

**KARTA PRZEDMIOTU****I. Dane podstawowe**

Nazwa przedmiotu	Inżynieria oprogramowania
Nazwa przedmiotu w języku angielskim	Software engineering
Kierunek studiów	Informatyka
Poziom studiów (I, II, jednolite magisterskie)	I
Forma studiów (stacjonarne, niestacjonarne)	stacjonarne
Dyscyplina	Informatyka
Język wykładowy	polski

Koordinator przedmiotu/osoba odpowiedzialna	Rafał Lizut
---	-------------

Forma zajęć ( <i>katalog zamknięty ze słownika</i> )	Liczba godzin	semestr	Punkty ECTS
wykład	30	5	5
konwersatorium			
ćwiczenia			
laboratorium	30	5	
warsztaty			
seminarium			
proseminarium			
lektorat			
praktyki			
zajęcia terenowe			
pracownia dyplomowa			
translatorium			
wizyta studyjna			

Wymagania wstępne	Znajomość programowania strukturalnego i obiektowego, język angielski
-------------------	---

**II. Cele kształcenia dla przedmiotu**

Podniesienie poziomu wiedzy studentów w zakresie inżynierii oprogramowania
Przedstawienie i szczegółowe omówienie wszystkich aspektów tworzenia oprogramowania od początkowej fazy specyfikacji aż do jego pielęgnacji po dacie rozpoczęcia użytkowania
Zapoznanie studentów i wyrobienie u nich umiejętności pracy zgodnie z metodykami strukturalną, obiektową i zwinną

### III. Efekty uczenia się dla przedmiotu wraz z odniesieniem do efektów kierunkowych

Symbol	Opis efektu przedmiotowego	Odniesienie do efektu kierunkowego
<b>WIEDZA</b>		
W_01	Student wie, czym jest inżynieria oprogramowania, proces tworzenia oprogramowania, zarządzanie przedsięwzięciami	K_W04, K_W06
W_02	Student wie, jak powinny być stawiane wymagania oprogramowaniu, jak wygląda proces inżynierii wymagań, modelowanie systemu, prototypowanie oprogramowania, weryfikacja, testowanie i odbiór zatwierdzonego oprogramowania	K_W04, K_W06
W_03	Student wie jakie są metody zarządzanie personelem, zarządzania jakością, szacowania oprogramowania, ulepszania oprogramowania	K_W04, K_W06
<b>UMIEJĘTNOŚCI</b>		
U_01	Student konstruuje wymagania niefunkcjonalne i sporządzać specyfikację oprogramowania	K_U02, K_U04, K_U13, K_U14, K_U17, K_U23, K_U29, K_U30
U_02	Student posługuje się diagramami opisu struktury i zachowań programu	K_U02, K_U04, K_U13, K_U14, K_U23, K_U30
U_03	Student korzysta z języka UML	K_U02, K_U04, K_U13, K_U14, K_U30
U_04	Student opracowuje plan przedsięwzięcia dotyczącego budowy oprogramowania	K_U02, K_U04, K_U13, K_U14, K_U17, K_U29, K_U30
U_05	Student kontroluje i zarządza wersjami tworzonego oprogramowania oraz stosuje się do zasad obowiązujących programistów podczas pracy w zespole	K_U02, K_U04, K_U13, K_U17, K_U23, K_U30
<b>KOMPETENCJE SPOŁECZNE</b>		
K_01	Student uznaje złożoność problemów, z którymi może spotkać się w życiu	K_K01, K_K02, K_K04, K_K05
K_02	Student umiejętnie rozwiązuje problemy inżynierii oprogramowania stosując poznane metody oraz obiektywnie ocenia uzyskane wyniki	K_K01, K_K02, K_K04, K_K05
K_03	Student pracuje zarówno indywidualnie, jak i zespołowo, właściwie planując pracę swoją i zespołu w kontekście postawionego celu	K_K01, K_K02, K_K04, K_K05

### IV. Opis przedmiotu/ treści programowe

<ul style="list-style-type: none"> <li>1 Wprowadzenie</li> <li>2 Procesy wytwarzania oprogramowania</li> <li>3 Inżynieria wymagań</li> <li>4 Metody strukturalne</li> <li>5 Metody obiektowe i podstawy UML</li> <li>6 Jakość kodu, inspekcje kodu</li> <li>7 Testowanie</li> <li>8 Dokumentacja użytkowa</li> </ul>
--

9 Konserwacja  
 10 Systemy krytyczne  
 11 Metody formalne  
 12 Wzorce projektowe

#### V. Metody realizacji i weryfikacji efektów uczenia się

Symbol efektu	Metody dydaktyczne <i>(lista wyboru)</i>	Metody weryfikacji <i>(lista wyboru)</i>	Sposoby dokumentacji <i>(lista wyboru)</i>
<b>WIEDZA</b>			
W_01	Wykład konwencjonalny/ Wykład problemowy	Egzamin	Test / Sprawdzian pisemny
W_02	Wykład konwencjonalny/ Wykład problemowy	Egzamin	Test / Sprawdzian pisemny
W_03	Wykład konwencjonalny/ Wykład problemowy	Egzamin	Test / Sprawdzian pisemny
<b>UMIEJĘTNOŚCI</b>			
U_01	Metoda projektu design thinking	Przygotowanie / wykonanie projektu	Karta oceny projektu
U_02	Metoda projektu design thinking	Przygotowanie / wykonanie projektu	Karta oceny projektu
U_03	Metoda projektu design thinking	Przygotowanie / wykonanie projektu	Karta oceny projektu
U_04	Metoda projektu design thinking	Przygotowanie / wykonanie projektu	Karta oceny projektu
U_05	Metoda projektu design thinking	Przygotowanie / wykonanie projektu	Karta oceny projektu
<b>KOMPETENCJE SPOŁECZNE</b>			
K_01	Metoda projektu design thinking	Przygotowanie / wykonanie projektu	Karta oceny projektu
K_02	Metoda projektu design thinking	Przygotowanie / wykonanie projektu	Karta oceny projektu
K_03	Metoda projektu design thinking	Przygotowanie / wykonanie projektu	Karta oceny projektu

#### VI. Kryteria oceny, wagi...

90 – 100% - bardzo dobry (5.0),  
 80 – 89% - dobry plus (4.5),  
 70 – 79% - dobry (4.0),  
 60 – 69% - dostateczny plus (3.5),  
 50 – 59% - dostateczny (3.0),  
 Poniżej 50% - niedostateczny (2.0).

#### Obciążenie pracą studenta

Forma aktywności studenta	Liczba godzin
Liczba godzin kontaktowych z nauczycielem	<b>80</b>

Liczba godzin indywidualnej pracy studenta	60
--	----

## VII. Literatura

Literatura podstawowa
1. K. Sacha, Inżynieria oprogramowania, PWN, Warszawa 2010 2. A. Jaskiewicz, Inżynieria oprogramowania, Helion, Gliwice 1997 3. I. Sommerville, Inżynieria oprogramowania, WNT, Warszawa 2003
Literatura uzupełniająca
1. L. Miękina, Inżynieria oprogramowania, Wydawnictwo AGH, Kraków 2009 2. P. Stevens, UML. Inżynieria oprogramowania, Helion, Gliwice 2007 3. J. Myers, C. Sandler, T. BadgeQ, T. M. Thomas, Sztuka testowania oprogramowania, Helion, Gliwice 2005 4. R. C. MarSn. Czysty kod. Podręcznik dobrego programisty, Helion, Gliwice 2010 5. Bruegge B., Dutoit A. H., Inżynieria oprogramowania w ujęciu obiektowym. UML, wzorce projektowe i Java, Helion, Gliwice 2011 6. <a href="http://wazniak.mimuw.edu.pl">hQp://wazniak.mimuw.edu.pl</a>