

**KARTA PRZEDMIOTU**

Kod przedmiotu	studia stacjonarne:	M#2-S2-MiBM-EM-112
	studia niestacjonarne:	M#2-N2-MiBM-EM-112
Nazwa przedmiotu	Zaawansowane techniki programowania	
Nazwa przedmiotu w języku angielskim	Advanced Programming Methods	
Obowiązuje od roku akademickiego	2024/2025	

USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	MECHANIKA I BUDOWA MASZYN
Poziom kształcenia	II stopień
Profil studiów	ogólnoakademicki
Forma i tryb prowadzenia studiów	Studia stacjonarne i niestacjonarne
Zakres	eksploatacja maszyn
Jednostka prowadząca przedmiot	Katedra Mechatroniki i Uzbrojenia
Koordynator przedmiotu	dr hab. inż. Izabela Krzysztofik, prof. PŚk
Zatwierdził	dr hab. Jakub Takosoglu, prof. PŚk, Dziekan Wydziału Mechatroniki i Budowy Maszyn

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	Przedmiot specjalnościowy	
Status przedmiotu	obowiązkowy	
Język prowadzenia zajęć	polski	
Usytuowanie w planie studiów - semestr	studia stacjonarne	Semestr I
	studia niestacjonarne	Semestr I
Wymagania wstępne		
Egzamin (TAK/NIE)	NIE	
Liczba punktów ECTS	2	

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



**EFEKTY UCZENIA SIĘ**

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Student ma uporządkowaną pogłębioną wiedzę w zakresie użycia nowoczesnych technik programowania. Rozumie złożoność współczesnych języków programowania, a także możliwość zastosowania w kontekście zaawansowanych programów wykorzystywanych w zagadnieniach związanych z mechaniką i budową maszyn.	MiBM2_W03
Umiejętności	U01	Student potrafi wykorzystywać współczesne techniki programowania do integracji z oprogramowaniem technicznym. Umie przygotowywać własne programy służące do obliczeń inżynierskich oraz automatyzacji zadań związanych z przygotowaniem projektu. Potrafi integrować własne oprogramowanie z powszechnie używanymi programami przez pisanie skryptów wykonawczych.	MiBM2_U01 MiBM2_U02
	U02	Student potrafi planować efekt pracy podczas procesu pisania własnego oprogramowania. Wdraża zaawansowane techniki programowania świadomie. Realizując zadania związane z integracją oprogramowania dostrzega potrzebę dalszego samorozwoju w zakresie wdrażania nowoczesnych technik programowania.	MiBM2_U16
Kompetencje społeczne	K01	Student jest gotów do krytycznej oceny posiadanej wiedzy oraz możliwości poszerzania i uzupełniania wiedzy w zakresie technik programowania z dziedziny mechaniki i budowy maszyn.	MiBM2_K01

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	Podstawy teoretyczne funkcjonowania języków programowania wysokiego poziomu. Omówienie struktur danych. Przedstawienie algorytmiki i zagadnień związanych z budową maszyn mogących być realizowanych przy pomocy napisanych programów. Wprowadzenie do technik programowania umożliwiających komunikację z programami dedykowanymi do projektowania i eksploatacji maszyn. Omówienie Rest API i możliwości automatyzacji zadań takich jak generowanie dokumentacji lub listy materiałów przy pomocy napisanych programów.
laboratorium	Zapoznanie z zaawansowanymi zintegrowanymi edytorami tekstu (IDE) używanymi we współczesnym programowaniu. Instalacja środowiska pracy. Wdrożenie do zaawansowanych metod programowania i zapoznanie z ich paradygmatami. Zapoznanie ze strukturami danych oraz procesem interpretacji oraz kompilacji programów. Zapoznanie z wybranymi wzorcami projektowymi i zaawansowanymi technikami pozwalającymi tworzyć kod programowy czytelny i wielokrotnie używalny. Pisanie programów służących do realizacji zadań inżynierskich polegających na przetwarzaniu danych i ich integracji z powszechnie używanym oprogramowaniem.



**METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ**

Symbol efektu	Metody sprawdzania efektów kształcenia					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			X			
U01			X			X
U02			X			X
K01						X

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład	zaliczenie z oceną	Pozytywne zaliczenie kolokwium końcowego. Uzyskanie co najmniej 50% punktów.
laboratorium	zaliczenie z oceną	Pozytywne zaliczenie kolokwiów cząstkowych oraz praca podczas zajęć. Ocena końcowa jest średnią arytmetyczną

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS												
Lp.	Rodzaj aktywności	Obciążenie studenta										Jednos tka
		studia stacjonarne					studia niestacjonarne					
		W	C	L	P	S	W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	15		15			9		9			h
2.	Inne (konsultacje, egzamin)	2		2			2		2			h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	34					22					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	1,4					0,9					ECTS
5.	Liczba godzin samodzielnej pracy studenta	16					28					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	0,6					1,1					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	25					25					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,0					1,0					ECTS
9.	Sumaryczne obciążenie pracą studenta	50					50					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25-30 godzin obciążenia studenta</i>	2										ECTS





Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



LITERATURA

1. Klabnik Steve, Nichols Carol: The Rust Programming Language. No Starch Press, San Francisco 2018
2. Downey Allen, Elkner Jeffrey, Meyers Chris: Think Python. How to Think Like a Computer Scientist. GreenTea Press Wellesley, Massachusetts 2002 (<http://www.greenteapress.com/thinkpython/>)
3. Pilgrim Martin: Dive into Python 3. (<http://www.diveintopython.net/>)
4. Wentworth Peter, Elkner Jeffrey, Downey Allen, Meyers Chris: How to Think Like a Computer Scientist: Learning with Python 3. Documentation Release 3rd Edition
5. Bjarne Stroustrup: The C++ Programming Language. Pearson Education, Boston 2013

