



Karta przedmiotu

Nazwa i kod przedmiotu	Projektowanie skalowalnych systemów obiektowych, PG_00047967						
Kierunek studiów	Informatyka						
Data rozpoczęcia studiów	październik 2021 r.	Rok akademicki realizacji przedmiotu			2024/2025		
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)	Odpowiedzialny za przedmiot	dr Adam Przybyłek					
	Prowadzący zajęcia z przedmiotu	dr Adam Przybyłek					
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 zapoznanie studentów z dobrymi praktykami projektowania obiektowego z wykorzystaniem wzorców projektowych oraz zasad SOLID. Dodatkowo studenci dowiedzą się o zagadnieniach przecinających, trudnościach implementacji takich zagadnień w językach obiektowych i sposobie radzenia sobie z nimi przy wykorzystaniu paradygmatu aspektowego. Omówione zostanie także przetwarzanie rozproszone z wykorzystaniem Java RMI, MapReduce oraz Storm.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[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
	[K6_U43] potrafi analizować dane oraz formułować, stosować i oceniać właściwe modele formalne i algorytmy rozwiązywania problemów w zakresie systemów i aplikacji informacyjnych	Student potrafi analizować masowe dane przy użyciu MapReduce.	[SU2] Ocena umiejętności analizy informacji
	[K6_W05] zna i rozumie w zaawansowanym stopniu metody wspomagania procesów i funkcji, specyficzne dla kierunku studiów	Student potrafi: tworzyć modułarne oprogramowanie zgodnie z zasadami obiektowości, wykorzystać testy jednostkowe do weryfikacji poprawności programu.	[SW1] Ocena wiedzy faktograficznej
	[K6_U08] potrafi przy identyfikacji i formułowaniu specyfikacji zadań inżynierskich związanych z kierunkiem studiów oraz ich rozwiązywaniu: – wykorzystać metody analityczne, symulacyjne i eksperymentalne, – dostrzegać ich aspekty systemowe i pozatechniczne, – dokonać wstępnej oceny ekonomicznej proponowanych rozwiązań i podejmowanych działań inżynierskich	Student potrafi użyć MapReduce do analizy Big Data.	[SU1] Ocena realizacji zadania
[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 AspectJ do implementacji zagadnień przecinających lub wchłonięcia nieoczekiwanych zmian, które powstają w związku z ewolucją wymagań biznesowych.	[SU1] Ocena realizacji zadania	
Treści przedmiotu	<ol style="list-style-type: none"> 1. Podstawowe zasady paradygmatu obiektowego – 2h 2. Obiektowa analiza i projektowanie – 1h 3. Wzorce projektowe – 3h 4. Obiektowość bazująca na prototypach – 1h 5. Paradygmat aspektowy – 3h 6. Test-Driven Development – 1h 7. Przetwarzanie rozproszone – 2h 8. Chmury obliczeniowe – 2h 		
Wymagania wstępne i dodatkowe			
Sposoby i kryteria oceniania osiąganych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Projekt	50.0%	50.0%
	Laborki	50.0%	15.0%
	Egzamin końcowy	50.0%	35.0%
Zalecana lista lektur	Podstawowa lista lektur	<ol style="list-style-type: none"> 1. Booch et al.: Object-Oriented Analysis and Design, with Applications. Addison-Wesley, 2007 2. Tegarden et al.: Systems Analysis and Design with UML. Wiley, 2012 3. Gamma et al.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston, MA, 1995 4. Maciaszek: Requirements Analysis and Systems Design. Addison-Wesley, 2007 5. Schach: Object-Oriented & Classical Software Engineering. McGraw Hill, New York, 2007 	

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