

KARTA PRZEDMIOTU (SYLABUS)

Opis przedmiotu

Kod przedmiotu		Nazwa przedmiotu	INŻYNIERIA OPROGRAMOWANIA	
UTH/I/A/IN/-/B ₁ /NST/1(i)/6L/11			SOFTWARE ENGINEERING	
Język wykładowy		język polski		
Rok akademicki		2019/2020		
Kierunek		Informatyka		
w zakresie				
Poziom studiów		studia pierwszego stopnia		
Profil studiów		ogólnoakademicki		
Forma studiów		studia niestacjonarne		
Semestr / semestry		semestr szósty letni		
Przynależność do grupy zajęć		B ₁ - Moduł przedmioty kierunkowe - obowiązkowe		
Status przedmiotu		Obowiązkowy		
Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS		Forma zajęć	Liczba godzin zajęć dydaktycznych	Liczba punktów ECTS
		Wykład	30 [h]	4 ECTS
		Laboratorium	30 [h]	
		
Powiązanie przedmiotu	z profilem studiów	związany z prowadzoną działalnością naukową w dyscyplinach, do których przyporządkowany jest kierunek studiów		4 ECTS
	z uprawnieniami	służy do zdobywania przez studenta kompetencji inżynierskich		4 ECTS
	z dyscypliną	informatyka techniczna i telekomunikacja informatyka		2 ECTS 2 ECTS
Forma nauczania		tradycyjna - zajęcia zorganizowane w Uczelni		
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		dr Elżbieta Mączyńska		
Osoby prowadzące		dr Elżbieta Mączyńska		
Adres strony internetowej pjo		http://www.ki.uniwersytetradom.pl/		
Adres e-mail, telefon koordynatora		elzbieta.maczynska@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><i>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.</i></p> <p><i>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.</i></p>
<p>Treści programowe:</p>	<p>Wykład - W1, W2, W3</p> <p><i>Wprowadzenie w metodologiczne podstawy tworzenia oprogramowania.(2h)</i> <i>Zaprezentowanie modeli cyklu życia oprogramowania, wyjaśnienie podstawowych pojęć i zakresu tematycznego inżynierii oprogramowania.(3h)</i> <i>Omówienie metodyki obiektowej języka UML.(3h)</i> <i>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)</i> <i>Rozważenie wymagań systemu i jego specyfikacji.(4h)</i> <i>Prześledzenie procesu rozwoju systemu w związku z jego architekturą.(2h)</i> <i>Zastosowanie narzędzia CASE.(2h)</i> <i>Omówienie zasad tworzenia dokumentacji systemu. (2h)</i> <i>Śledzenie etapu implementacji i testowania oprogramowania, z podkreśleniem właściwego zarządzania przedsięwzięciem programistycznym.(4h)</i> <i>Harmonogramowanie i monitorowanie oraz szacowanie kosztów oprogramowania. (2h)</i></p> <p>Laboratorium – U1, U2, K1, K2</p> <p><i>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</i></p> <p><i>Harmonogram laboratorium obejmuje:</i></p> <ul style="list-style-type: none"> - <i>nabycie umiejętności prowadzenia całego procesu produkcji oprogramowania;(2h)</i> - <i>rozmowa z klientem – zamawiającym system informatyczny;(4h)</i> - <i>przygotowanie założeń projektowanego systemu: funkcjonalnych i niefunkcjonalnych;(4h)</i> - <i>wykonanie diagramów projektu w języku UML;(4h)</i> - <i>wybranie modelu otrzymania kodu wynikowego na podstawie teorii inżynierii oprogramowania;(2h)</i> - <i>zaimplementowanie w ten sposób zdefiniowanego systemu w obiektowym języku;(6h)</i> - <i>testowanie;(4h)</i> - <i>wykonanie dokumentacji otrzymanej aplikacji.(4h)- U1, U2, K1, K2</i>
<p>Metody dydaktyczne (kształcenia):</p>	<p><i>Metody podające – wykład z wykorzystaniem technik multimedialnych – W1, W2, W3</i></p> <p><i>Metody programowane z wykorzystaniem komputera – ćwiczenia laboratoryjne- U1,U2, K1, K2</i></p> <p><i>Metody praktyczne – ćwiczenia laboratoryjne - U1,U2, K1, K2</i></p>

	<p>Wszystkie zastosowane metody umożliwiają rozpoznawanie i zaspokajanie indywidualnych potrzeb studentów, w tym studentów niepełnosprawnych oraz indywidualizację toku studiów.</p>
<p>Rygor zaliczenia, kryteria oceny osiągniętych efektów uczenia się, sposób obliczania oceny końcowej:</p>	<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. Sposób obliczenia oceny końcowej z przedmiotu określony został uchwałą Rady Wydziału.</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 samodzielnie 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 oprogramowanie 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.

Literatura uzupełniająca:

1. Cockburn A.: *Jak pisać efektywne przypadki użycia*. WNT, Warszawa 2004.
2. Muller R. J.: *Bazy danych: język UML w modelowaniu danych*, Mikom, Warszawa 2000.
3. Schuller J.: *UML dla każdego*, Helion, Gliwice, 2003.
4. Schneider G., Winters J. P.: *Stosowanie przypadków użycia*, WNT, Warszawa 2004.
5. Wrycza S., Marcinkowski B., Wyrzykowski K.: *Język UML 2.0 w modelowaniu systemów informatycznych*, Helion, Gliwice 2005.

Pomoce naukowe: Materiały dydaktyczne opracowane przez wykładowcę

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	10 [h]	X
Udział w ćwiczeniach laboratoryjnych	X	X	30[h]
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	X	10 [h]	X
Udział w konsultacjach	8 [h]	X	X
Przygotowanie do egzaminu	X	10 [h]	X
Udział w egzaminie	2 [h]	X	X
Sumaryczne obciążenie pracą studenta	10[h]/ 0,4 ECTS	30 [h]/1,2 ECTS	60[h]/ 2,4 ECTS
Punkty ECTS za przedmiot	4 ECTS		

Informacje dodatkowe, uwagi

Studentowi przysługuje jeden termin podstawowy i jeden termin poprawkowy zaliczenia dla każdej formy zajęć. Obecność na zajęciach laboratoryjnych jest obowiązkowa. Obecność na wykładach jest zalecana. W przypadku zaległości powstałych wskutek nieobecności studenta na zajęciach projektowych student jest zobowiązany do uczestnictwa w zajęciach innej grupy (tzw. odrobienie zajęć) lub wykonania (w przypadku braku możliwości odrobienia) i zaliczenia dodatkowego projektu.

Zgodnie z Regulaminem Studiów UTHRad podstawowym terminem uzyskania zaliczenia jest ostatni dzień zajęć w danym semestrze. Termin zaliczenia poprawkowego (tryb i warunki ustala prowadzący moduł na zajęciach początkowych) nie może być późniejszy niż ostatni termin egzaminu w sesji poprawkowej (dla przedmiotów kończących się egzaminem) lub ostatni dzień trwania semestru (dla przedmiotów niekończących się egzaminem).

Terminy odbywania zajęć: semestr letni, zgodnie z rozkładem zajęć.

Miejsce odbywania zajęć: ul. Malczewskiego 20A, Radom; sala 105 Olimp