



UNIWERSYTET
IM. ADAMA MICKIEWICZA
W POZNANIU

Inżynieria oprogramowania Sylabus zajęć

Informacje podstawowe

Kierunek studiów Informatyka Specjalność - Jednostka organizacyjna Wydział Matematyki i Informatyki Poziom studiów studia inżynierskie pierwszego stopnia Forma studiów studia niestacjonarne Profil studiów profil ogólnoakademicki		Cykl dydaktyczny 2023/24 Kod zajęć 06INFN.310P.02426.23 Języki wykładowe polski Obligatoryjność Obowiązkowy Blok zajęciowy Przedmioty podstawowe
Koordynator zajęć	Jacek Marciniak	
Prowadzący zajęcia	Jacek Marciniak	
Okres Semestr 5	Forma zajęć / liczba godzin / forma zaliczenia - Wykład: 15, Zaliczenie z oceną - Laboratorium: 15, Zaliczenie z oceną	Liczba punktów ECTS 4

Cele kształcenia dla zajęć

Kod	Cel
C1	Zapoznanie studentów z tematyką inżynierii oprogramowania.
C2	Prezentacja podstawowych zasad oraz narzędzi wykorzystywanych podczas rozwoju oprogramowania.
C3	Omówienie tematyki analizy wymagań, projektowania, implementacji i testowania oprogramowania.
C4	Zaprezentowanie procesów utrzymania i pielęgnacji oprogramowania.
C5	Zaprezentowanie podstawowych zasad zarządzania zespołem programistycznym.

Wymagania wstępne

Wymagana jest podstawowa wiedza i umiejętności w zakresie programowania w wybranych językach programowania oraz znajomość podstawowych konstrukcji programistycznych i implementacji algorytmów. Umiejętność pracy z materiałami dodatkowymi oraz samodzielnego pozyskiwania informacji i wyciągania wniosków.

Efekty uczenia się dla zajęć

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
Wiedzy - Student/ka:			
W1	Zna problemy pojawiające się podczas tworzenia oprogramowania.	INF_K3_W03, INF_K3_W04_inz	Test, Projekt
W2	Zna różne podejścia do tworzenia oprogramowania i związane z nimi środowiska projektowo-programistyczne.	INF_K3_W03, INF_K3_W04_inz, INF_K3_W06_inz, INF_K3_W11_inz	Test, Projekt
W3	Zna metody i narzędzia pozwalające na zbieranie wymagań oraz projektowanie oprogramowania zgodnie z metodyką obiektową.	INF_K3_W03, INF_K3_W04_inz	Test, Projekt
W4	Zna metody i narzędzia wykorzystywane w procesie budowy oprogramowania.	INF_K3_W03, INF_K3_W04_inz	Test, Projekt
W5	Zna metody, techniki i narzędzia wykorzystywane w procesie projektowania systemów informatycznych.	INF_K3_W03, INF_K3_W04_inz, INF_K3_W06_inz, INF_K3_W11_inz	Test, Projekt
W6	Zna techniki prowadzące do otrzymania oprogramowania wysokiej jakości.	INF_K3_W03, INF_K3_W04_inz	Test, Projekt
W7	Zna wybrane wzorce projektowe.	INF_K3_W03, INF_K3_W04_inz	Test, Projekt
Umiejętności - Student/ka:			
U1	Potrafi stworzyć projekt prostego systemu informatycznego w wybranym narzędziu (np. w języku UML).	INF_K3_U02, INF_K3_U04_inz, INF_K3_U06_inz, INF_K3_U08, INF_K3_U09	Projekt, Raport
U2	Potrafi oceniać przydatność różnych podejść do rozwoju oprogramowania i związanych z nimi środowisk programistycznych do rozwiązywania problemów pojawiających na różnych etapach budowy oprogramowania.	INF_K3_U02, INF_K3_U04_inz, INF_K3_U06_inz, INF_K3_U08, INF_K3_U09	Projekt, Raport
U3	Potrafi projektować oprogramowanie zgodnie z metodyką obiektową.	INF_K3_U02, INF_K3_U04_inz, INF_K3_U06_inz, INF_K3_U08, INF_K3_U09	Projekt, Raport
U4	Potrafi oceniać przydatność metod i narzędzi wykorzystywanych w procesie budowy oprogramowania oraz wybrać i zastosować właściwą metodę i narzędzia do rozwiązania typowych zadań związanych z rozwojem oprogramowania.	INF_K3_U02, INF_K3_U04_inz, INF_K3_U06_inz, INF_K3_U08, INF_K3_U09	Projekt, Raport

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
U5	Potrafi zgodnie z zadaną specyfikacją zaprojektować oraz zbudować prosty system informatyczny, używając właściwych metod, technik i narzędzi.	INF_K3_U02, INF_K3_U04_inz, INF_K3_U06_inz, INF_K3_U08, INF_K3_U09	Projekt, Raport
U6	Potrafi stosować techniki prowadzące do otrzymania oprogramowania wysokiej jakości.	INF_K3_U02, INF_K3_U04_inz, INF_K3_U06_inz, INF_K3_U08, INF_K3_U09	Projekt, Raport
U7	Potrafi posługiwać się wzorcami projektowymi w procesie budowy oprogramowania.	INF_K3_U02, INF_K3_U04_inz, INF_K3_U06_inz, INF_K3_U08, INF_K3_U09	Projekt, Raport
Kompetencji społecznych - Student/ka:			
K1	Jest świadomy/a roli inżynierii oprogramowania w procesie budowy oprogramowania wysokiej jakości.	INF_K3_K03, INF_K3_K05, INF_K3_K06	Test, Projekt
K2	Rozumie proces ewolucji narzędzi i metod inżynierii oprogramowania.	INF_K3_K03, INF_K3_K06	Test, Projekt

Treści programowe dla zajęć

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
1.	Czym jest oprogramowanie, problemy pojawiające się przy rozwoju oprogramowania.	W1, K1	Wykład
2.	Cykl życia oprogramowania z uwzględnieniem różnych modeli. Wyróżnienie składowych procesu rozwoju oprogramowania. Analiza modeli cyklu życia oprogramowania na przykładzie metodyki hybrydowej (Rational Unified Process) oraz zwinnej (SCRUM).	W2, U2, U4, U6, K1, K2	Wykład, Laboratorium
3.	Analiza wymagań: wymagania funkcjonalne i нефункционалне, specyfikacja wymagań, proces pozyskiwania wymagań (scenariusze, przypadki użycia, user stories), zarządzanie wymaganiami. Zbieranie wymagań przy pomocy przypadków użycia oraz user stories	W3, U4, U5, U6, K1	Wykład, Laboratorium
4.	Projektowanie systemu informatycznego. UML w projektowaniu. Projektowanie interakcji (diagramy interakcji), modelowanie struktury systemu (model logiczny, diagram klas, diagram wdrożenia), modelowania zachowania (diagram stanów, diagram aktywności).	W4, W5, W6, U1, U2, U3, U4, U6, K1	Wykład, Laboratorium
5.	Architektura systemu informatycznego, wzorce architektoniczne (klient-serwer, aplikacje wielowarstwowe), wzorce projektowe (MVC, ORM). Wpływ architektury systemu na tworzone oprogramowanie (przykłady).	W4, W5, W6, U1, U2, U5, U6, U7, K1, K2	Wykład, Laboratorium
6.	Analiza i projektowanie obiektowe: model wiedzy dziedzinowej, identyfikacja klas, przypisywanie obiektom odpowiedzialności, wielokrotne użycie.	W3, W4, W5, W7, U1, U3, U4, U5, U6, K1	Wykład, Laboratorium

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
7.	Testowanie oprogramowania. Rodzaje testów. Testy prowadzone przez programistę (testy jednostkowe, testy modułowe, testy integracyjne). Test-driven development. Ciągła integracja. Scenariusze testowe i testy akceptacyjne. Testy użyteczności.	W2, W4, W6, U2, U4, U5, U6, K1, K2	Wykład, Laboratorium
8.	Utrzymanie i pielęgnacja oprogramowania.	W2, W4, W6, U2, U4, U6, K1, K2	Wykład, Laboratorium
9.	Zarządzanie projektem informatycznym: zarządzanie zespołem, planowanie prac, zarządzanie jakością, zarządzanie wersjami.	W1, W2, W4, W5, W6, U2, U4, U5, U6, K1, K2	Wykład, Laboratorium
10.	Wykorzystanie narzędzi wspierających rozwój oprogramowania (repozytorium kodu, issue tracker, bug tracker).	W2, W4, W6, U2, U4, U5, U6, K1, K2	Wykład, Laboratorium
11.	Realizacja grupowego mini-projektu w wybranej metodyce. Realizacja w dużym zespole z podziałem na role.	W1, W2, W3, W4, W5, W6, W7, U1, U2, U3, U4, U5, U6, U7, K1, K2	Laboratorium

Informacje dodatkowe

Forma zajęć	Metody i formy prowadzenia zajęć
Wykład	Wykład z prezentacją multimedialną wybranych zagadnień, Dyskusja
Laboratorium	Metoda laboratoryjna, Metoda projektu, Praca w grupach

Forma zajęć	Warunki zaliczenia zajęć
Wykład	Warunkiem zaliczenia jest przystąpienie do testów w trakcie zajęć. Warunkiem zaliczenia wykładu jest zaliczenia laboratoriów. Skala ocen: 1. bardzo dobry (bdb; 5,0) – od 90% punktów, 2. dobry plus (db plus; 4,5) – od 80% punktów, 3. dobry (db; 4,0) – od 70% punktów, 4. dostateczny plus (dst plus; 3,5) – od 60% punktów, 5. dostateczny (dst; 3,0) – od 50% punktów, 6. niedostateczny (ndst; 2,0) – poniżej 50% punktów
Laboratorium	Na ocenę końcową składają się punkty z projektów i zadań (raportów) cząstkowych. Skala ocen: 1. bardzo dobry (bdb; 5,0) – od 90% punktów, 2. dobry plus (db plus; 4,5) – od 80% punktów, 3. dobry (db; 4,0) – od 70% punktów, 4. dostateczny plus (dst plus; 3,5) – od 60% punktów, 5. dostateczny (dst; 3,0) – od 50% punktów, 6. niedostateczny (ndst; 2,0) – poniżej 50% punktów.

Literatura

Obowiązkowa

1. Sommerville, I. Inżynieria oprogramowania. Wydawnictwo naukowe PWN, 2020

Dodatkowa

1. Larman, C. Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and the Unified Process. Englewood Cliff, NJ: Prentice Hall, 2002
2. Gamma, E., Helm, R., Johnson, R. and Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, Mass.: Addison-Wesley, 1995.
3. Jorgensen, P. C., Software Testing: A Craftsman's Approach, CRC Press, Taylor&Francis Group, 2014
4. E. Yourdon, Mityczny osobomiesiąc, WNT, Warszawa 2001
5. Kan, S. Metryki i modele w inżynierii jakości oprogramowania, WN PWN, Warszawa, 2006
6. Bruegge, B. Dutoit, A.H. Inżynieria oprogramowania w ujęciu obiektowym. UML, wzorce projektowe i Java. (Helion 2011)

Nakład pracy studenta i punkty ECTS

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
Wykład	15
Laboratorium	15
Przygotowanie raportu	10
Przygotowanie projektu	35
Przygotowanie do zaliczenia	15
Czytanie wskazanej literatury	10
Łączny nakład pracy studenta	Liczba godzin 100
Liczba punktów ECTS	ECTS 4

* godzina (lekcyjna) oznacza 45 minut

Efekty uczenia się dla kierunku

Kod	Treść
INF_K3_K03	Absolwent/ka jest gotów/gotowa do pełnienia roli zawodowej informatyka ze świadomością ciągłych zmian stosowanych paradygmatów i technologii
INF_K3_K05	Absolwent/ka jest gotów/gotowa do przedstawiania w sposób przystępny podstawowych zagadnień z zakresu informatyki, porozumiewania się przy użyciu słownictwa technicznego w środowisku zawodowym, również w języku angielskim oraz z wykorzystaniem narzędzi informatycznych
INF_K3_K06	Absolwent/ka jest gotów/gotowa do uczestniczenia w procesach gospodarczych związanych z informatyką i świadczeniem wybranych usług informatycznych
INF_K3_U02	Absolwent/ka potrafi pozyskiwać wiarygodne informacje z literatury, baz wiedzy, Internetu oraz innych źródeł, integrować je, interpretować oraz wyciągać wnioski i formułować opinie
INF_K3_U04_inz	Absolwent/ka potrafi opracować, przeanalizować, zaprojektować klasyczne algorytmy i systemy informatyczne
INF_K3_U06_inz	Absolwent/ka potrafi ocenić, na podstawowym poziomie, przydatność metod i narzędzi informatycznych oraz wybrać i zastosować właściwą metodę i narzędzia do typowych zadań informatycznych
INF_K3_U08	Absolwent/ka potrafi posługiwać się językiem angielskim zgodnie z wymaganiami określonymi dla poziomu B2 Europejskiego Systemu Opisu Kształcenia Językowego oraz zna język angielski w stopniu umożliwiającym czytanie ze zrozumieniem dokumentacji oprogramowania, podręczników i artykułów informatycznych
INF_K3_U09	Absolwent/ka potrafi przygotować dokumentację, opracowania i raporty w języku polskim i języku angielskim, w tym z wykorzystaniem podstawowych ujęć teoretycznych, a także różnych źródeł
INF_K3_W03	Absolwent/ka zna i rozumie narzędzia, technologie i urządzenia informatyczne właściwe dla wybranych obszarów zastosowań oraz podstawy ich działania
INF_K3_W04_inz	Absolwent/ka zna i rozumie zaawansowane pojęcia, konstrukcje i procesy związane z językami programowania i inżynierią programowania
INF_K3_W06_inz	Absolwent/ka zna i rozumie zaawansowane pojęcia związane z informatyczną architekturą sprzętowo-programową
INF_K3_W11_inz	Absolwent/ka zna i rozumie problemy dotyczące prawnych i społecznych aspektów informatyki, w tym odpowiedzialności zawodowej i etycznej oraz zasad tworzenia i rozwoju różnych form przedsiębiorczości, w tym indywidualnej