



KARTA PRZEDMIOTU

Kod przedmiotu	studia stacjonarne:	
	studia niestacjonarne:	
Nazwa przedmiotu	Inżynieria Oprogramowania 2	
Nazwa przedmiotu w języku angielskim	Software Engineering 2	
Obowiązuje od roku akademickiego	2022/23	

USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	I stopień
Profil studiów	ogólnoakademicki
Forma i tryb prowadzenia studiów	studia stacjonarne i niestacjonarne
Zakres	wszystkie specjalności
Jednostka prowadząca przedmiot	Katedra Systemów Informatycznych
Koordynator przedmiotu	dr inż. Arkadiusz Chrobot dr inż. Adam Krechowicz
Zatwierdził	Dziekan Wydziału Elektrotechniki, Automatyki i Informatyki dr hab. inż. Roman Deniziak, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów		przedmiot kierunkowy
Status przedmiotu		obowiązkowy
Język prowadzenia zajęć		polski
Usytuowanie w planie studiów - semestr	studia stacjonarne	semestr VI
	studia niestacjonarne	semestr VI
Wymagania wstępne		Inżynieria Oprogramowania 1
Egzamin (TAK/NIE)		NIE
Liczba punktów ECTS		1

Forma prowadzenia zajęć		wykład	ćwiczenia	laboratorium	projekt	inne
Liczba godzin w semestrze	studia stacjonarne:				15	
	studia niestacjonarne:				9	

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty uczenia się	Odniesienie do efektów kierunkowych
Wiedza	W01	Zna i rozumie zagadnienia związane z pracą w zespole opracowującym projekt informatyczny.	INF1_W20
	W02	Zna i rozumie zagadnienia związane z projektowaniem i tworzeniem oprogramowania.	INF1_W20
	W03	Zna i rozumie zagadnienia związane z weryfikacją i zatwierdzaniem oprogramowania.	INF1_W20
	W04	Zna i rozumie semantykę języka UML.	INF1_W20
	W05	Zna i rozumie wzorce projektowe GoF: konstrukcyjne, strukturalne, czynnościowe.	INF1_W20
Umiejętności	U01	Potrafi czytać diagramy w notacji UML oraz używać ich do modelowania oprogramowania.	INF1_U20
	U02	Potrafi praktycznie zastosować wzorce projektowe GoF.	INF1_U20
	U03	Potrafi piastować różne role w zespole tworzącym oprogramowanie.	INF1_U20
Kompetencje społeczne	K01	Jest gotów do obiektywnej oceny swojej roli w projekcie informatycznym, zna wiążącą się z nią odpowiedzialność.	INF1_K02
	K02	Jest gotów do obiektywnej oceny pomysłów swoich oraz innych osób.	INF1_K02
	K03	Jest gotów do rozpoznawania potrzeb osób spoza branży informatycznej w zakresie funkcji tworzonego oprogramowania.	INF1_K01

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
projekt	Studenci wykonują zadanie w wieloosobowym zespole. Jego przedmiotem jest wykonanie oprogramowania o stopniu złożoności wymagającym zastosowania wiedzy i metod z zakresu współczesnej inżynierii oprogramowania. Może to być np. system do zarządzania firmą lub organizacją (np. hotel, akademik, zajezdnia autobusowa). Wykonanie zadania wymaga od studentów przeprowadzenia typowych czynności realizowanych w ramach projektu tworzenia oprogramowania, takich jak specyfikacja wymagań, projektowanie architektury, implementacji, opracowania testów. W trakcie zajęć każdy ze studentów powinien poznać metody pracy zespołu wytwarzającego oprogramowanie, typowe role poszczególnych jego członków oraz zakres ich odpowiedzialności.

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		
W04				X		
W05				X		
U01				X		
U02				X		

U03				X		
K01				X		
K02				X		
K03				X		

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
projekt	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów za wykonane oprogramowanie, dokumentację do niego i obronę projektu.

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS												
Lp.	Rodzaj aktywności	Obciążenie studenta										Jednos tka
		studia stacjonarne					studia niestacjonarne					
1.	Udział w zajęciach zgodnie z planem studiów	W	C	L	P	S	W	C	L	P	S	h
					15					9		
2.	Inne (konsultacje, egzamin)				1					1		h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	16					10					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	0,64					0,4					ECTS
5.	Liczba godzin samodzielnej pracy studenta	9					15					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	0,36					0,6					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	15					9					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	0,6					0,36					ECTS
9.	Sumaryczne obciążenie pracą studenta	25					25					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	1										ECTS

LITERATURA

1. Ian Sommerville „Inżynieria oprogramowania”, PWN, Warszawa 2020
2. Krzysztof Sacha „Inżynieria oprogramowania”, PWN, Warszawa 2010
3. Mariusz Flasiński „Zarządzanie projektami informatycznymi”, PWN, Warszawa 2006
4. Adam Roman „Testowanie i jakość oprogramowania”, PWN, Warszawa 2015
5. Robert C. Martin „Czysty kod. Podręcznik dobrego programisty”, Helion, Gliwice 2010
6. Robert C. Martin „Czysta architektura. Struktura i design oprogramowania. Przewodnik dla profesjonalistów”, Helion, Gliwice 2018

7. Adam Roman, Karolina Zmitrowicz „Testowanie oprogramowania w praktyce”, PWN, Warszawa 2017
8. Radosław Smilgin „Zawód tester”, PWN, Warszawa, 2016
9. Erich Gamma, Richard Helm, John Vlissides, Ralph Johnson „Wzorce projektowe: Elementy oprogramowania obiektowego wielokrotnego użytku” Helion, Gliwice 2021