

OPIS PRZEDMIOTU KSZTAŁCENIA

Nazwa przedmiotu Inżynieria oprogramowania I					
Nazwa jednostki prowadzącej przedmiot Instytut Matematyki					
Kierunek	Specjalność	Specjalizacja	Semestr/y	Poziom kształcenia/ profil kształcenia	Forma studiów
Informatyka	Programowanie	-	5	SPS/ praktyczny	stacjonarne/ niestacjonarne
Nazwisko osoby prowadzącej (osób prowadzących) dr inż. Zbigniew Ledóchowski, dr Ryszard Motyka					
Formy zajęć	Liczba godzin				Liczba punktów ECTS
	N (nauczyciel)		S (student)		
	studia stacjonarne	studia niestacjonarne	studia stacjonarne	studia niestacjonarne	
(W) wykład	15	9	35	41	2
Przygotowanie do zaliczenia			25	30	
Studiowanie literatury			10	11	
(CL) ćwiczenia laboratoryjne	20	12	40	48	2
Przygotowanie do zajęć (w tym rozwiązywanie zadań domowych)			10	10	
Przygotowanie projektów			30	38	
Razem	35	21	75	89	4
Metody dydaktyczne					
<ul style="list-style-type: none"> • (W) wykład: wykład problemowy wspomagany pokazem multimedialnym • (CL) ćwiczenia laboratoryjne: ćwiczenia praktyczne - rozwiązywanie zadań z wykorzystaniem komputera, metoda projektu, metoda problemowa 					
Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi					
<p>A. Wymagania formalne: Algorytmy i struktury danych, Programowanie I-V, Wstęp do gromadzenia i przetwarzania danych, Bazy danych</p> <p>B. Wymagania wstępne: wiadomości i umiejętności z przedmiotów związanych z architekturą komputera i podstawami informatyki oraz wiedza z zakresu podstaw programowania</p>					
Cele przedmiotu					
<ul style="list-style-type: none"> • zapoznanie z cyklem życia oprogramowania i jego szczegółowymi etapami. • wytworzenie kompetencji umożliwiających analizowanie i tworzenie wymagań dotyczących projektów informatycznych • zapoznanie z pojęciami dotyczącymi projektowania (zwłaszcza z wykorzystaniem modelu obiektowego), wytwarzania i testowania oprogramowania oraz ewolucji oprogramowania • nabycie podstawowych umiejętności w zakresie wykorzystywania wzorców projektowych • zapoznanie z narzędziami informatycznymi wspomagającymi proces projektowania programowania 					
Treści przedmiotu					
<ol style="list-style-type: none"> 1. Przedmiot inżynierii oprogramowania. 2. Cykl życia oprogramowania na poziomie ogólnym. 3. Kontekst biznesowy projektów informatycznych. Analiza wymagań biznesowych. 4. Analiza i inżynieria wymagań. Zbieranie, porządkowanie, wycena i poziomy wymagań. Wymagania pozafunkcjonalne dotyczące systemów informatycznych (wydajność, użyteczność, niezawodność, bezpieczeństwo). 5. Studium wykonalności projektów informatycznych. 6. Sposobu pozyskiwania wymagań. Specyfikacja wymagań. Przypadki użycia. Modelowanie i dokumentowanie 					

<p>wymagań. Wzorzec SRS. Walidacja wymagań.</p> <p>7. Model obiektowy w programowaniu. Najpopularniejsze wzorce projektowe w programowaniu obiektowym.</p> <p>8. Sposoby projektowania rozwiązania informatycznego na bazie wymagań na oprogramowanie - wprowadzenie do języka UML. Diagramy UML.</p> <p>9. Narzędzia CASE do etapu analitycznego i do projektowania oprogramowania.</p>															
<p>Efekty kształcenia</p> <p>Wiedza</p> <p>W_01 opisuje etapy tworzenia i funkcjonowania oprogramowania , także ewolucję oprogramowania</p> <p>W_02 opisuje najważniejsze wymagania dotyczące systemów informatycznych</p> <p>W_03 zna metody modelowania wymagań, projektowania i wytwarzania oprogramowania w tym uwzględniające wykorzystanie wzorców projektowych oraz wspomagania procesu tworzenia oprogramowania przez narzędzia informatyczne</p> <p>Umiejętności</p> <p>U_01 dokonuje analizy wymagań dotyczących projektu interfejsu graficznego</p> <p>U_02 tworzy specyfikację wymagań oraz studium wykonalności dla projektowanego systemu informatycznego wykorzystując różne techniki dokumentowania wymagań</p> <p>U_03 konstruuje diagramy UML związane z modelowaniem oprogramowania na bazie wymagań</p> <p>U_04 wykorzystuje wzorce projektowe oraz narzędzia CASE do wspomagania procesu analizy i projektowania oprogramowania.</p> <p>U_05 stosuje przy modelowaniu rozwiązania informatycznego techniki obiektowe.</p> <p>Kompetencje społeczne</p> <p>K_01 wykazuje kreatywność przy rozwiązywaniu problemów</p> <p>K_02 ma świadomość ograniczeń wynikających z niedostatecznej wiedzy i umiejętności i rozumie potrzebę dalszego kształcenia, jest otwarty na poszukiwanie niestandardowych rozwiązań</p>	<p>Sposób zaliczenia oraz formy i podstawowe kryteria oceny/wymagania egzaminacyjne</p> <p>A. Sposób zaliczenia</p> <p>W – zaliczenie z oceną CL – zaliczenie z oceną</p> <p>B. Sposoby weryfikacji i oceny efektów</p> <p>Wykład –test końcowy – efekty: W_01, W_02,W_03, K_03</p> <p>Oceną zaliczenia wykładu jest ocena uzyskaną za test końcowy -ocena A.</p> <p>Maksymalnie w teście można uzyskać x punktów, a ocena A jest ustalana na podstawie następujących kryteriów.</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">A ∈ [0% x, 50% x)</td> <td style="text-align: right;">niedostateczna</td> </tr> <tr> <td>A ∈ [50% x, 60% x)</td> <td style="text-align: right;">dostateczna</td> </tr> <tr> <td>A ∈ [60% x, 70% x)</td> <td style="text-align: right;">dostateczna plus</td> </tr> <tr> <td>A ∈ [70% x, 80% x)</td> <td style="text-align: right;">dobra</td> </tr> <tr> <td>A ∈ [80% x, 90% x)</td> <td style="text-align: right;">db plus</td> </tr> <tr> <td>A ∈ [90% x, 100% x]</td> <td style="text-align: right;">bardzo dobra</td> </tr> </table> <p>(CL) Ćwiczenia laboratoryjne</p> <p>-praca projektowa nr 1 dotycząca tworzenia specyfikacji wymagań- efekty: U_01, U_02 , K_01, K_02 (30%)</p> <p>- praca projektowa nr 2 dotycząca studium wykonalności projektów informatycznych - efekty: U_02 , K_01, K_02 (20%)</p> <p>- praca projektowa nr 3 dotycząca modelowania oprogramowania na bazie wymagań - efekty: U_03, U_04,U_05, K_01, K_02 (50%)</p> <p>Każda z form oceny CL jest punktowana, a suma punktów możliwych do uzyskania to y.</p> <p>W nawiasach przy każdej z form oceniania CL podano jaki procent y można uzyskać maksymalnie z tej formy oceniania. Ocena B dla zaliczenia CL jest ustalana na podstawie uzyskanej przez studenta sumy punktów P według zasady:</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">P ∈ [0% y, 50% y)</td> <td style="text-align: right;">niedostateczna</td> </tr> </table>	A ∈ [0% x, 50% x)	niedostateczna	A ∈ [50% x, 60% x)	dostateczna	A ∈ [60% x, 70% x)	dostateczna plus	A ∈ [70% x, 80% x)	dobra	A ∈ [80% x, 90% x)	db plus	A ∈ [90% x, 100% x]	bardzo dobra	P ∈ [0% y, 50% y)	niedostateczna
A ∈ [0% x, 50% x)	niedostateczna														
A ∈ [50% x, 60% x)	dostateczna														
A ∈ [60% x, 70% x)	dostateczna plus														
A ∈ [70% x, 80% x)	dobra														
A ∈ [80% x, 90% x)	db plus														
A ∈ [90% x, 100% x]	bardzo dobra														
P ∈ [0% y, 50% y)	niedostateczna														

	<p>P ∈ [50% y, 60% y) dostateczna</p> <p>P ∈ [60% y, 70% y) dostateczna plus</p> <p>P ∈ [70% y, 80% y) dobra</p> <p>P ∈ [80% y, 90% y) db plus</p> <p>P ∈ [90% y, 100% y] bardzo dobra</p> <p>Końcowa ocena z zaliczenia przedmiotu jest wyliczona jako średnia ważona ocen otrzymanych za wykład (A) i ćwiczenia laboratoryjne (B), dla których wagami są przypisane im liczby punktów ECTS.</p>
--	---

Matryca efektów kształcenia dla przedmiotu

Numer (symbol) efektu kształcenia	Odniesienie do efektów kształcenia dla programu	Odniesienie do charakterystyki drugiego stopnia PRK dla obszaru/ obszarów
W_01	K1_W24	P6S_WG
W_02	K1_W24	P6S_WG
W_03	K1_W18, K1_W22, K1_W24	P6S_WG
U_01	K1_U06, K1_U09	P6S_UW
U_02	K1_U30	P6S_UW
U_03	K1_U04, K1_U30, K1_U31	P6S_UW
U_04	K1_U04, K1_U32	P6S_UW
U_05	K1_U26	P6S_UW
K_01	K1_02, K1_08	P6S_KO
K_02	K1_01, K1_08	P6S_KK, P6S_KO

Wykaz literatury

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

1. Beatty J., Wiegiers K. *Specyfikacja oprogramowania. Inżynieria wymagań*, Helion, Gliwice 2014
2. Chrabski B., Żmitrowicz K. *Inżynieria wymagań w praktyce* PWN, Warszawa 2015
3. Sacha K. *Inżynieria oprogramowania*, PWN, Warszawa 2010
4. Shalloway A., Trott J.R. *Programowanie zorientowane obiektowo. Wzorce projektowe*. Helion, Gliwice 2005
5. Wrycza S., Marcinkowski B., Wyrzykowski K. *Język UML 2.0 w modelowaniu systemów informatycznych, modelowanie biznesowe, metodyki projektowe oparte na UML, narzędzia CASE*, Helion, Gliwice 2005

B. Literatura uzupełniająca

1. Adzic G. *Specyfikacja na przykładach. Poznaj zwinne metody pracy i dostarczaj właściwe oprogramowanie.*, Helion, Gliwice 2014
2. Jaskiewicz A. *Inżynieria oprogramowania*, Helion. Gliwice 1997

Kontakt

dr inż. Zbigniew Ledóchowski, dr Ryszard Motyka
 zbigniew.ledochowski@apsl.edu.pl, ryszard.motyka@apsl.edu.pl