KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu
Projektowanie i modelowanie oprogramowania [S2Inf1E-IO>PMO]

Przedmiot

Kierunek studiów
Informatyka/Computing

Rok/Semestr
1/1

Studia w zakresie (specjalność)
Inżynieria oprogramowania

Profil studiów
ogólnoakademicki

Poziom studiów
drugiego stopnia

Język oferowanego przedmiotu
angielski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
30

Laboratorium
30

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

4,00

Koordynatorzy

dr hab. inż. Bartosz Walter prof. PP
bartosz.walter@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z podstaw programowania oraz inżynierii oprogramowania. Powinien również posiadać umiejętność pozyskiwania informacji ze wskazanych źródeł oraz mieć gotowość do podjęcia współpracy w ramach zespołu.

Cel przedmiotu

Przekazanie studentom wiedzy z metod modelowania oraz obiektowego projektowania systemów informatycznych z wykorzystaniem dobrych praktyk oraz wzorców projektowych. Rozwijanie u studentów umiejętności oceny jakości projektu oraz stosowania wybranych mechanizmów dostępnych w obiektowych językach programowania.

Przedmiotowe efekty uczenia się

Wiedza:
1. posiada ugruntowaną wiedzę teoretyczną dotyczącą cyklu rozwojowego systemu informatycznego
2. posiada wiedzę na temat wybranych metod, języków i notacji modelowania oprogramowania
3. posiada wiedzę na temat wzorców projektowych oraz dobrych praktyk w zakresie projektowania oprogramowania

4. zna wybrane metody pomiaru niektórych charakterystyk oprogramowania (np. rozmiaru, złożoności)

Umiejętności:
1. potrafi zaprojektować system informatyczny korzystając z mechanizmów i cech języków obiektowych
2. potrafi dokonać oceny jakości projektu systemu informatycznego
3. potrafi stworzyć model oprogramowania korzystając z wybranych cech języka uml

Kompetencje społeczne:
1. potrafi współpracować w grupie
2. potrafi poszerzać swoją wiedzę, korzystając z dostępnych źródeł i dokonując ich selekcji

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:
Wiedza przedstawiona w ramach wykładu weryfikowana jest poprzez realizację podczas wykładu, w ramach współpracy w grupach, dwóch zadań projektowych oraz egzamin końcowy -- test wielokrotnego wyboru sprawdzający stopień zrozumienia i przyswojenia treści wykładowych. Oceny uzyskane z tych form są uśredniane z wagami 30% i 70%. Próg zaliczeniowy wynosi 50%. Zagadnienia zaliczeniowe zostaną przedstawione podczas ostatniego wykładu w ramach przedmiotu.
Umiejętności nabyte w ramach laboratorium weryfikowane są poprzez realizację w grupach 3-4 projektów, dotyczących poszczególnych zagadnień omawianych w trakcie zajęć. Próg zaliczeniowy wynosi 50%.

Treści programowe

1. Wykład: Przegląd metod i zagadnień związanych z programowaniem obiektowym. Metody modelowania oprogramowania. Testowanie jednostkowe. Pomiary i metryki związane z kodem programów. Wzorce projektowe. Programowanie aspektowe. Programowanie funkcyjne. Zasada odwrócenia sterowania.
2. Laboratorium: Modelowanie z wykorzystaniem kart CRC oraz elementów języka UML. Testowanie jednostkowe programów obiektowych. Metryki oprogramowania. Dobór i implementacja wzorców projektowych. Zastosowanie wybranych paradygmatów programowania w praktyce. Implementacja odwróconego sterowania.

Tematyka zajęć

1. Wykład: Przegląd mechanizmów i technik związanych z programowaniem obiektowym. Metody modelowania oprogramowania (karty CRC, UML, C4). Testowanie jednostkowe (JUnit, TestNG). Pomiary i metryki związane z kodem
programów (MOOD, CK Suite, metryki Martina, prawo Demeter). Wzorce projektowe. Programowanie aspektowe (AspectJ). Programowanie funkcyjne. Zasada odwrócenia sterowania (IoC)
2. Laboratorium: Modelowanie z wykorzystaniem kart CRC i/lub C4. Testowanie
jednostkowe programów obiektowych z wykorzystaniem typowych bibliotek. Zbieranie i analiza statycznych metryk oprogramowania. Dobór i implementacja wzorców
projektowych. Zastosowanie wybranych paradygmatów programowania w praktyce. Dwa zadania do samodzielnego (lub grupowego) rozwiązania

Metody dydaktyczne

1. Wykład: prezentacja multimedialna
2. Laboratorium: prezentacja ilustrowana przykładami podanymi na tablicy, wykonanie w grupach zadań przedstawionych przez prowadzącego, dyskusja nad rozwiązaniami

Literatura

Podstawowa
1. E. Gamma et al.: Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku. Helion, 2012
2. R. C. Martin: Clean code. A Handbook of agile software craftmanship. Prentice Hall, 2008
3. B. Eckel: Thinking in Java. Edycja polska. Wydanie IV. Helion, 2017.

Uzupełniająca
1. B. Meyer: Programowanie zorientowane obiektowo (Second Edition). Helion, 2005.
2. J. Backfield: Programowanie funkcyjne. Krok po kroku. Helion, 2015

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwiów/egzaminu, wykonanie projektu)	40	1,50