

(pieczęć wydziału)

KARTA PRZEDMIOTU

1. Nazwa przedmiotu: INŻYNIERIA OPROGRAMOWANIA 2. Kod przedmiotu: IO-NSM				
3. Karta przedmiotu ważna od roku akademickiego: 2012				
4. Forma kształcenia: studia drugiego stopnia				
5. Forma studiów: studia niestacjonarne				
6. Kierunek studiów: INFORMATYKA, WYDZIAŁ AEII				
7. Profil studiów: ogólnoakademicki				
8. Specjalność: –				
9. Semestr: 1				
10. Jednostka prowadząca przedmiot: Instytut Informatyki, RAU2				
11. Prowadzący przedmiot: dr inż. Krzysztof Simiński				
12. Przynależność do grupy przedmiotów: przedmioty wspólne				
13. Status przedmiotu: obowiązkowy				
14. Język prowadzenia zajęć: polski				
15. Przedmioty wprowadzające oraz wymagania wstępne: Inżynieria oprogramowania (I stopień), Programowanie komputerów (I stopień). Zakłada się, że student posiada przygotowanie w zakresie programowania w języku C++.				
16. Cel przedmiotu: Celem przedmiotu jest rozszerzenie wiadomości i umiejętności nabytych w trakcie kursu Inżynierii Oprogramowania na stopniu inżynierskim (IO-NSI). Studenci zapoznają się z zaawansowanymi aspektami notacji UML, wzorcami projektowymi (wg <i>Gang of Four</i>) oraz problemami zarządzania jakością oprogramowania.				
17. Efekty kształcenia: K_W06 Ma wiedzę z zakresu modelowania i analizy systemów informatycznych rozumianych jako systemy komputerowe i sieci teleinformatyczne oraz związanych z nimi artefaktów. K_W14 Ma szczegółową wiedzę w zakresie algorytmów w tym algorytmów współbieżnych, języków i paradygmatów programowania, inżynierii programowania i sztucznej inteligencji. K_W16 Ma podstawową wiedzę dotyczącą zarządzania, w tym zarządzania jakością i prowadzenia działalności gospodarczej. K_U17 Potrafi – zgodnie z zadaną specyfikacją, uwzględniając aspekty pozatechniczne – zaprojektować złożone urządzenie, system lub proces informatyczny oraz zrealizować ten projekt – co najmniej w części – używając właściwych metod, technik i narzędzi, w tym przystosowując do tego istniejące lub opracowując nowe narzędzia. K_U18 Potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego z zakresu informatyki, w tym dostrzec ograniczenia tych metod i narzędzi. K_K03 Potrafi współdziałać i pracować w grupie przyjmując różne role. K_K06 Potrafi myśleć i działać w sposób kreatywny i przedsiębiorczy.				
nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
1	Potrafi zamodelować strukturę systemu z wykorzystaniem UML	Laboratorium I	laboratorium	K_W06 K_W14 K_U17 K_U18 K_K03 K_K06
2	Potrafi zamodelować działanie systemu z wykorzystaniem UML	Laboratorium II	laboratorium	K_W06 K_W14 K_U17 K_U18 K_K03 K_K06
3	Potrafi wykorzystać zaawansowane aspekty UML w modelowaniu systemu	Laboratorium III	laboratorium	K_W06 K_W14 K_U17 K_U18 K_K03 K_K06

4	potrafi zastosować strukturalne wzorce projektowe	Laboratorium IV	laboratorium	K_W14 K_U17 K_U18 K_K03 K_K06
5	potrafi zastosować kreatywne wzorce projektowe	Laboratorium V	laboratorium	K_W14 K_U17 K_U18 K_K03 K_K06
6	potrafi zastosować behawioralne wzorce projektowe	Laboratorium VI	laboratorium	K_W14 K_U17 K_U18 K_K03 K_K06
7	Zna UML w stopniu umożliwiającym zaawansowane modelowanie systemu	Kolokwium	wykład	K_W06 K_W14
8	Zna własności, wady, zalety i zastosowanie wzorców projektowych	Kolokwium	wykład	K_W06 K_W14
9	Zna metody i problemy zarządzania jakością	Kolokwium	wykład	K_W16 K_U18
10	Zna narzędzia Ishikawy	Kolokwium	wykład	K_W16 K_U18

18. Formy zajęć dydaktycznych i ich wymiar (liczba godzin)

15 W. 15 Lab.

19. Treści kształcenia:

Wykład:

zaawansowane aspekty modelowania z wykorzystaniem UML, wzorce projektowe, zarządzanie jakością oprogramowania

Ćwiczenia:

zaawansowane aspekty modelowania z wykorzystaniem UML, wzorce projektowe

20. Egzamin: nie

21. Literatura podstawowa:

1. S. Wrycza, B. Marcinkowski, K. Wyrzykowski: Język UML 2.0 w modelowaniu systemów informatycznych Helion, Gliwice, 2006.
2. G. Booch, J. Rumbaugh, I. Jacobson: UML. Przewodnik użytkownika, WNT, Warszawa, 2001, 2002.
3. I. Sommerville: Inżynieria Oprogramowania, WNT 2003.
4. Craig Larman: UML i wzorce projektowe, Helion 2011
5. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Wzorce projektowe, Helion 2010

22. Literatura uzupełniająca:

1. R.S. Pressman: Praktyczne podejście do inżynierii oprogramowania, WNT 2004
2. W. Dąbrowski, A. Stasiak, M. Wolski: Modelowanie systemów informatycznych w języku UML 2.1, PWN, Warszawa 2009
3. A. Jaskiewicz: Inżynieria oprogramowania, Helion, 1997.

23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia		
Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1	Wykład	15/15
2	Ćwiczenia	- / -
3	Laboratorium	15 / 15
4	Projekt	- / -
5	Seminarium	- / -
6	Inne	- / -
	Suma godzin	30/30
24. Suma wszystkich godzin: 60		
25. Liczba punktów ECTS: 3		
26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego: 1		
27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty): 2		
26. Uwagi:		

Zatwierdzono:

.....
(data i podpis prowadzącego)

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