



KARTA MODUŁU / KARTA PRZEDMIOTU

Kod modułu	E-1IZ2-1004-s2
Nazwa modułu	Technologie obiektowe
Nazwa modułu w języku angielskim	Object-oriented technologies
Obowiązuje od roku akademickiego	2017/2018

A. USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	II stopień (I stopień / II stopień)
Profil studiów	Ogólnoakademicki (ogólno akademicki / praktyczny)
Forma i tryb prowadzenia studiów	niestacjonarne (stacjonarne / niestacjonarne)
Specjalność	Systemu informacyjne
Jednostka prowadząca moduł	Katedra Systemów Informatycznych Zakład Informatyki
Koordinator modułu	Mariusz Bedla
Zatwierdził:	Dziekan WEAiI Dr hab. inż. Antoni Różowicz, prof. PŚk

B. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	Kierunkowy (podstawowy / kierunkowy / inny HES)
Status modułu	obowiązkowy (obowiązkowy / nieobowiązkowy)
Język prowadzenia zajęć	polski
Usytuowanie modułu w planie studiów - semestr	2
Usytuowanie realizacji przedmiotu w roku akademickim	Semestr letni (semestr zimowy / letni)
Wymagania wstępne	Programowanie w języku Java Inżynieria programowania Bazy danych 2 Projektowanie aplikacji internetowych 2 (kody modułów / nazwy modułów)
Egzamin	nie (tak / nie)
Liczba punktów ECTS	4

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	inne
w semestrze	18			18	



C. EFEKTY KSZTAŁCENIA I METODY SPRAWDZANIA EFEKTÓW KSZTAŁCENIA

Cel modułu	Poznanie podstawowych cechy i wynikających z nich zasad dotyczących paradygmatu programowania obiektowego, zagadnień dotyczących przechowywania obiektów oraz wybranych zagadnień dotyczących inżynierii oprogramowania
-------------------	---

Symbol efektu	Efekty kształcenia	Forma prowadzenia zajęć (w/c/l/p/inne)	odniesienie do efektów kierunkowych	odniesienie do efektów obszarowych
W_01	Znajomość podstawowych cechy i wynikające z nich zasad dotyczących paradygmatu programowania obiektowego.	W	K_W10	T2A_W04 T2A_W05
W_02	Znajomość wybranych zagadnień dotyczących inżynierii oprogramowania	W	K_W10	T2A_W04 T2A_W05
W_03	Znajomość zagadnień dotyczących przechowywania obiektów	W	K_W10	T2A_W04 T2A_W05
U_01	Umiejętność projektowania i implementacji programów wykorzystujących technologie obiektowe.	P	K_U17	T2A_U19
K_01	Umiejętność projektowania i implementacji programów w zespołach programistycznych.	P	K_K02	T2A_K03

Treści kształcenia:

1. Treści kształcenia w zakresie wykładu

Nr wykładu	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1 - 2	Podstawowe cechy i wynikające z nich zasady dotyczące paradygmatu programowania obiektowego	W_01
3	Metryki obiektowe	W_01, W_02
4	Refaktoryzacja	W_01, W_02
5	Refaktoryzacja do wzorców projektowych	W_01, W_02
6	Relacyjno - obiektowe bazy danych	W_01, W_03
7	Obiektowe bazy danych	W_01, W_03
8	Testowanie systemów obiektowych	W_01, W_02
9	Kolokwium zaliczeniowe	W_01, W_02, W_03

2. Charakterystyka zadań projektowych

Tematyka zadań projektowych obejmuje stworzenie aplikacji wraz z dokumentacją na wybrany temat dotyczący zagadnień związanych z technologiami obiektowymi.

Metody sprawdzania efektów kształcenia

Symbol efektu	Metody sprawdzania efektów kształcenia (sposób sprawdzenia, w tym dla umiejętności – odwołanie do konkretnych zadań projektowych, laboratoryjnych, itp.)
W_01	Kolokwium zaliczeniowe
W_02	Kolokwium zaliczeniowe
W_03	Kolokwium zaliczeniowe
U_01	Zaliczenie na podstawie sprawozdania z projektu, prawidłowego podziału zadań w grupie, stopnia realizacji zadań oraz odpowiedzi.
K_01	Zaliczenie na podstawie wyników realizacji zagadnień w ramach projektu oraz odpowiedzi wykazującej stopień zaangażowania w pracę zespołu.



D. NAKŁAD PRACY STUDENTA

Bilans punktów ECTS		
	Rodzaj aktywności	obciążenie studenta
1	Udział w wykładach	18
2	Udział w ćwiczeniach	
3	Udział w laboratoriach	
4	Udział w konsultacjach (2-3 razy w semestrze)	2
5	Udział w zajęciach projektowych	18
6	Konsultacje projektowe	
7	Udział w egzaminie	
8		
9	Liczba godzin realizowanych przy bezpośrednim udziale nauczyciela akademickiego	38 (suma)
10	Liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczyciela akademickiego (1 punkt ECTS=25-30 godzin obciążenia studenta)	1,25
11	Samodzielne studiowanie tematyki wykładów	20
12	Samodzielne przygotowanie się do ćwiczeń	
13	Samodzielne przygotowanie się do kolokwium	20
14	Samodzielne przygotowanie się do laboratoriów	
15	Wykonanie sprawozdań	
15	Przygotowanie do kolokwium końcowego z laboratorium	
17	Wykonanie projektu lub dokumentacji	42
18	Przygotowanie do egzaminu	
19		
20	Liczba godzin samodzielnej pracy studenta	82 (suma)
21	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy (1 punkt ECTS=25-30 godzin obciążenia studenta)	2,75
22	Sumaryczne obciążenie pracą studenta	120
23	Punkty ECTS za moduł 1 punkt ECTS=25-30 godzin obciążenia studenta	4
24	Nakład pracy związany z zajęciami o charakterze praktycznym Suma godzin związanych z zajęciami praktycznymi	60
25	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym 1 punkt ECTS=25-30 godzin obciążenia studenta	2



E. LITERATURA

Wykaz literatury	1. Inżynieria oprogramowania. Metody obiektowe w teorii i w praktyce. Ian Graham, WNT, Warszawa 2004 2. Słownik terminów z zakresu obiektowości. Kazimierz Subieta, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1999 3. Metryki i modele w inżynierii jakości oprogramowania, Stephen H. Kan, PWN, 2006 4. Refaktoryzacja Ulepszanie struktury istniejącego kodu, Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts, Helion, 2011 5. Refaktoryzacja do wzorców projektowych, Joshua Kerievsky, Helion, 2005 6. Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku, Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides, Helion 2010 7. Eric Freeman, Bert Bates, Kathy Sierra, Elisabeth Robson "Wzorce projektowe. Rusz głową!" Helion, 2010 8. Thomas Connolly, Carolyn Begg, Systemy baz danych, Wydawnictwo RM, 2004 9. Ramez Elmasri, Shamkant B. Navathe, Wprowadzenie do systemów baz danych, Helion, 2005 10. Jason Price, Oracle Database 12c i SQL. Programowanie, Helion, 2015 11. Michael McLaughlin, Oracle Database 12c. Programowanie w języku PL/SQL, Helion, 2015 12. Java Data Objects, Sameer Tyagi, Keiron McCammon, Michael Vorburger, Heiko Bobzin, Helion, 2004 13. The Object Database Standard: ODMG 3.0, R.G.G. Cattell, Douglas K. Barry, Morgan Kaufmann Publishers, Inc. San Francisco, California 14. Binder Robert V. "Testowanie systemów obiektowych", WNT, 2010 15. Roman Adam, Testowanie i jakość oprogramowania. Modele, techniki, narzędzia, PWN, 2017 16. Miles R., Hamilton K.: "UML 2.0", Helion, 2007. 17. Pilone D., Pitman N.: "UML 2.0 Almanach", Helion, 2007. 18. Wrycza S., Marcinkowski B.: "Język UML 2.0 w modelowaniu systemów informatycznych", Helion, 2006 19. Larman C. "UML i wzorce projektowe. Analiza i projektowanie obiektowe oraz iteracyjny model wytwarzania aplikacji", Wydanie III Helion, 2011 20. Zarządzanie projektami IT. Przewodnik po metodykach, Adam Koszlajda, helion, 2012 21. Szyperski C. "Oprogramowanie komponentowe. Obiekty to za mało" WNT, 2001
Witryna WWW modułu/przedmiotu	Strona na platformie Moodle: https://www.weaii-moodle.tu.kielce.pl/course/view.php?id=101