzeni.</li> </ul>										
Wymagania wstępne i dodatkowe	<ul style="list-style-type: none"> <li>• ma podstawową wiedzę z zakresu matematyki, obejmującą algebrę oraz logikę</li> <li>• zna podstawowe pojęcia związane z programowaniem funkcyjnym i obiektowym, takie jak na przykład zmienna, funkcja, klasa, dziedziczenie, pętla, warunek</li> <li>• zna dobre praktyki związane z tworzeniem oprogramowania i podstawowe wzorce projektowe</li> <li>• wie jak wyglądają i czym się charakteryzują podstawowe formaty plików tekstowych, takie jak CSV, XML lub JSON</li> <li>• zna podstawy protokołu HTTP</li> </ul>										
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	<table border="1"> <tr> <th>Sposób oceniania (składowe)</th> <th>Próg zaliczeniowy</th> <th>Składowa oceny końcowej</th> </tr> <tr> <td>Kolokwium zaliczeniowe</td> <td>60.0%</td> <td>50.0%</td> </tr> <tr> <td>Laboratoria</td> <td>60.0%</td> <td>50.0%</td> </tr> </table>		Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej	Kolokwium zaliczeniowe	60.0%	50.0%	Laboratoria	60.0%	50.0%
Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej									
Kolokwium zaliczeniowe	60.0%	50.0%									
Laboratoria	60.0%	50.0%									
Zalecana lista lektur	<p>Podstawowa lista lektur</p> <ul style="list-style-type: none"> <li>• Mark Lutz. <i>Python. Wprowadzenie</i>. Helion, 2022. Print.</li> <li>• Allen B. Downey. <i>Myśl w języku Python</i>. Helion, 2025. Print.</li> </ul>										

	Uzupełniająca lista lektur	• Luciano Ramalho. <i>Fluent Python. 2nd Edition, 2022</i> , Helion.
	Adresy eZasobów	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<ul style="list-style-type: none"> <li>• Które typy danych są mutowane?</li> <li>• Jakie są rodzaje argumentów przyjmowane przez funkcję?</li> <li>• Jaka jest różnica między modułem a pakietem?</li> <li>• Jakie elementy języka należy użyć do skutecznego czyszczenia zasobów przy kończeniu danego zadania?</li> <li>• Do jakich zastosowań wskazane jest wykorzystać generator, a kiedy należy unikać jego używania?</li> <li>• Za pomocą jakich mechanizmów języka optymalnie zaimplementować wykonywanie operacji obciążających procesor?</li> </ul>	
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy	

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.