



Karta przedmiotu

Nazwa i kod przedmiotu	Programowanie w językach wysokiego poziomu II, PG_00067435						
Kierunek studiów	Automatyka, cybernetyka i robotyka						
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu			2027/2028		
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	2	Język wykładowy			polski		
Semestr studiów	3	Liczba punktów ECTS			3.0		
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 hab. inż. Michał Czubenko					
	Prowadzący zajęcia z przedmiotu	dr hab. inż. Michał Czubenko					
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	0.0	0.0	15.0	0.0	0.0	15
	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	15	4.0	56.0	75		
Cel przedmiotu	Celem przedmiotu jest zapoznanie studentów z zaawansowanymi technikami programowania w języku wysokiego poziomu, który łączy przejrzystość składni z dużą elastycznością w tworzeniu nowoczesnych aplikacji. W ramach kursu studenci nauczą się projektowania i implementacji aplikacji wykorzystujących różne paradygmaty, ze szczególnym uwzględnieniem programowania obiektowego oraz komponentów graficznych przy użyciu popularnych frameworków okienkowych (Qt lub GTK). Omawiane będą również zagadnienia związane z przetwarzaniem danych, takie jak serializacja i deserializacja. Duży nacisk położony zostanie na opanowanie zaawansowanych elementów języka, takich jak deskryptory, dekoratory, protokoły, funkcje jako obiekty pierwszoklasowe, klasy metapoziomu oraz techniki metaprogramowania, które pozwalają pisać bardziej elastyczny i wielokrotnie używalny kod. Poruszone zostaną aspekty wzorców projektowych (fabryki, singletony, adaptory).						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[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	potrafi obsługiwać debuggery, analizować błędy interpretera/ kompilatora oraz programować przy pomocy użycia testów	[SU2] Ocena umiejętności analizy informacji
	[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	potrafi obsługiwać metody powiązane z metaprogramowaniem oraz zna podstawowe wzorce projektowe	[SW1] Ocena wiedzy faktograficznej
	[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	potrafi posługiwać się językiem programowania wysokiego poziomu w stopniu pozwalającym na użycie zaawansowanych metod programowania	[SU1] Ocena realizacji zadania
Treści przedmiotu	<p>Treści przedmiotu - laboratoria</p> <p>W ramach przedmiotu prowadzonych będzie 7 ćwiczeń komputerowych powiązanych z następującymi tematami (lista tematów nie jest zamknięta):</p> <ol style="list-style-type: none"> <li>1. Programowanie obiektowe i struktura aplikacji (utrwalenie aspektów obiektowych, klasy, dziedziczenie, hermetyzacja, metody specjalne, organizacja kodu)</li> <li>2. Przetwarzanie danych serializacja i deserializacja (obsługa formatów takich jak JSON, CSV, Pickle, wczytywanie i zapisywanie obiektów, obsługa błędów i walidacja)</li> <li>3. Programowanie graficzne (proste okienka, obsługujące logikę, sygnały sloty, obsługa zdarzeń, komunikacja między komponentami, aspekty blokowania interfejsu)</li> <li>4. Zaawansowane elementy języka (tworzenie klas metapoziomu, automatyczne modyfikowanie klas, deskryptory)</li> <li>5. Aplikacje webowe (struktura projektów webowych, routing i obsługa HTTP, komunikacja między frontendem a backendem)</li> <li>6. REST API i testowanie (projektowanie prostego serwera REST, testy jednostkowe)</li> <li>7. Wzorce projektowe (singleton, fabryka, strategia, dekorator, adapter i inne)</li> </ol>		
Wymagania wstępne i dodatkowe	<ul style="list-style-type: none"> <li>• ma podstawową wiedzę z zakresu matematyki, obejmującą analizę matematyczną, algebrę oraz zasady geometrii przestrzennej</li> <li>• potrafi programować w języku cpp oraz wybranym języku wysokiego poziomu</li> <li>• zna podstawowe koncepcje składni języków programowania</li> <li>• zna podstawy programowania obiektowego oraz aspekty programowania funkcyjnego i strukturalnego</li> </ul>		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Ćwiczenia Laboratoryjne	60.0%	100.0%
Zalecana lista lektur	Podstawowa lista lektur	<p>Allen B. Downey. <i>Myśl w języku Python</i>. Helion, 2025. Print.</p> <p>Houlahan, Padraig. <i>Prototyping Python Dashboards for Scientists and Engineers: Build and Deploy a Complete Dashboard with Python</i>. 1st ed. Berkeley, CA: Apress L. P, 2024. Web.</p>	

	Uzupełniająca lista lektur	Lanaro, G., Nguyen, Q., & Kasampalis, S. (2019). <i>Advanced Python Programming: Build high performance, concurrent, and multi-threaded apps with Python using proven design patterns</i> . Packt Publishing Ltd.
	Adresy eZasobów	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<ol style="list-style-type: none"> <li>1. Modelowanie systemu (np. biblioteki lub systemu rezerwacyjnego).</li> <li>2. Implementacja klasy zapisującej dane użytkowników do pliku i odczytującej je z pliku.</li> <li>3. Aplikacja z graficznym formularzem wejściowym i walidacją danych.</li> <li>4. Przykład klasy, która dynamicznie tworzy metody.</li> <li>5. Aplikacja webowa z prostym panelem administracyjnym.</li> <li>6. Interfejs do zarządzania obiektami przez API.</li> <li>7. Implementacja klasy rejestrującej i zarządzającej obiektami (Fabryka).</li> </ol>	
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy	

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.