

Nazwa przedmiotu <i>Inżynieria oprogramowania</i> <i>Software Engineering</i>		Kod ECTS		
Nazwa jednostki prowadzącej przedmiot <i>Uniwersytet Opolski, Wydział Matematyki, Fizyki i Informatyki, Instytut Matematyki i Informatyki</i>				
Studia				
	Kierunek <i>Informatyka</i>	stopień <i>Pierwszy</i>	tryb <i>Stacjonarne</i> <i>Niestacjonarne *</i>	specjalność
				specjalizacja
Nazwisko osoby prowadzącej (osób prowadzących) <i>Krótkiewicz, Wojtkiewicz</i>				
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS: 6 <i>Bilans nakładu pracy przeciętnego studenta:</i> <ul style="list-style-type: none"> • 30 godz. – udział w wykładach [^{*)}18] • 30 godz. – udział w laboratoriach [^{*)}18] • 15 godz. – analiza i przyswojenie treści wykładu [^{*)}30] • 5 godz. – udział w konsultacjach do wykładu [^{*)}2] • 15 godz. – przygotowanie do laboratoriów [^{*)}30] • 35 godz. – przygotowanie aplikacji zaliczeniowej na laboratorium. [^{*)}35] 		
A. Formy zajęć <ul style="list-style-type: none"> • wykład (W), • laboratorium (L) 		B. Sposób realizacji <ul style="list-style-type: none"> • zajęcia w sali wykładowej/laboratoryjnej 		
C. Liczba godzin <i>Wykład – 30 godzin</i> <i>Laboratorium – 30 godzin</i> <i>*) Studia niestacjonarne:</i> <i>Wykład – 18 godz. (2T+16Z)</i> <i>Laboratorium – 18 godzin</i>		<i>Łączny nakład pracy studenta: 130 godzin, co odpowiada 5 pkt ECTS</i> <i>w tym</i> <ul style="list-style-type: none"> • nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich: 30+30+5=65 godz., co odpowiada 3 pkt ECTS; • nakład pracy związany z zajęciami o charakterze praktycznym: 30+5+15+35= 85 godz., co odpowiada 4 pkt ECTS <i>*) na studiach niestacjonarnych:</i> <ul style="list-style-type: none"> • nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich: 18+18+2=38 godz., co odpowiada 2 pkt. ECTS; • nakład pracy związany z zajęciami o charakterze praktycznym: 18+2+30+35 = 85 godz., co odpowiada 4 pkt. ECTS 		
Status przedmiotu <ul style="list-style-type: none"> • obowiązkowy (kanon) 		Język wykładowy <i>Polski (możliwość realizacji w języku angielskim)</i>		
Metody dydaktyczne <ul style="list-style-type: none"> • wykład z prezentacją multimedialną • ćwiczenia laboratoryjne: opracowanie dokumentacji projektowej wg zadanego szablonu, wykonanie aplikacji komputerowej 		Forma i sposób zaliczenia oraz podst. kryteria oceny lub wymagania egzaminacyjne <i>Na ogólnych zasadach określonych w programie kształcenia, a w szczególności</i>		
		A. Sposób zaliczenia <ul style="list-style-type: none"> • zaliczenie z oceną (laboratorium) • egzamin (wykład) 		
		B. Formy zaliczenia <ul style="list-style-type: none"> • (W) egzamin na ocenę (wykład) – forma pisemna (80%), punkty z zaliczenia laboratorium (20%); • (L) zaliczenie (0-20 pkt.): ustalenie zaliczenia na podstawie ocen częściowych otrzymywanych w trakcie trwania semestru za poszczególne elementy aplikacji zaliczeniowej 		
		C. Podstawowe kryteria <ul style="list-style-type: none"> • (W)(L) uzyskanie pozytywnej oceny; 		
Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi <i>Należy określić:</i> <ul style="list-style-type: none"> A. <u>Wymagania formalne</u>: pozytywna ocena z przedmiotów: Programowanie 3 oraz Bazy Danych 1 B. <u>Wymagania wstępne</u>: umiejętność programowania obiektowego, umiejętność projektowania i implementacji relacyjnych baz danych, znajomość UML, umiejętność wykorzystania pakietu biurowego MS Office oraz umiejętność korzystania z oprogramowania do modelowania np. Sybase Power Designer. 				
Cele przedmiotu <i>Przekazanie wiedzy i umiejętności z zakresu inżynierii oprogramowania umożliwiającej studentowi przyjąć różne role w procesie tworzenia oprogramowania. Student powinien rozumieć i poprawnie identyfikować problemy związane z tym obszarem oraz powinien posiadać umiejętność praktycznej realizacji projektu informatycznego związanego z zaprojektowaniem i wykonaniem aplikacji komputerowej.</i>				

Treści programowe

A. Problematyka wykładu:

Inżynieria oprogramowania jako dziedzina wiedzy. Zarządzanie projektem, aspekty ekonomiczne i prawne realizacji projektów informatycznych. Zarządzanie ryzykiem. Podstawy zarządzania zespołami ludzkim. Istota studium dopuszczalności przedsięwzięcia oraz zasady postępowania podczas jego tworzenia. Inżynieria wymagań. Analiza problemu, modelowanie strukturalne i behawioralne. Projektowanie składowych systemu informatycznego. Implementacja, testowanie, dokumentacja i wdrożenie oprogramowania.

B. Problematyka laboratorium:

Wykonanie dokumentacji projektowej (zgodnie z szablonem) zawierającej m.in.: studium wykonalności, identyfikację punktów widzenia, hierarchiczną strukturę wymagań, diagram przypadków użycia, rozkład funkcjonalny, model systemu, architekturę wdrożeniową systemu, wykres Gantta, dokumentację użytkownika.

Wykonanie autorskiej aplikacji komputerowej zgodnej z przygotowaną dokumentacją, przy zastosowaniu wybranych technologii. Aplikacja komputerowa musi zawierać co najmniej jedną funkcjonalność zrealizowaną w formie modułu napisanego w języku programowania.

Wykaz literatury

A. Literatura wymagana do ostatecznego zaliczenia zajęć (zdania egzaminu):

A.1. wykorzystywana podczas zajęć

1. Wrycza, Stanisław : Język UML 2.0 w modelowaniu systemów informatycznych : diagramy języka UML, modelowanie biznesowe, metodyki projektowe oparte na UML, narzędzia CASE / Gliwice : Wydawnictwo Helion, cop. 2005, ISBN 83-7361-892-9
2. Malina, Witold. : Metodologia i techniki programowania / Warszawa : Wydawnictwo Naukowe PWN : Mikom, cop. 2008, ISBN 978-83-01-15469-1 (PWN).

A.2. studiowana samodzielnie przez studenta

1. Cadle, James. : Zarządzanie procesem tworzenia systemów informacyjnych / Warszawa : Wydawnictwa Naukowo-Techniczne, cop. 2004, Inżynieria Oprogramowania, ISBN 83-204-2928-5
2. Cantor, Murray. : Jak kierować zespołem programistów / Warszawa : Wydawnictwa Naukowo-Techniczne, 2004, Inżynieria Oprogramowania, ISBN 83-204-2880-7

B. Literatura uzupełniająca

1. Inżynieria systemów informatycznych, Beynon-Davies Paul, WNT 2004
2. Metryki i modele w inżynierii jakości oprogramowania, Stephen H.Kan, PWN 2006
3. inne podręczniki dostępne on-line poprzez Bibliotekę Główną UO („ibuk”)

Efekty kształcenia	Wiedza			
	Symb.	Efekt	Metoda weryfikacji	Odniesienie
	W01	Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie inżynierii oprogramowania	praca kontrolna/ projekt	K_W03
	W02	Zna podstawy komunikacji człowiek-komputer, w tym podstawę budowy interfejsów graficznych		K_W010
	W03	Ma wiedzę na temat zarządzania informacją		K_W012
	W04	Ma wiedzę na temat projektowania oprogramowania; specyfikacji i analizy wymagań; walidacji i testowania oprogramowania; zarządzania przedsięwzięciem programistycznym; zna narzędzia i środowiska wytwarzania oprogramowania; procesy wytwarzania oprogramowania		K_W013
	W05	Ma podstawową wiedzę dotyczącą prawnych i społecznych aspektów informatyki oraz uwarunkowań działalności inżynierskiej, w tym odpowiedzialności zawodowej i etycznej.		K_W015
	W06	Zna ogólne zasady tworzenia i rozwoju małych przedsiębiorstw informatycznych, ma podstawową wiedzę dotyczącą zarządzania.		K_W016
	Umiejętności:			
	Symb.	Efekt	Metoda weryfikacji	Odniesienie
U01	Potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym	praca kontrolna/ projekt	K_U09	
U02	Potrafi stworzyć model obiektowy prostego systemu (np. w języku UML)		K_U25	
U03	Ma umiejętność budowy prostych systemów bazodanowych wykorzystujących przynajmniej jeden z najbardziej popularnych systemów zarządzania bazą danych		K_U26	
U04	Potrafi formułować zapytania do bazy danych w wybranym języku zapytań, zanurzać zapytania w języku programowania, oceniać strategię wykonywania zapytań rozproszonych		K_U27	
U05	Ocenia przydatność różnych paradygmatów i związanych z nimi środowisk programistycznych do rozwiązywania różnego typu problemów		K_U28	
U06	Projektuje i implementuje oprogramowanie zgodnie z metodyką obiektową		K_U29	
U07	Potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować		K_U32	

	<i>prosty system informatyczny, używając właściwych metod, technik i narzędzi</i>		
U08	<i>Potrafi wykonać prostą analizę sposobu funkcjonowania systemu informatycznego i ocenić istniejące rozwiązania informatyczne, przynajmniej w odniesieniu do ich cech funkcjonalnych</i>		K_U33
U09	<i>Tworzy, ocenia i realizuje plan testowania oraz jest przygotowany do efektywnego uczestniczenia w inspekcji oprogramowania</i>		K_U34
U10	<i>Zna przynajmniej jeden system zarządzania wersjami</i>		K_U35
U11	<i>Potrafi dokonać wstępnej analizy ekonomicznej projektowanego systemu informatycznego</i>		K_U37
U12	<i>Potrafi posługiwać się wzorcami projektowymi</i>		K_U38
Kompetencje społeczne (postawy)			
Symb.	Efekt	Odniesienie	
Kontakt:			
Wykaz numerów telefonicznych i adresów mailowych pracowników znajduje się na stronie Instytutu Matematyki i Informatyki: www.math.uni.opole.pl			