



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	E-1IZ2-1004-s2
Nazwa przedmiotu	Technologie obiektowe
Nazwa przedmiotu w języku angielskim	Object-oriented technologies
Obowiązuje od roku akademickiego	2019/20

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	
Poziom kształcenia	
Profil studiów	
Forma i tryb prowadzenia studiów	
Zakres	
Jednostka prowadząca przedmiot	
Koordynator przedmiotu	dr inż. Mariusz Bedla
Zatwierdził	Dziekan Wydziału Elektrotechniki Automatyki i Informatyki Dr hab. inż. Antoni Różowicz, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	
Status przedmiotu	
Język prowadzenia zajęć	
Usytuowanie modułu w planie studiów - semestr	
Wymagania wstępne	
Egzamin (TAK/NIE)	
Liczba punktów ECTS	

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	Inne
Liczba godzin w semestrze	18	0	0	18	0

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Znajomość podstawowych cech i wynikających z nich zasad dotyczących paradygmatu programowania obiektowego.	INF2_W10
	W02	Znajomość wybranych zagadnień dotyczących inżynierii oprogramowania.	INF2_W10
	W03	Znajomość zagadnień dotyczących przechowywania obiektów.	INF2_W10
Umiejętności	U01	Umiejętność projektowania i implementacji programów wykorzystujących technologie obiektowe.	INF2_U19
Kompetencje społeczne	K01	Umiejętność projektowania i implementacji programów w zespołach programistycznych.	INF2_K02

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
Wykład	1. - 2. Podstawowe cechy i wynikające z nich zasady dotyczące paradygmatu obiektowego
	3. Metryki obiektowe
	4. Refaktoryzacja kodu w językach obiektowych
	5. Refaktoryzacja do wzorców projektowych
	6. Obiektowość i bazy danych
	7. Testowanie systemów obiektowych
	8. Metodyki wykorzystywane podczas tworzenia systemów obiektowych
	9. Kolokwium zaliczeniowe
projekt	Tematyka zadań projektowych obejmuje stworzenie aplikacji wraz z dokumentacją na wybrany temat dotyczący zagadnień związanych z technologiami obiektowymi.

*) zostawić tylko realizowane formy zajęć

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			X			
W02			X			
W03			X			
U01				X	X	
K01				X	X	

A.

FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład		Uzyskanie co najmniej 50% punktów z kolokwium zaliczeniowego
projekt		Uzyskanie co najmniej 50% punktów na podstawie sprawozdania z projektu, prawidłowego podziału zadań w grupie, stopnia realizacji zadań oraz odpowiedzi.

*) zostawić tylko realizowane formy zajęć

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
1.	Udział w zajęciach zgodnie z planem studiów	W	C	L	P	S	h
		30			30		
3.	Inne (konsultacje, egzamin)*	4			2		h
4.	Razem przy bezpośrednim udziale nauczyciela akademickiego	42					h
5.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	1,68					ECTS
6.	Liczba godzin samodzielnej pracy studenta	58					h
7.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	2,32					ECTS
8.	Nakład pracy związany z zajęciami o charakterze praktycznym	18					h
9.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,71					ECTS
10.	Sumaryczne godzinowe obciążenie pracą studenta	100					h
11.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	4					

* wszelkie formy weryfikacji efektów, w tym egzaminy oraz nie więcej niż 2 godziny konsultacji dla każdej formy zajęć

LITERATURA

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
22. Farcic V., Garcia A. "TDD Programowanie w Javie sterowane testami", Helion, 2016
23. „BDD w działaniu. Sterowanie zachowaniem w rozwoju aplikacji”, John Ferguson Smart, Helion, 2016
24. Fowler M., "Architektura systemów zarządzania przedsiębiorstwem. Wzorce projektowe", Helion, 2005.

Uwaga: wykaz literatury winien uwzględniać aktualne i dostępne publikacje