

(pieczęć wydziału)

KARTA PRZEDMIOTU

1. Nazwa przedmiotu:		2. Kod przedmiotu:		
INŻYNIERIA OPROGRAMOWANIA		MK_17		
3. Karta przedmiotu ważna od roku akademickiego: 2013/2014				
4. Forma kształcenia:		studia pierwszego stopnia		
5. Forma studiów:		studia stacjonarne		
6. Kierunek studiów:		INFORMATYKA (RAU)		
7. Profil studiów:		ogólnoakademicki		
8. Specjalność: -				
9. Semestr:		4, 5		
10. Jednostka prowadząca przedmiot:		Instytut Informatyki		
11. Prowadzący przedmiot:		dr inż. Przemysław Szmał, dr inż. Jacek Szedel		
12. Przynależność do grupy przedmiotów:		przedmioty wspólne		
13. Status przedmiotu:		obowiązkowy		
14. Język prowadzenia zajęć:		polski		
15. Przedmioty wprowadzające oraz wymagania wstępne: Przedmioty wprowadzające: Podstawy programowania komputerów, Programowanie Komputerów				
16. Cel przedmiotu: Celem przedmiotu jest przedstawienie wybranych zagadnień wchodzących w zakres inżynierii programowania, ze szczególnym uwzględnieniem tych, które odnoszą się do cyklu życia jednostek oprogramowania o ograniczonych rozmiarach. Między innymi studenci zapoznają się ogólnie z problematyką projektowania systemów informatycznych z wykorzystaniem nowoczesnych metodologii i narzędzi CASE.				
17. Efekty kształcenia:¹				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
1	Ogólna wiedza zakresu inżynierii oprogramowania	SP	WT	K1A_W08 K1A_W12 K1A_W14
2	Umiejętność modelowania systemów informatycznych	RP	P	K1A_U09
3	Umiejętność projektowania oprogramowania	RP	P	K1A_U20 K1A_U22
4	Umiejętność posługiwania się narzędziem CASE	RP	P	K1A_U20
5	Umiejętność testowania i poprawiania wydajności oprogramowania	CL	L	K1A_U14
18. Formy zajęć dydaktycznych i ich wymiar (liczba godzin)				
	W.	L.	P.	
Sem. 4	30	15	-	
Sem. 5	-	-	30	

¹ należy wskazać ok. 5 – 8 efektów kształcenia

19. Treści kształcenia:

Treść wykładów:

Wprowadzenie, geneza i przedmiot inżynierii programowania. Kryzys oprogramowania, modele cyklu życia oprogramowania. Zarys inżynierii systemów, zarządzanie projektem informatycznym. Faza strategiczna. Faza określania wymagań. Zastosowanie narzędzi CASE w fazie strategicznej i w fazie formułowania wymagań. Faza analizy, projekt systemu. Metodyki strukturalne. Metodyki obiektowe (przed powstaniem UML). Język i metodyka UML. Implementacja i testowanie. Niezawodność oprogramowania.

Laboratorium (wykaz ćwiczeń laboratoryjnych):

Testowanie i uruchamianie. Ocena i poprawa efektywności oprogramowania Komunikacja programów z użytkownikiem.

Projekt (etapy realizacji projektu):

1. Modelowanie wymagań
 - a. Instrukcja dotycząca metodyki projektu
 - b. Instrukcja posługiwania się wybranym narzędziem CASE
 - c. Przygotowanie wstępnej specyfikacji wymagań
2. Modelowanie słownictwa i struktury
 - a. Identyfikacja słownictwa, sporządzenie słownika projektu
 - b. Kontrola spójności słownika ze specyfikacją wymagań
3. Model przypadków użycia
 - a. Realizacja warsztatów przypadków użycia
 - b. Asocjowanie przypadków użycia z wymaganiami
 - c. Wprowadzenie modelu do narzędzia CASE
4. Realizacja przypadków użycia
 - a. Wybór fragmentu do realizacji
 - b. Budowa interakcji
5. Projektowanie i generowanie kodu
 - a. Zaprojektowanie klas i interfejsów dla wskazanej realizacji przypadku użycia
 - b. Parametryzacja generatora kodu
 - c. Generowanie kodu
6. Kompilacja i implementacja
 - a. Uruchamianie skompilowanego szablonu projektu
 - b. Implementacja wybranej interakcji

20. Egzamin: nie

21. Literatura podstawowa:

- Ben-Ari, Podstawy programowania współbieżnego, WNT, Warszawa 1996.
- I. Sommerville: "Inżynieria Oprogramowania", WNT 2003
- K. Sacha: "Inżynieria oprogramowania", Wydawnictwo Naukowe PWN, Warszawa, 2010
- S. Wrycza, B. Marcinkowski, K. Wyrzykowski: „Język UML 2.0 w modelowaniu systemów informatycznych”, Helion, Gliwice, 2006.
- W. Dąbrowski, A. Stasiak, M. Wolski: „Modelowanie systemów informatycznych w języku UML 2.1”, PWN, Warszawa 2009.
- G. Booch, J. Rumbaugh, I. Jacobson: "UML. Przewodnik użytkownika", WNT, Warszawa, 2001, 2002
- P. Szał (red.): "Inżynieria programowania. Metody i ćwiczenia laboratoryjne", Wydawnictwo Politechniki Śląskiej, Gliwice, 2003

22. Literatura uzupełniająca:

- R.S. Pressman, "Praktyczne podejście do inżynierii oprogramowania", WNT 2004
- A.Hunt, D.Thomas, „Pragmatyczny programista, od czeladnika do mistrza”, WNT, 2002
- 3. J. Górski (red.): „Inżynieria oprogramowania w projekcie informatycznym”, wyd. II rozszerzone. Mikom, Warszawa 2000
- 4. A. Jaszkievicz: "Inżynieria oprogramowania", Helion, 1997
- 5. G. Schneider, J. Winters, „Stosowanie przypadków użycia”, WNT 2004

23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia

Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1	Wykład	30 / 30
2	Ćwiczenia	- / -
3	Laboratorium	15 / 15
4	Projekt	30 / 30
5	Seminarium	- / -
6	Inne (egzamin)	- / -
	Suma godzin	75 / 75

24. Suma wszystkich godzin: 150**25. Liczba punktów ECTS:² 6 w tym 3 (sem.4), 3 (sem.5)****26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego: 3****27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty): 3****26. Uwagi: -**

Zatwierdzono:

² 1 punkt ECTS – 25 - 30 godzin.

.....
(data i podpis prowadzącego)

.....
(data i podpis dyrektora instytutu/kierownika katedry/
Dyrektora Kolegium Języków Obcych/kierownika lub
dyrektora jednostki międzywydziałowej)