

SYLABUS (KARTA PRZEDMIOTU/MODUŁU)

Nazwa przedmiotu/modułu (zgodna z zatwierdzonym programem studiów na kierunku) Inżynieria oprogramowania		Punkty ECTS 3	Numer katalogowy
Nazwa w j. angielskim Software engineering			
Jednostka(i) realizująca(e) przedmiot/moduł (instytut/katedra) Instytut Inżynierii Biosystemów			
Kierownik przedmiotu/modułu dr inż. Janina Rudowicz-Nawrocka			
Kierunek studiów Informatyka stosowana	Poziom Studia I stopnia	Profil ogólnoakademicki	Semestr 4 S / 5 N
Specjalność -	Specjalizacja inżynierska -		
RODZAJE ZAJĘĆ I ICH WYMIAR GODZINOWY (zajęcia zorganizowane i praca własna studenta)			
Forma studiów: stacjonarne		Forma studiów: stacjonarne	
- wykłady	15	- wykłady	10
- ćwiczenia laboratoryjne	30	- ćwiczenia laboratoryjne	20
- inne z udziałem nauczyciela	10	- inne z udziałem nauczyciela	10
- praca własna	25	- praca własna	40
Łączna liczba godzin: 80		Łączna liczba godzin: 80	
CEL PRZEDMIOTU/MODUŁU			
<p>Celem przedmiotu jest uzyskanie przez studentów wiedzy obejmującej podstawowe pojęcia i metody z zakresu inżynierii oprogramowania na wszystkich etapach wytwarzania oprogramowania; uporządkowanie wiedzy z zakresu metod i systemów notacji w procesie analizy dziedziny problemu, metod testowania oprogramowania, oceny jakości i ryzyka oraz zarządzania w projekcie informatycznym; nabycie umiejętności przygotowywania projektu informatycznego; pozyskanie kompetencji społecznych w zakresie zespołowego przygotowywania projektu z uwzględnieniem zasad organizacji pracy zespołowej i podziału ról; rozwinięcie umiejętności komunikowania się w procesie zespołowego rozwiązywania problemów.</p> <p>Po zakończonym kursie student nabywa wiedzę oraz praktyczne umiejętności pozwalające na samodzielne i zespołowe wytworzenie projektu systemu informatycznego oraz nadzorowanie jego realizacji, z uwzględnieniem problematyki produkcji rolno-spożywczej.</p>			
METODY DYDAKTYCZNE			
<ol style="list-style-type: none"> Wykłady – przedstawienie wiedzy teoretycznej w postaci oryginalnych prezentacji multimedialnych. Wykłady – prezentacja i analiza poszczególnych fragmentów projektów systemów informatycznych. Ćwiczenia w sali komputerowej – samodzielne opracowywanie fragmentów projektów informatycznych. Ćwiczenia – omówienie i analiza zagadnień związanych z realizowanymi w trakcie ćwiczeń projektami. 			
EFEKTY KSZTAŁCENIA			Odniesienie do efektów kierunkowych
Wiedza	E1. Zna zasady wytwarzania systemów informatycznych w oparciu o podstawowe modele procesu tworzenia oprogramowania. E2. Ma wiedzę z zakresu formułowania wymagań funkcjonalnych i pozafunkcjonalnych w obszarze dziedziny problemu tworzonego systemu informatycznego. E3. Ma wiedzę dotyczącą obiektowego modelowania dziedziny problemu w notacji UML w aspektach statycznym i dynamicznym. E4. Zna zasady reprezentacji modeli i analizy dziedziny problemu z wykorzystaniem wszystkich diagramów dostępnych w notacji UML. E5. Ma wiedzę z zakresu metod testowania oprogramowania. E6. Ma wiedzę odnośnie wdrażania i utrzymywania systemów informatycznych. E7. Ma wiedzę o wiodących metodach zarządzania przedsięwzięciem informatycznym. E8. Ma wiedzę o metodach oceny i zapewniania jakości tworzonych systemów. E9. Zna metody analizy ryzyka w zakresie procesu wytwarzania oprogramowania.		IS1A_W16
Umiejętności	E10. Umie sformułować zestaw wymagań funkcjonalnych i pozafunkcjonalnych dla dowolnej dziedziny problemu, ze wsparciem specjalisty z tej dziedziny. E11. Umie zbudować model dziedziny problemu z wykorzystaniem notacji UML. E12. Potrafi zaprojektować system informatyczny zgodnie z zasadami inżynierii oprogramowania. E13. Umie przygotować schemat zarządzania przedsięwzięciem informatycznym. E14. Potrafi opracować zestaw przypadków testowych i przeprowadzić testy oprogramowania. E15. Umie dokonać analizy jakości procesu i produktu informatycznego. E16. Potrafi przeprowadzić analizę ryzyka w realizacji projektu informatycznego.		IS1A_U09 IS1A_U11
Kompetencje społeczne	E17. Ma świadomość możliwości metod inżynierii oprogramowania oraz zakresu ich zastosowania. E18. Wykazuje kreatywność w zakresie projektowania systemów informatycznych z wykorzystaniem metod inżynierii oprogramowania. E19. Ma umiejętności współpracy w zespole w zakresie opracowania wymagań, dokumentacji modelu i projektu oraz implementacji i testowania systemu informatycznego. E20. Ma świadomość dynamicznie rozwijających się metod tworzenia oprogramowania i konieczności ciągłego doskonalenia.		IS1A_K05 IS1A_K07

Metody weryfikacji efektów kształcenia 1. Kontrola programów realizowanych w ramach ćwiczeń. 2. Kolokwia przy komputerze sprawdzające umiejętności praktyczne. 3. Projekt w postaci aplikacji komputerowej o wysokim poziomie złożoności. 4. Egzamin ustny z wykorzystaniem opracowanego projektu.	Numery efektów E1 - E9 E10 - E16 E17 - E20
TREŚCI KSZTAŁCENIA	
<ol style="list-style-type: none"> 1. Wprowadzenie do inżynierii oprogramowania. Metody i narzędzia CASE. Modele procesu wytwarzania oprogramowania. 2. Planowanie przedsięwzięcia informatycznego. Określenia wymagań stawianych tworzonemu oprogramowaniu: wymagania funkcjonalne i pozafunkcjonalne, dokumentowanie wymagań. 3. Analiza dziedziny problemu – wprowadzenie: model dziedziny problemu, metodyki i systemy notacji, modelowanie strukturalne - modele funkcji i modele danych, modelowanie obiektowe: rodzaje diagramów w notacji UML. 4. Analiza dziedziny problemu - modelowanie obiektowe: diagram przypadków użycia: elementy notacji, analiza aktorów i przypadków użycia. 5. Analiza dziedziny problemu - diagram przypadków użycia: analiza relacji między przypadkami użycia, dokumentowanie przypadków użycia, przykłady. 6. Analiza dziedziny problemu - diagramy klas: reprezentacja klasy, składnia pól klasy, przykłady klas, dziedziczenie - specjalizacja i uogólnienie, aspekt specjalizacji, etykiety, dziedziczenie wielokrotne, dziedziczenie dynamiczne. 7. Analiza dziedziny problemu - diagramy klas: wzorce klas (klasy sparametryzowane), wystąpienie klas (obiekt), klasa abstrakcyjna i klasa konkretna, interfejsy i realizacje oraz metody ich graficznego odwzorowywania na diagramach, atrybuty i metody - specyfikacja i rodzaje, polimorfizm, wiązania statyczne i dynamiczne, przeciążanie metod. 8. Analiza dziedziny problemu - diagramy klas: powiązania dotyczące klas i obiektów, oznaczenia dotyczące powiązań - nazwa, kierunek odczytu, liczebność, powiązania skierowane, powiązania zwrotne, role, agregacja, agregacja całkowita, powiązania kwalifikowane, ograniczenia, przykłady złożonych diagramów klas. 9. Analiza dziedziny problemu - diagramy maszyny stanów i czynności: symbole i przykłady. 10. Analiza dziedziny problemu - diagramy kolejności, komunikacji, harmonogramowania i przeglądu interakcji: symbole i przykłady. 11. Analiza dziedziny problemu - diagramy komponentów, rozmieszczenia, pakietów i struktur złożonych: symbole i przykłady. 12. Projektowanie, implementacja i dokumentacja oprogramowania: opis wykonywanych zadań i tworzonych dokumentów, określenie zasad tworzenia dokumentacji, omówienie elementów składowych dokumentacji, przykłady. 13. Testowanie oprogramowania: podstawowe pojęcia, poziomy i rodzaje testowania, techniki wykonywania testów. 14. Instalacja i konserwacja oprogramowania. Zarządzanie przedsięwzięciem informatycznym. 15. Zarządzanie jakością i zarządzanie ryzykiem w projekcie informatycznym. 	
Formy i kryteria zaliczenia przedmiotu/modułu 1. Ćwiczenia: Kolokwium zaliczeniowe w formie praktycznej polegającej na opracowaniu wybranych fragmentów projektu informatycznego. 2. Ćwiczenia: Projekt zaliczeniowy zespołowy wybranego systemu informatycznego. 3. Wykłady: Egzamin ustny połączony z analizą wykonanego projektu.	Procentowy udział w końcowej ocenie 70% oceny z ćwiczeń 30% oceny z wykładów 100% oceny z wykładów
WYKAZ LITERATURY	
<ol style="list-style-type: none"> 1. Sommerville I. 2003. Inżynieria oprogramowania, WNT, Warszawa. 2. Wrycza S., Marcinkowski B., Wyrzykowski K. 2006. Język UML 2.0 w modelowaniu systemów informatycznych, Helion, Gliwice. 3. Wrycza S. (red.). 2007. UML 2.1, Helion, Gliwice. 4. Weres J. 2016. Inżynieria oprogramowania, Wykłady udostępnione w sieci. IIB WRiB UP Poznań. 5. Sommerville I. 2015. Software Engineering, 10th ed., Addison Wesley, Upper Saddle River, NJ. 6. Dąbrowski W., Subieta K. 2005. Podstawy inżynierii oprogramowania, Wyd. PJWSTK. 7. Śmiałek M. 2005. Zrozumieć UML 2.0. Metody modelowania obiektowego. Helion, Gliwice. 8. Graessle P., Baumann H., Baumann P. 2005. UML 2.0 w akcji. Przewodnik oparty na projektach. Helion, Gliwice. 9. Larman C. 2011. UML i wzorce projektowe. Analiza i projektowanie obiektowe oraz iteracyjny model wytwarzania aplikacji, wyd. 3. Helion, Gliwice. 10. Martin R.C., Martin M. 2008. Agile Principles, Patterns, and Practices in C#. Helion, Gliwice. 	