



Karta przedmiotu

Nazwa i kod przedmiotu	Projektowanie skalowalnych systemów obiektowych, PG_00047967						
Kierunek studiów	Informatyka						
Data rozpoczęcia studiów	październik 2024 r.	Rok akademicki realizacji przedmiotu			2027/2028		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	4	Język wykładowy			polski		
Semestr studiów	7	Liczba punktów ECTS			4.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			egzamin		
Jednostka prowadząca	Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Inżynierii Oprogramowania						
Imię i nazwisko wykładowcy (wykładowców)	Od odpowiedzialny za przedmiot	dr inż. Michał Wróbel					
	Prowadzący zajęcia z przedmiotu	dr inż. Michał Wróbel					
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	15.0	15.0	0.0	45
	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	45	4.0		51.0		100
Cel przedmiotu	Celem przedmiotu jest dostarczenie kompleksowej wiedzy pozwalającej projektować i implementować skalowane systemy informatyczne z zastosowaniem paradygmatu obiektowego. W ramach przedmiotu studenci zapoznają się z zasadami SOLID oraz ze wzorcami projektowymi. Ponadto zostaną wprowadzone elementy programowania funkcyjnego. Omówione zostaną również popularne architektury zorientowane na usługi (SOA).						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K6_W10] zna i rozumie w zaawansowanym stopniu podstawowe procesy zachodzące w cyklu życia urządzeń, obiektów i systemów technicznych oraz metody wspomaganie procesów i funkcji, specyficzne dla kierunku studiów	Studenci są w stanie projektować i implementować systemy oprogramowania w taki sposób, aby można je było łatwo utrzymywać i rozwijać.	[SW1] Ocena wiedzy faktograficznej
	[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	Student potrafi wykorzystać elementy paradygmatu funkcyjnego do zwiększenia niezawodności implementowanych systemów informatycznych.	[SU1] Ocena realizacji zadania
	[K6_U08] potrafi przy identyfikacji i formułowaniu specyfikacji zadań inżynierskich związanych z kierunkiem studiów oraz ich rozwiązywaniu: – wykorzystywać metody analityczne, symulacyjne i eksperymentalne, – dostrzegać ich aspekty systemowe i pozatechniczne, – dokonać wstępnej oceny ekonomicznej proponowanych rozwiązań i podejmowanych działań inżynierskich	Student rozumie i umie w praktyce zastosować dobre praktyki projektowania obiektowego SOLID.	[SU3] Ocena umiejętności wykorzystania wiedzy uzyskanej w ramach przedmiotu
[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	Student potrafi: projektować łatwo modyfikowalne systemy przy użyciu wzorców projektowych.	[SW1] Ocena wiedzy faktograficznej	
Treści przedmiotu	<ol style="list-style-type: none"> <li>1. Paradygmat programowania obiektowego</li> <li>2. Zasady SOLID</li> <li>3. Kreatywne wzorce projektowe</li> <li>4. Strukturalne wzorce projektowe</li> <li>5. Behawioralne wzorce projektowe</li> <li>6. Elementy programowania funkcyjnego</li> <li>7. Service-oriented architecture</li> </ol>		
Wymagania wstępne i dodatkowe			
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Egzamin końcowy	50.0%	35.0%
	Laboratorium	50.0%	35.0%
	Projekt	50.0%	30.0%
Zalecana lista lektur	Podstawowa lista lektur	<ol style="list-style-type: none"> <li>1. Booch et al.: Object-Oriented Analysis and Design, with Applications. Addison-Wesley, 2007</li> <li>2. Tegarden et al.: Systems Analysis and Design with UML. Wiley, 2012</li> <li>3. Gamma et al.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston, MA, 1995</li> <li>4. Maciaszek: Requirements Analysis and Systems Design. Addison-Wesley, 2007</li> <li>5. Schach: Object-Oriented &amp; Classical Software Engineering. McGraw Hill, New York, 2007</li> </ol>	

	Uzupełniająca lista lektur	<ol style="list-style-type: none"> <li>1. Fowler: UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley, 2004</li> <li>2. Booch et al.: The Unified Modeling Language User Guide. Addison-Wesley, 2005</li> <li>3. Górski: Inżynieria oprogramowania w projekcie informatycznym. Mikom, 2000</li> <li>4. Wrycza et al.: Język UML 2.0 w modelowaniu systemów informatycznych. Helion, 2006</li> <li>5. Martin &amp; Odell: Podstawy metod obiektowych. WNT, 1997</li> </ol>
	Adresy eZasobów	Uzupełniająca Adresy na platformie eNauczanie:
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania		
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy	

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.