



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	E-1IZ3-06-s6
Nazwa przedmiotu	Inżynieria Programowania
Nazwa przedmiotu w języku angielskim	Software Engineering
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. Arkadiusz Chrobot
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	Programowanie obiektowe (Java)
Egzamin (TAK/NIE)	
Liczba punktów ECTS	

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	Inne
Liczba godzin w semestrze	18	0	18	9	0

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Zna zagadnienia związane z zarządzaniem projektem informatycznym.	INF1_W15 INF1_W16
	W02	Zna zagadnienia związane z projektowaniem i tworzeniem systemów informatycznych.	INF1_W15
	W03	Zna zagadnienia związane z weryfikacją i zatwierdzeniem oprogramowania.	INF1_W16
	W04	Zna semantykę języka UML.	INF1_W15
	W05	Zna wzorce projektowe GoF: konstrukcyjne, strukturalne, czynnościowe.	INF1_W15
Umiejętności	U01	Potrafi czytać diagramy w notacji UML oraz używać ich do modelowania oprogramowania.	INF1_U19
	U02	Potrafi praktycznie zastosować wzorce projektowe GoF.	INF1_U19 INF1_U23
	U03	Potrafi pracować zespołowo nad tworzeniem oprogramowania, przyjmując w zespole różne role.	INF1_U19 INF1_U24 INF1_U25 INF1_U27
Kompetencje społeczne	K01	Potrafi obiektywnie ocenić swoją rolę w projekcie, zna wiążącą się z nią odpowiedzialność.	INF1_K03 INF1_K04
	K02	Potrafi obiektywnie ocenić pomysły swoje oraz innych osób.	INF1_K03 INF1_K04 INF1_K05
	K03	Potrafi rozpoznać potrzeby osób spoza branży informatycznej w zakresie funkcji tworzonego oprogramowania.	INF1_K06 INF1_K07

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	1. Wprowadzenie do inżynierii oprogramowania
	2. Unified Modeling Language (diagramy strukturalne, behawioralne i interakcji)
	3. Inżynieria wymagań.
	4. Projektowanie architektoniczne oprogramowania.
	5. Wzorce projektowe GoF – konstrukcyjne, strukturalne i operacyjne
	6. Zatwierdzanie i weryfikacja oprogramowania.
	7. Testowanie oprogramowania.
laboratorium	1. Modelowanie perspektywy przypadków użycia systemu.
	2. Modelowanie perspektywy projektowej systemu – wzorce konstrukcyjne.
	3. Modelowanie perspektywy projektowej systemu – wzorce strukturalne.
	4. Modelowanie perspektywy projektowej systemu – wzorce czynnościowe.
	5. Modelowanie perspektywy implementacyjnej systemu.
projekt	Studenci wykonują zadanie w zespole pięcioosobowym. Każdy z członków zespołu ma przyporządkowaną określoną rolę (analityk, architekt, developer, tester, serwisant). Przedmiotem zadania jest wykonanie systemu zarządzania firmą lub organizacją (np. hotel, akademik, zajezdnia autobusowa). Wykonanie zadania wymaga przeprowadzenia typowych czynności realizowanych w ramach projektu tworzenia systemu informatycznego: specyfikacji wymagań, projektowania architektury oprogramowania, implementacji, opracowania testów i posadowienia oraz dostosowania systemu do potrzeb klienta.

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

	Metody sprawdzania efektów uczenia się
--	--

Symbol efektu	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01		x				
W02		x				
W03		x				
W04		x				
W05		x				
U01				x		x
U02				x		x
U03				x		x
K01				x		x
K02				x		x
K03				x		x

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład		Uzyskanie co najmniej 50% punktów z egzaminu.
laboratorium		Uzyskanie co najmniej 50% punktów z wykonywanych zadań.
projekt		Uzyskanie co najmniej 50% za wykonane oprogramowanie, dokumentację do niego i obronę projektu.

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
		W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	18		18	9		h
3.	Inne (konsultacje, egzamin)	3		2	1		h
4.	Razem przy bezpośrednim udziale nauczyciela akademickiego	51					h
5.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	2,04					ECTS
6.	Liczba godzin samodzielnej pracy studenta	74					h
7.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	2,96					ECTS
8.	Nakład pracy związany z zajęciami o charakterze praktycznym	27					h
9.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	2,65					ECTS
10.	Sumaryczne godzinowe obciążenie pracą studenta	125					h
11.	Punkty ECTS za moduł	5					

LITERATURA

1. Ian Sommerville „Inżynieria oprogramowania", WNT, Warszawa 2003
2. Włodzimierz Dąbrowski, Andrzej Stasiak, Michał Wolski „Modelowanie systemów informatycznych w języku UML 2.1 w praktyce", PWN, Warszawa 2009
3. Gamma Erich, Helm Richard, Johnson Ralph, Vlissides John „Wzorce projektowe. Elementy oprogramowania wielokrotnego użytku", Helion, Gliwice 2017
4. Krzysztof Sacha „Inżynieria oprogramowania", PWN, Warszawa 2010
5. Mariusz Flasiński „Zarządzanie projektami informatycznymi", PWN, Warszawa 2006
6. Robert C. Martin, „Czysty kod. Podręcznik dobrego programisty", Helion, Gliwice 2010
7. Adam Roman „Testowanie i jakość oprogramowania", PWN, Warszawa 2017
8. Adam Roman, Karolina Zmitrowicz „Testowanie oprogramowania w praktyce", PWN, Warszawa 2018
9. Radosław Smilgin „Zawód tester", PWN, Warszawa 2016