

KARTA PRZEDMIOTU (SYLABUS)

Opis przedmiotu

Kod przedmiotu		Nazwa przedmiotu	INŻYNIERIA OPROGRAMOWANIA	
I/O/1(i)/NST/B1-11			SOFTWARE ENGINEERING	
Język wykładowy		język polski		
Rok akademicki		2020/2021		
Kierunek		Informatyka		
w zakresie				
Poziom studiów		studia pierwszego stopnia		
Profil studiów		ogólnoakademicki		
Forma studiów		studia niestacjonarne		
Semestr / semestry		szósty		
Przynależność do grupy zajęć		B1. Grupa zajęć kierunkowych - obowiązkowych		
Status przedmiotu		obowiązkowy		
Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS		Forma zajęć	Liczba godzin zajęć dydaktycznych	Liczba punktów ECTS
		Wykład	30 [h]	3ECTS
		Laboratorium	30 [h]	
Powiązanie przedmiotu	z profilem studiów	związany z prowadzoną działalnością naukową w dyscyplinach, do których przyporządkowany jest kierunek studiów		3 ECTS
	z uprawnieniami	służy do zdobywania przez studenta kompetencji inżynierskich		3 ECTS
	z dyscypliną	informatyka techniczna i telekomunikacja informatyka		1 ECTS 2 ECTS
Forma nauczania		tradycyjna - zajęcia zorganizowane w Uczelni i/lub zajęcia z wykorzystaniem metod i technik kształcenia na odległość (max. 1,2 ECTS)		
Wymagania wstępne		Wymagana dobra znajomość przedmiotów: baz danych, języki i paradygmaty programowania oraz obiektowe programowanie pozwalająca na tworzenie oraz implementację oprogramowania składowego systemów informatycznych		
Jednostka prowadząca		Katedra Informatyki		
Koordynator		prof. A. Sachenko		
Adres strony internetowej pjo		www.wteii.uniwersytetradom.pl		
Adres e-mail, telefon koordynatora		a.sachenko@uthrad.pl tel. +48 (48) 361-78-06		

EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE, REALIZACJA ZAJĘĆ DYDAKTYCZNYCH, WERYFIKACJA EFEKTÓW UCZENIA SIĘ

<p>Cel kształcenia:</p>	<p>Celem przedmiotu jest zapoznanie studentów z różnymi aspektami inżynierii oprogramowania. Wykład jest zorganizowany według faz życia oprogramowania: faza strategiczna, faza wymagań użytkownika, faza analizy, faza projektowania, faza konstrukcji, faza testowania oraz faza instalacji i konserwacji. Wykład stanowi analizę (i przykład zastosowania) programowania obiektowego w projektowaniu systemów informatycznych w oparciu o język UML.</p> <p>Celem zajęć laboratoryjnych jest praktyczne zapoznanie studenta z kolejnymi fazami wytwarzania oprogramowania w trakcie tworzenia projektu oraz implementacji wybranego modułu systemu informatycznego z uwzględnieniem analizy projektowania obiektowego w języku UML. Praca odbywa się przy wykorzystaniu dostępnych narzędzi: Visual Studio i Visio Professional lub darmowego oprogramowania wspomagającego modelowanie (np.: Star UML, DIA). Studenci mogą do projektu zaliczeniowego używać wybranego oprogramowania: MySQL, MS Access, HTML, PHP, C#, C++, Delphi lub innych narzędzi, natomiast dokumentację projektową mogą wykonać w MS Word.</p>
<p>Treści programowe:</p>	<p>Wykład -W1, W2, W3</p> <p>Wprowadzenie w metodologiczne podstawy tworzenia oprogramowania.(2h) Zaprezentowanie modeli cyklu życia oprogramowania, wyjaśnienie podstawowych pojęć i zakresu tematycznego inżynierii oprogramowania.(3h) Omówienie metodyki obiektowej języka UML.(3h) Przedstawienie modelu przypadków użycia oraz modeli obiektowych; w procesie tworzenia modelu obiektowego: identyfikację obiektów i klas, identyfikacji związków między klasami, diagramy interakcji, diagramy stanu i inne.(6h) Rozważenie wymagań systemu i jego specyfikacji.(4h) Prześledzenie procesu rozwoju systemu w związku z jego architekturą.(2h) Zastosowanie narzędzia CASE.(2h) Omówienie zasad tworzenia dokumentacji systemu. (2h) Śledzenie etapu implementacji i testowania oprogramowania, z podkreśleniem właściwego zarządzania przedsięwzięciem programistycznym. (4h) Harmonogramowanie i monitorowanie oraz szacowanie kosztów oprogramowania. (2h)</p> <p>Laboratorium – U1, U2, K1, K2</p> <p>Laboratorium obejmuje samodzielne opracowanie fazy projektu: strategicznej, wymagań użytkownika, analizy, projektowania, elementów konstrukcji i implementacji dla wybranego modułu systemu informatycznego metodą obiektową. Student realizuje zadany projekt modułów biznesowych używając dostępnych narzędzi do projektowania i implementacji. Student buduje model systemu, a następnie częściowo go implementuje w wybranym przez siebie języku programowania np.: C++, C#, PHP, Delphi oraz przygotowuje dokumentację systemu. - U1, U2, K1, K2</p> <p>Harmonogram laboratorium obejmuje:</p> <ul style="list-style-type: none"> – nabycie umiejętności prowadzenia całego procesu produkcji oprogramowania;(2h) – rozmowa z klientem – zamawiającym system informatyczny;(4h) – przygotowanie założeń projektowanego systemu: funkcjonalnych i niefunkcjonalnych;(4h) – wykonanie diagramów projektu w języku UML;(4h) – wybranie modelu otrzymania kodu wynikowego na podstawie teorii inżynierii oprogramowania;(2h) – zaimplementowanie w ten sposób zdefiniowanego systemu w obiektowym języku;(6h) – testowanie;(4h) – wykonanie dokumentacji otrzymanej aplikacji (4h) - U1, U2, K1, K2
<p>Metody dydaktyczne (kształcenia):</p>	<p>Metody podające – wykład z wykorzystaniem technik multimedialnych – W1, W2, W3</p> <p>Metody programowane z wykorzystaniem komputera – ćwiczenia laboratoryjne- U1, U2, K1, K2</p> <p>Metody praktyczne – ćwiczenia laboratoryjne - U1, U2, K1, K2</p> <p>Wszystkie zastosowane metody umożliwiają rozpoznawanie i zaspokajanie indywidualnych potrzeb studentów, w tym studentów niepełnosprawnych oraz indywidualizację toku studiów.</p>

Rygor zaliczenia, kryteria oceny osiągniętych efektów uczenia się, sposób obliczania oceny końcowej:	<p>Warunkiem zaliczenia przedmiotu jest osiągnięcie wszystkich wymaganych efektów kształcenia określonych dla danego przedmiotu. Uzyskanie pozytywnych ocen z zaliczenia laboratorium oraz z egzaminu jest równoznaczne z zaliczeniem przedmiotu i zdobyciem przez studenta liczby punktów ECTS przyporządkowanej temu przedmiotowi.</p> <p>Wykład: Ocena końcowa z przedmiotu stanowi 100% oceny z egzaminu. Egzamin przeprowadzany jest w formie pisemnej, polega na samodzielnym rozwiązaniu przez studenta testu oraz wykonania zadań z zakresu tematyki omawianej na wykładach.</p> <p>Laboratorium: Warunkiem zaliczenia laboratorium jest osiągnięcie wszystkich wymaganych efektów kształcenia dla tej formy zajęć i uzyskanie pozytywnych ocen za pomocą przyjętych dla przedmiotu metod oceniania. W czasie laboratorium student wykonuje projekt polegający na zaprojektowaniu, zaimplementowaniu, przetestowaniu i uruchomieniu wybranego modułu systemu informatycznego. Ocena końcowa z laboratorium stanowi 100% oceny za wykonanie samodzielnego projektu na zaliczenie.</p>
--	--

Efekty uczenia się dla przedmiotu w odniesieniu do efektów kierunkowych i formy zajęć				Metody weryfikacji efektów uczenia się	
Numer efektu uczenia się	Opis efektów uczenia się dla przedmiotu (PEU) Student, który zaliczył przedmiot (W) zna i rozumie/ (U) potrafi / (K) jest gotów do:	Kierunkowy efekt uczenia się (KEU)	Forma zajęć	Forma weryfikacji (zaliczeń)	Metody sprawdzania i oceny
W1	Ma wiedzę na temat projektowania oprogramowania i korzystania z API (Application Programming Interface).	K_WG06	wykład, ćwiczenia laboratoryjne	egzamin pisemny, zaliczenie na ocenę	egzamin pisemny, wykonanie projektu
W2	Zna narzędzia i środowiska wytwarzania oprogramowania oraz procesy wytwarzania oprogramowania i metody projektowania.	K_WG06 K_WG13	wykład, ćwiczenia laboratoryjne	egzamin pisemny, zaliczenie na ocenę	egzamin pisemny, wykonanie projektu
W3	Ma wiedzę na temat metod walidacji i testowania oprogramowania, ewolucji oprogramowania oraz zarządzania przedsięwzięciem programistycznym.	K_WG14	wykład, ćwiczenia laboratoryjne	egzamin pisemny, zaliczenie na ocenę	egzamin pisemny, wykonanie projektu
U1	Umie posługiwać się wzorcami Projektowymi, projektować oprogramowania zgodnie z metodyką obiektową oraz dokonać przeglądu projektu oprogramowania.	K_UW07 K_UW09	ćwiczenia laboratoryjne	egzamin pisemny, zaliczenie na ocenę	egzamin pisemny, wykonanie projektu
U2	Umie wybrać narzędzia wspomagające budowę oprogramowania, dobrać model procesu wytwarzania oprogramowania do specyfiki przedsięwzięcia.	K_UW14 K_UW16 K_UW17	ćwiczenia laboratoryjne	egzamin pisemny, zaliczenie na ocenę	egzamin pisemny, wykonanie projektu
K1	Potrafi wyspecyfikować wymagania dotyczące oprogramowania i dokonać ich przeglądu. Potrafi utworzyć, ocenić i zrealizować plan testowania.	K_KK03 K_KO06	ćwiczenia laboratoryjne	egzamin pisemny, zaliczenie na ocenę	egzamin pisemny, wykonanie projektu
K2	Potrafi zarządzać konfiguracją oprogramowania oraz opracowywać plan przedsięwzięcia dotyczącego budowy oprogramowania.	K_KK03 K_KO06	ćwiczenia laboratoryjne	egzamin pisemny, zaliczenie na ocenę	Egzamin pisemny, wykonanie projektu
Stopień osiągnięcia kierunkowych efektów uczenia się: K_WG06-+++, K_WG13-+++, K_WG14-+++, K_UW14-+++, K_UW16-+++, K_UW17-+++, K_UW07-+++, K_UW08-+++, K_KK03-++, K_KO06-++					

Literatura podstawowa, literatura uzupełniająca, pomoce naukowe
<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> 1. Flower M.: <i>UML w kropelce</i>, wyd. III, UML wersja 2.0, Oficyna Wydawnicza LPT, Warszawa 2005. 2. Górski J. (red.): <i>Inżynieria oprogramowania w projekcie informatycznym</i>. Wyd.2, Mikom, Warszawa 2000. 3. Marcinkowski B., Wrycza S.: <i>Język SysML w modelowaniu systemów informatycznych</i>, Helion 2008. 4. Sommerville I.: <i>Inżynieria oprogramowania</i>, WNT, Warszawa 2003. 5. Subieta K.: <i>Wprowadzenie do inżynierii oprogramowania</i>, PJWSTK, Warszawa 2002. 6. Wykrzykowski K.: <i>Język UML do modelowania systemów informatycznych</i>, Helion 2010. <p>Literatura uzupełniająca:</p> <ol style="list-style-type: none"> 1. Cockburn A.: <i>Jak pisać efektywne przypadki użycia</i>. WNT, Warszawa 2004. 2. Muller R. J.: <i>Bazy danych: język UML w modelowaniu danych</i>, Mikom, Warszawa 2000. 3. Schmuller J.: <i>UML dla każdego</i>, Helion, Gliwice, 2003. 4. Schneider G., Winters J. P.: <i>Stosowanie przypadków użycia</i>, WNT, Warszawa 2004. 5. Wrycza S., Marcinkowski B., Wyrzykowski K.: <i>Język UML 2.0 w modelowaniu systemów informatycznych</i>, Helion, Gliwice 2005. <p>Pomoce naukowe: Materiały dydaktyczne opracowane przez wykładowcę</p>

Nakład pracy studenta potrzebny do osiągnięcia zakładanych efektów uczenia się – bilans punktów ECTS

Udział w zajęciach, aktywność	Obciążenie studenta [h]		
	Inne godz. kontaktowe (IGK)	Zajęcia bez nauczyciela-praca własna studenta (ZBN)	Zajęcia dydaktyczne
Udział w wykładach	X	X	30 [h]
Samodzielne studiowanie tematyki wykładów	X	15 [h]	X
Udział w ćwiczeniach laboratoryjnych	X	X	30[h]
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	X	5 [h]	X
Udział w konsultacjach	8 [h]	X	X
Przygotowanie do egzaminu	X	5 [h]	X
Udział w egzaminie	2 [h]	X	X
Sumaryczne obciążenie pracą studenta	10[h]/ 0,4 ECTS	15 [h]/0,6 ECTS	60[h]/ 2 ECTS
Punkty ECTS za przedmiot	3 ECTS		

Informacje dodatkowe, uwagi