



Karta przedmiotu

Nazwa i kod przedmiotu	Języki programowania, PG_00047657						
Kierunek studiów	Informatyka						
Data rozpoczęcia studiów	październik 2021 r.	Rok akademicki realizacji przedmiotu			2022/2023		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów		
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	Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Algorytmów i Modelowania Systemów						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr inż. Piotr Mironowicz					
	Prowadzący zajęcia z przedmiotu	dr Magdalena Godlewska dr inż. Piotr Mironowicz					
Formy zajęć i metody nauczania	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						
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_W05] zna i rozumie w zaawansowanym stopniu metody wspomaganie procesów i funkcji, specyficzne dla kierunku studiów	Zna koncepcyjnie i historycznie istotne języki programowania, takie jak Modula, Ada, Smalltalk, Haskell, Prolog. Jest świadomy tego, z jakich problemów wynikły poszczególne rozwiązania.	[SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym
	[K6_U41] potrafi wytwarzać, testować lub oceniać oprogramowanie, wykorzystując nowoczesne platformy, narzędzia, języki i paradygmaty programowania różnych poziomów, a także posługiwać się pakietami oprogramowania wspierającymi naukowo-badawcze i biznesowe procesy decyzyjne oraz pracę zespołową	Umie programować w językach paradygmatu proceduralnego i obiektowego (Modula, Smalltalk), funkcyjnego (Haskell) i w logice (Prolog). Zna środowiska i pakiety wspomagające programowanie.	[SU4] Ocena umiejętności korzystania z metod i narzędzi
	[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
Treści przedmiotu	<ol style="list-style-type: none"> 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. 16. 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 LISP 24. Programowanie w logice. PROLOG 		
Wymagania wstępne i dodatkowe			

Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Kolokwia	50.0%	40.0%
	Projekt	50.0%	60.0%
Zalecana lista lektur	Podstawowa lista lektur	<p>S. Mangano: XSLT receptury, wyd.2, Helion 2007 Cincom Smalltalk Downloads, http://www.cincomsmalltalk.com/ SAXON - The XSLT and XQuery Processor, http://saxon.sourceforge.net/ W.F. Clocksin, W.F., Mellish, C.S.: Prolog Programowanie. Helion 2003 Ada Programming, http://en.wikibooks.org/wiki/Ada SWI-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 1998 Polski serwis języka Smalltalk, http://www.objectspace.net/</p>	
	Uzupełniająca lista lektur	http://en.wikipedia.org/wiki/Programming_paradigm	
	Adresy eZasobów	Adresy na platformie eNauczanie: Języki Programowania - 2022/23 - Moodle ID: 25342 https://enauczanie.pg.edu.pl/moodle/course/view.php?id=25342	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		