

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Inżynieria oprogramowania</b>		Kod <b>1010334551010330109</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>(brak)</b>	Rok / Semestr <b>3 / 5</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>12</b> Ćwiczenia: <b>-</b> Laboratoria: <b>8</b> Projekty/seminaria: <b>-</b>		Liczba punktów <b>3</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>3 100%</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 projektowania i implementowania algorytmów, metod i paradygmatów programowania, a także zna podstawy weryfikowania poprawności programów.
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>		
zapoznanie studentów z podstawami systematycznego wytwarzania oprogramowania; przedstawienie wybranych metod i narzędzi stosowanych w tej dziedzinie.		
<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>		

<p>Wykład: zaliczenie pisemne. Laboratorium: ocena postępu prac nad zadaniem semestralnym polegającym na przygotowaniu (w standardzie UML) częściowego modelu wskazanego systemu informatycznego.</p>		
<b>Treści programowe</b>		
<p>Wykład. Problematyka inżynierii oprogramowania. Podstawy modelowania programów. Dokumentowanie wymagań. Standard UML i modelowanie systemów informatycznych w tym standardzie. Formalne metody modelowania programów. Aktualizacja 2017: przedstawienie dodatkowych przykładów modeli wykonanych w standardzie UML. Laboratorium. Zapoznanie się z wybranymi narzędziami wspomagającymi projektowanie w standardzie UML. Opracowanie częściowego modelu wskazanego systemu informatycznego, obejmującego dokument wymagań oraz diagram przypadków użycia i diagramy klas w standardzie UML. Zastosowane metody kształcenia: - wykład z prezentacją slajdów uzupełniany przykładami podawanymi na tablicy - laboratorium - korzystanie z narzędzi umożliwiających studentom wykonanie zadań w domu, szczegółowe recenzowanie sprawozdań przez prowadzącego połączone z omówieniem typowych błędów oraz propozycjami ich usuwania.</p>		
<b>Literatura podstawowa:</b>		
<p>1. Wrycza S., Marcinkowski B., Wyrzykowski K., Język UML 2.0 w modelowaniu systemów informatycznych, Helion, Gliwice 2006</p>		
<b>Literatura uzupełniająca:</b>		
<p>1. Begier B., Inżynieria oprogramowania - problematyka jakości, Wydawnictwo Politechniki Pozn., Poznań 1999 2. Hamlet D., Maybee J., Podstawy techniczne inżynierii oprogramowania, WNT, Warszawa 2003 3. Pressman R. S., Praktyczne podejście do inżynierii oprogramowania, WNT, Warszawa 2004 4. Unified Modeling Language. Object Management Group, <a href="http://www.omg.org/spec/UML">http://www.omg.org/spec/UML</a> (dostęp: 15.09.2017)</p>		
<b>Bilans nakładu pracy przeciętnego studenta</b>		
<b>Czynność</b>	<b>Czas (godz.)</b>	
1. Wykłady	12	
2. Laboratoria	8	
3. Przygotowanie do ćwiczeń lab., przygotowanie sprawozdań	20	
4. Przygotowanie do zaliczenia	10	
5. Konsultacje i zaliczenie	5	
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	55	3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	25	1
Zajęcia o charakterze praktycznym	28	2