



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	E-2IZ2-1008-s4
Nazwa przedmiotu	Systemy odporne na błędy
Nazwa przedmiotu w języku angielskim	Fault tolerant systems
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	Grzegorz Łukawski
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	Architektura systemów komputerowych, Programowanie systemów rozproszonych, Inżynieria programowania
Egzamin (TAK/NIE)	
Liczba punktów ECTS	

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

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Student potrafi wymienić rodzaje błędów występujących w systemach komputerowych, rozumie potrzebę zapewnienia odporności na błędy.	INF2_W04 INF2_W05
	W02	Student zna podstawowe mechanizmy tolerowania błędów sprzętowych.	INF2_W05
	W03	Student zna podstawowe mechanizmy tolerowania błędów programowych.	INF2_W05
Umiejętności	U01	Student potrafi zaproponować metodę zapewnienia odporności na błędy dla wybranego problemu sprzętowego lub programowego.	INF2_U01 INF2_U06 INF2_U08
Kompetencje społeczne	K01	Student umie podzielić problem programistyczny na elementy i współpracować w grupie przy jego implementacji.	INF2_K01 INF2_K02

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	1. Rodzaje błędów, redundancja, odległość Hamminga, rodzaje systemów odpornych na błędy.
	2-4. Metody detekcji i korekcji błędów sprzętowych.
	5-9. Metody detekcji i korekcji błędów programowych.
laboratorium	1-2. Metody tolerowania błędów sprzętowych – symulacja programowa.
	3-4. Metody tolerowania błędów programowych.
projekt	1-4. Zadanie projektowe polegające na przygotowaniu aplikacji realizującej wybrane algorytmy odporności na błędy programowe lub sprzętowe, na drodze symulacji.

*) 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 końcowego.
laboratorium		Uzyskanie co najmniej 50% punktów za wykonane zadania laboratoryjne oraz z kolokwium końcowego, kolokwium w trakcie zajęć lub kolokwium końcowego i kolokwium w trakcie zajęć.
projekt		Uzyskanie co najmniej 50% punktów za wykonane zadanie projektowe oraz sprawozdanie.

*) zostawić tylko realizowane formy zajęć

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		9	9		h
3.	Inne (konsultacje, egzamin)*	2		2	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	33					h
7.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	1,32					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	1,29					ECTS
10.	Sumaryczne godzinowe obciążenie pracą studenta	75					h
11.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	3					

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

LITERATURA

1. Sapięcha K.: Testowanie i diagnostyka systemów cyfrowych, PWN, Warszawa 1987.
2. Sosnowski J.: Testowanie i niezawodność systemów komputerowych, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2005.
3. Tanenbaum Andrew S.: Strukturalna organizacja systemów komputerowych, Helion 2006.