

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Inżynieria oprogramowania		Kod 1010331561010330109
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) (brak)	Rok / Semestr 3 / 6
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: - Projekty/seminaria: 15		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (brak)		(ogólnouczelniany, z innego kierunku) (brak)
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 4 100% 4 100%
Odpowiedzialny za przedmiot / wykładowca: dr inż. Andrzej Sikorski email: andrzej.sikorski@put.poznan.pl tel. 6653958 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Znajomość Visual Paradigm. Znajomość teoretycznych podstaw informatyki w zakresie programowania obiektowego, podstawowych algorytmów. Model relacyjny.
2	Umiejętności:	Inżynieria oprogramowania w zakresie 5 sem. (diagramy klas, przypadków użycia, diagramy wymagań). Programowanie w językach C,C++,C#. Podstawy baz danych. Programowanie współbieżne.
3	Kompetencje społeczne	Umiejętność pracy w zespole. Poczucie odpowiedzialności, lojalność oraz dyskrecja w sprawach poufnych.
Cel przedmiotu: Pozyskanie wiedzy z zakresu modelowania obiektowego, relacyjnego oraz współbieżnego. Biegłość w posługiwaniu się diagramami UML do celów. Przedmiot jest zorientowany na wsparcie studenta w zakresie ćwiczeń laboratoryjnych, podczas których powinien nabrać umiejętności w zakresie zagadnień omawianych na wykładzie.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. Ma uporządkowaną i podbudowaną metodologicznie wiedzę w zakresie inżynierii oprogramowania. - [K_W12] 2. Orientuje się w obecnym stanie oraz najnowszych trendach rozwojowych informatyki. - [K_W19]		
Umiejętności:		
1. Potrafi pozyskiwać informacje z dokumentacji technicznej narzędzi programistycznych, bibliotek, API. - [K_U16] 2. Potrafi odwzorowywać zadania programistyczne na struktury dostępne w środowiskach programistycznych. - [K_U03]		
Kompetencje społeczne:		
1. Ma świadomość ważności dokładnego wykonania oprogramowania, zachowania standardów notacyjnych - [K_K07] 2. Ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole - [K_K04]		

Sposoby sprawdzenia efektów kształcenia		
<p>Egzamin pisemny. zadania UML oraz programistyczne. Seminarium lub kolokwium w trakcie semestru. Zadania o podwyższonym stopniu trudności proponowane w trakcie semestru. Rozwiązywanie zadań technicznych z dziedziny przedmiotu proponowanych w treści wykładu.</p>		
Treści programowe		
<p>Diagramy UML dynamiczne: interakcji, sekwencji, przebiegów czasowych, stanów. Wzorce programowe w zakresie programowania współbieżnego. Specyfikacja w UML mechanizmów synchronizacyjnych wysokiego poziomu. Modelowanie systemów czasu rzeczywistego. Projektowanie relacyjne. Wzorce projektowe z zakresu modelu relacyjnego. Dzielenie relacyjne, anty-złączenie, techniki przepisywania zapytań SQL.</p>		
Literatura podstawowa:		
<p>1. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Addison-Wesley Professional; 2 edition (May 29, 2005) (dostępne też po polsku) 2. Erich Gamma et al. Design Patterns: Elements of Reusable Object-Oriented Software 3. Intel? Threading Building Blocks (Intel?TBB) User Guide (dostępne https://www.threadingbuildingblocks.org/)</p>		
Literatura uzupełniająca:		
<p>1. Bjarne Stroustrup: Język C++. Kompendium wiedzy, Helion 2. Chris Date: Wprowadzenie do systemów baz danych, Wydawnictwa Naukowo Techniczne , 2000 3. Stanisław Wrycza, Bartosz Marcinkowski, Jacek Maślankowski: UML 2.x. Ćwiczenia zaawansowane, Helion</p>		
Bilans nakładu pracy przeciętnego studenta		
Czynność		Czas (godz.)
1. Wykład		30
2. Zajęcie projektowe		15
3. Praca własna		20
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	65	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	3
Zajęcia o charakterze praktycznym	15	1