

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Inżynieria oprogramowania</b>		Kod <b>1010334561010330109</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>(brak)</b>	Rok / Semestr <b>3 / 6</b>
Ścieżka obieralności/specjalność <b>-</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>I stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>niestacjonarna</b>	
Godziny Wykłady: <b>16</b> Ćwiczenia: <b>-</b> Laboratoria: <b>-</b> Projekty/seminaria: <b>8</b>	Liczba punktów <b>4</b>	
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>(brak)</b>	(ogólnouczelniany, z innego kierunku) <b>(brak)</b>	
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>	Podział ECTS (liczba i %) <b>4 100%</b>	
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
dr inż. Adam Meissner email: Adam.Meissner@put.poznan.pl tel. 61 665 37 24 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Student ma uporządkowaną i podbudowaną teoretycznie wiedzę z zakresu podstaw inżynierii oprogramowania obejmującą elementy modelowania i realizacji systemów informatycznych.
2	<b>Umiejętności:</b>	Student potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł; potrafi integrować uzyskane informacje, dokonywać ich interpretacji, a także wyciągać wnioski oraz formułować opinie; potrafi pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zadania; potrafi opracować i zrealizować harmonogram prac; posługuje się językiem angielskim w stopniu wystarczającym do porozumiewania się oraz do czytania ze zrozumieniem wskazanej literatury przedmiotowej.
3	<b>Kompetencje społeczne</b>	Student rozumie potrzebę i zna możliwości ciągłego dokształcania się, podnoszenia kompetencji językowych, zawodowych, osobistych i społecznych; ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.
<b>Cel przedmiotu:</b>		
w ramach drugiej części przedmiotu studentów zapoznaje się z wybranymi metodami modelowania i wytwarzania systemów informatycznych oraz z metodami testowania i weryfikowania ich poprawności.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. Student ma uporządkowaną i podbudowaną metodologicznie wiedzę w zakresie inżynierii oprogramowania - [K_W12] 2. Student orientuje się w obecnym stanie oraz najnowszych trendach rozwojowych informatyki - [K_W19]		
<b>Umiejętności:</b>		
1. Student potrafi opracować dokumentację dotyczącą realizacji zadania inżynierskiego i przygotować tekst zawierający omówienie wyników realizacji tego zadania - [K_U03] 2. Student potrafi sformułować wymagania, opracować model obiektowy oraz ocenić prosty system informatyczny, uwzględniając realizowane funkcje i powiązania między elementami składowymi - [K_U16]		
<b>Kompetencje społeczne:</b>		
1. Student ma świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżyniera-informatyka i związaną z tym odpowiedzialność za podejmowane decyzje - [K_K02] 2. Student ma świadomość ważności dokładnego wykonania projektu, zachowania standardów notacyjnych, przestrzegania poprawności językowej i terminowego oddania prac - [K_K07]		
<b>Sposoby sprawdzenia efektów kształcenia</b>		

Wykład: egzamin pisemny. Projekt: ocena projektu systemu informatycznego wykonanego w standardzie UML.		
<b>Treści programowe</b>		
<p>Wykład. Przegląd modeli cyklu rozwojowego oprogramowania. Wytwarzanie oprogramowania w ujęciu systematycznym, standard ISO 9000, model CMM. Programowanie zwinne i programowanie ekstremalne. Wzorce projektowe. Metody refaktoryzacji programów, "zapachy" kodu. Metody testowania i walidacji oprogramowania.</p> <p>Projekt. Kontynuacja prac nad modelem wskazanego systemu informatycznego, prowadzonych w ramach pierwszej części przedmiotu. Model wykonuje się w standardzie UML, a jego nowymi elementami są diagramy czynności, diagramy kolejności oraz diagram wdrożenia.</p>		
<b>Literatura podstawowa:</b>		
<ol style="list-style-type: none"> <li>Paulish D.J., Inżynieria oprogramowania. Zarządzanie architekturocentrycznym procesem tworzenia oprogramowania. Przewodnik praktyczny, WNT, Warszawa, 2007</li> <li>Shore S., Warden S., Agile Development. Filozofia programowania zwinnego, Wyd. Helion, Gliwice, 2008</li> <li>Zmitrowicz K., Jakość projektów informatycznych. Rozwój i testowanie oprogramowania, Wyd. Helion, Gliwice, 2015</li> </ol>		
<b>Literatura uzupełniająca:</b>		
<ol style="list-style-type: none"> <li>Jeffries R., Programowanie ekstremalne w C#, PWN, Warszawa, 2005</li> </ol>		
<b>Bilans nakładu pracy przeciętnego studenta</b>		
<b>Czynność</b>		<b>Czas (godz.)</b>
1. Wykłady		16
2. Zajęcia projektowe		8
3. Przygotowanie do zajęć projektowych		20
4. Przygotowanie do egzaminu		15
5. Konsultacje i egzamin		5
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	64	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	29	2
Zajęcia o charakterze praktycznym	28	2