



Karta przedmiotu

Nazwa i kod przedmiotu	Inżynieria oprogramowania, PG_00051071						
Kierunek studiów	Fizyka Techniczna						
Data rozpoczęcia studiów	październik 2022 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	3	Język wykładowy			polski		
Semestr studiów	6	Liczba punktów ECTS			7.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydział Fizyki Technicznej i Matematyki Stosowanej -> Instytut Fizyki i Informatyki Stosowanej -> Zakład Fizyki Teoretycznej i Informatyki Kwantowej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr hab. inż. Marta Łabuda					
	Prowadzący zajęcia z przedmiotu	dr hab. inż. Marta Łabuda					
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	0.0	45.0	0.0	75
	W tym liczba godzin zajęć na odległość: 0.0						
	Dodatkowe informacje: Wykłady i zajęcia laboratoryjne.						
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	75	10.0	90.0	175		
Cel przedmiotu	Student wie, czym wyróżnia się inżynieria oprogramowania Student zna różne modele procesu tworzenia oprogramowania Student zna zasady zbierania i dokumentowania wymagań Student potrafi modelować system przy użyciu języka UML Student zna podstawy zarządzania przedsięwzięciem informatycznym. Student zna podstawowe architektury systemów informatycznych.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K6_U02] Potrafi analizować i rozwiązywać proste problemy naukowe i techniczne w oparciu o posiadaną wiedzę, stosując metody analityczne, numeryczne, symulacyjne i eksperymentalne.	Student realizuje zadania projektowe dotyczące inżynierii oprogramowania.	[SU2] Ocena umiejętności analizy informacji [SU1] Ocena realizacji zadania
	[K6_U03] Posiada umiejętność programowania w wybranym języku oraz stosowania podstawowych pakietów oprogramowania.	Student zna różne języki oprogramowania oraz narzędzia i nowoczesne technologie wspierające projektowanie architektury systemów informatycznych	[SU1] Ocena realizacji zadania
	[K6_W05] Posiada podstawową wiedzę w zakresie metodyki i technik programowania oraz wykorzystywania wybranych narzędzi informatycznych w fizyce i technice.	Student wie, czym wyróżnia się inżynieria oprogramowania Student zna różne modele procesu tworzenia oprogramowania Student zna zasady zbierania i dokumentowania wymagań Student potrafi modelować system przy użyciu języka UML Student zna podstawy zarządzania przedsięwzięciem informatycznym Student zna podstawowe architektury systemów informatycznych.	[SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym

Treści przedmiotu	<p>WYKŁAD</p> <p>1. Wprowadzenie do inżynierii oprogramowania. Właściwości inżynierii systemów komputerowych. Systemy i ich środowiska. Modelowanie pojęciowe systemu.</p> <p>2. Planowanie projektu. Projekt informatyczny; podstawowe charakterystyki, pojęcia, udziałowcy projektu; cykl życia i zakres projektu. Planowanie zadań. Identyfikacja problemu; Wzbogacony wizerunek (Rich Picture). Decyzje strategiczne i wizja rozwiązania.</p> <p>3. Studium Wykonalności projektu informatycznego. Cele, płaszczyzny oceny techniczna, ekonomiczna, organizacyjna i prawna; ryzyko przedsięwzięcia.</p> <p>4. Określenie i analiza wymagań, proces inżynierii wymagań. Wymagania stawiane oprogramowaniu i ich dokumentacja. Cechy dobrego wymagania. Metody pozyskiwania wymagań. Podział i klasyfikacja wymagań. Zatwierdzenie wymagań. Zarządzanie wymaganiami.</p> <p>5. Strategie i procesy prowadzenia projektów informatycznych; tradycyjne (model kaskadowy, model V, prototypowanie, przyrostowy, spiralny) i nowoczesne cykle wytwarzania oprogramowania (reuse i komponentowość), MDA, ponowna inżynieria.</p> <p>6. Zwinne metodyki wytwarzania oprogramowania. Programowanie ekstremalne. SCRUM: procesy, artefakty, role. Dobór strategii prowadzenia projektu.</p> <p>7. Język UML. Narzędzia CASE</p> <p>8. Projektowanie architektoniczne: przedstawienie koncepcji architektury oprogramowania i projektowania architektonicznego. Strukturalizacja systemu, modele sterowania, rozkład na moduły, architektury charakterystyczne dla różnych dziedzin. Architektury systemów rozproszonych. Architektury wieloprotocole, klient-serwer, obiektów rozproszonych.</p> <p>9. Projektowanie obiektowe: przedstawienie podejścia do oprogramowania, przy którym projekt ma strukturę oddziałujących na siebie obiektów. Obiekty i klasy obiektów, procesy projektowania obiektowego, ewolucja projektu.</p> <p>10. Optymalizacja projektu dla specyfiki produktu. Projektowanie systemów rozproszonych, krytycznych, czasu rzeczywistego i wielokrotnego użytku.</p> <p>11. Outsourcing w inżynierii oprogramowania.</p> <p>LABORATORIUM W ramach zajęć laboratoryjnych studenci wykonują ćwiczenia składające się na podstawowe kroki identyfikacji i analizy wymagań oraz modelowania obiektowego, przy wykorzystaniu wspomagających projektowanie narzędzi CASE. Celem ćwiczeń jest nabycie przez studentów praktycznej umiejętności posługiwania się modelami koncepcyjnymi i narzędziami wspomagającymi modelowanie. Praca odbywa się w zespołach kilkuosobowych. Każda grupa wykonuje zestaw ćwiczeń w odniesieniu do wybranego przez siebie i przeznaczonego do zinfomatyzowania wycinka rzeczywistości. Wynikiem jest kompletna dokumentacja oraz projekt systemu oraz (ewentualnie) zarys jego implementacji.</p>		
Wymagania wstępne i dodatkowe	Brak		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Sprawdziany	50.0%	15.0%
	Zadania	50.0%	85.0%

Zalecana lista lektur	Podstawowa lista lektur	<p>1. Sommerville I.: Inżynieria oprogramowania, WNT 2003</p> <p>2. Inżynieria oprogramowania w projekcie informatycznym, red. J. Górski, MIKOM, 2000</p> <p>3. Booch G., Rumbaugh J., Jacobson I.: UML przewodnik użytkownika, WNT 2002</p>
	Uzupełniająca lista lektur	<p>1. Subieta K.: Wprowadzenie do inżynierii oprogramowania, PJWSTK, 2002</p> <p>2. Jaskiewicz A.: Inżynieria oprogramowania, Helion, 2000</p> <p>3. Brooks F. P.: Eseje o inżynierii oprogramowania, WNT, 2000</p>
	Adresy eZasobów	Adresy na platformie eNauczanie: Inżynieria oprogramowania 2025 - Moodle ID: 45007 <a href="https://enauczanie.pg.edu.pl/moodle/course/view.php?id=45007">https://enauczanie.pg.edu.pl/moodle/course/view.php?id=45007</a>
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<ol style="list-style-type: none"> <li>1. Procesy tworzenia oprogramowania.</li> <li>2. Inżynieria wymagań.</li> <li>3. Zwinne metodyki tworzenia oprogramowania.</li> <li>4. Programowanie ekstremalne.</li> <li>5. Zarządzanie przedsięwzięciem programistycznym.</li> <li>6. Projektowanie obiektowe.</li> <li>7. Język UML.</li> <li>8. Szacowanie kosztu oprogramowania</li> <li>9. Projektowanie architektoniczne.</li> <li>10. Projektowanie systemów rozproszonych.</li> <li>11. Projektowanie systemów czasu rzeczywistego.</li> <li>12. Systemy krytyczne.</li> <li>13. Projektowanie z użyciem wielokrotnym.</li> <li>14. Wzorce projektowe.</li> </ol>	
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy	

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.