

Karta przedmiotu

Nazwa i kod przedmiotu	Języki programowania, PG_00047657						
Kierunek studiów	Informatyka						
Data rozpoczęcia studiów	październik 2025 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	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 Algorytmów i Modelowania Systemów						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr hab. inż. Piotr Mironowicz				
	Prowadzący zajęcia z przedmiotu		mgr inż. Tomasz Goluch dr hab. inż. Piotr Mironowicz dr hab. inż. Jan Daciuk dr Magdalena Godlewska prof. dr hab. inż. Bogdan Wiszniewski				
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	0.0	15.0	0.0	30
	W tym liczba godzin zajęć na odległość: 0.0						
	Adresy kursu na platformie eNauczanie: Moodle ID: 1368 Języki programowania 2025/26 https://enauczanie.pg.edu.pl/2025/course/view.php?id=1368						
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		42.0	75
Cel przedmiotu	Student poznał popularne paradygmaty programowania i nauczył się z nich korzystać.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K6_U03] potrafi zaprojektować, zgodnie z zadaną specyfikacją, oraz wykonać typowe dla kierunku studiów proste urządzenie, obiekt, system lub zrealizować proces, używając odpowiednio dobranych metod, technik, narzędzi i materiałów, korzystając ze standardów i norm inżynierskich, stosując właściwe dla kierunków studiów technologie i wykorzystując doświadczenie zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską	Umie zaprogramować rozwiązanie wyspecyfikowanego problemu stosując specyfikę wybranego paradygmatu programowania. Rozpoznaje, które z podejść modelowania będą najwłaściwsze dla zadanego zagadnienia.	[SU1] Ocena realizacji zadania
	[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	Jest zapoznany z wszystkimi istotnymi paradygmatami programowania oraz ich znaczenia dla zasad tworzenia oprogramowania. Rozumie, jaka jest zależność między językami wysokiego poziomu a specyfiką mikroprocesorów.	[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	Studenci potrafią stosować odpowiednie paradygmaty programowania (imperatywne, obiektowe, funkcyjne, logiczne) w zależności od wymagań projektowych i charakterystyki problemu. Uczą się dobierać odpowiednie języki i narzędzia programistyczne do implementacji oprogramowania, zarówno w kontekście aplikacji użytkowych, jak i systemów osadzonych. Dzięki realizacji projektów zespołowych rozwijają umiejętność praktycznego wykorzystania zdobytej wiedzy w rzeczywistych scenariuszach programistycznych.	[SU1] Ocena realizacji zadania [SU4] Ocena umiejętności korzystania z metod i narzędzi
	[K6_W10] zna i rozumie w zaawansowanym stopniu parametry, funkcje oraz metody analizy, projektowania i optymalizacji układów i systemów elektronicznych, definicje błędów i niepewności pomiaru, metody pomiarowe, a w tym pomiarów czasu, częstotliwości i fazy, właściwości przetworników, oraz metody cyfrowego przetwarzania sygnałów, a także podstawowe procesy zachodzące w cyklu życia urządzeń, obiektów i systemów technicznych oraz metody wspomagania procesów i funkcji, specyficzne dla kierunku studiów	Student zna i rozumie paradygmaty programowania stosowane w projektowaniu systemów informatycznych i technicznych, w szczególności podejścia imperatywne (strukturalne i obiektowe) oraz deklaratywne (funkcyjne i logiczne), wraz z metodami abstrakcji, parametryzacji oraz modularizacji. Potrafi analizować i projektować rozwiązania programistyczne z wykorzystaniem wybranych języków (m.in. Ada, Smalltalk, Haskell, Prolog), stosując odpowiednie metody reprezentacji problemów i optymalizacji algorytmów w kontekście modelowania systemów technicznych.	[SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym

Treści przedmiotu	Treści przedmiotu - wykład 1. Programowanie proceduralne. 2. Składnia liniowa. FORTRAN. 3. Rekordy aktywacyjne i podprogramy. 4. Rekurencyjne wywołanie procedur. 5. Składnia blokowa. Abstrakcja przepływu sterowania. 6. Wiązanie nazw z obiektami. Zasięg wiązania .7. Parametry wywołania procedury. Tryby przekazu parametrów. 8. Rekordy aktywacyjne dla języków z rekurencją. 9. Wywołania statyczne i dynamiczne. ALGOL. PASCAL. 10. Ograniczenia języków blokowych. 11. Abstrakcja danych i ochrona dostępu. 12. Modularyzacja. MODULA-2. ADA83, ADA95. 13. Sytuacje wyjątkowe. Modele obsługi wyjątków. 14. Współprogramy i procedury współbieżne. Rendezvous. 15. Programowanie obiektowe. Obiekty, klasy, hierarchie.1 6. Typy dynamiczne. Polimorfizm. SMALLTALK. C++. 17. Rekurencyjne interpretowanie poleceń. 18. Przekształcenia symboliczne. Rekurencja ogonowa.19. LISP. Atomy i listy. 20. Programowanie funkcjonalne. Haskell, XSL. 21. Redukcja, filtrowanie i rzutowanie. 22. Rachunek lambda. 23. Zarządzanie pamięcią w systemach LISP24. Programowanie w logice. PROLOG		
	Treści przedmiotu - projekt 1. Programowanie w języku ADA 2. Programowanie w języku Smalltalk 3. Programowanie w języku Haskell 4. Programowanie w języku Prolog		
Wymagania wstępne i dodatkowe			
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Projekt	50.0%	60.0%
	Kolokwia	50.0%	40.0%
Zalecana lista lektur	Podstawowa lista lektur	S. Mangano: XSLT receptury, wyd.2, Helion 2007Cincom Smalltalk Downloads, http://www.cincomsmalltalk.com/SAXON - The XSLT and XQuery Processor, http://saxon.sourceforge.net/W.F.Clocksini,W.F.,Mellish,C.S.:PrologProgramowanie.Helion2003 Ada Programming, http://en.wikibooks.org/wiki/AdaSWI-Prolog_downloads , www.swi-prolog.org/download.html ADA Core, the GNAT Pro Company, http://www.adacore.com/home , https://libre.adacore.com/D . S. Touretzky: Common Lisp: A Gentle Introduction to Symbolic Computation, http://www.cs.cmu.edu/~dst/LispBook/Z . Huzar, Z. Fryźlewicz, I. Dubielewicz, B. Hnatk: Ada 95, Helion 1998Polski serwis języka Smalltalk, http://www.objectspace.net/	
	Uzupełniająca lista lektur	http://en.wikipedia.org/wiki/Programming_paradigm	
		Adresy eZasobów	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.