

# AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

## KARTA OPISU PRZEDMIOTU

<b>Wydział</b>		<b>Informatyki</b>	
<b>Kierunek</b>		<b>Informatyka</b>	
<b>Specjalność</b>			
<b>Semestr</b>	<b>III</b>	<b>Program studiów, dla którego obowiązuje sylabus</b>	<b>2025/2026</b>
<b>Stopień studiów</b>	<b>I</b>		

Nazwa przedmiotu	Podstawy inżynierii oprogramowania			
Kod przedmiotu	PIO			
Łączna liczba godzin	18	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	wykład + ćwiczenia			
Język przedmiotu	polski			
Liczba punktów ECTS	2 (1+1)			

<b>Prowadzący zajęcia</b>	
<b>Forma prowadzonych zajęć</b>	<b>Wykład</b>
<b>Wymiar zajęć</b>	<b>9 h</b>
<b>Stopień (tytuł) naukowy</b>	
<b>Imię</b>	
<b>Nazwisko</b>	

<b>Prowadzący zajęcia</b>	
<b>Forma prowadzonych zajęć</b>	<b>Ćwiczenia</b>
<b>Wymiar zajęć</b>	<b>9 h</b>
<b>Stopień (tytuł) naukowy</b>	
<b>Imię</b>	
<b>Nazwisko</b>	

<b>Wymagania wstępne</b>	Wiedza objęta programem nauczania informatyki, matematyki i fizyki z zakresu szkoły średniej.
<b>Założenia i cele przedmiotu</b>	Celem przedmiotu jest opanowanie przez studentów wiedzy i umiejętności z zakresu posługiwania się wzorcami projektowymi; projektowania oprogramowania zgodnie z metodyką strukturalną lub obiektową; dokonywania przeglądu projektu oprogramowania; wybierania narzędzi wspomagających budowę oprogramowania; doboru modelu procesu wytwarzania oprogramowania do specyfiki przedsięwzięcia; specyfikowania wymagań dotyczących oprogramowania i przeprowadzania ich przeglądu; tworzenia, oceny i realizacji planu testowania; uczestniczenia w inspekcji kodu; zarządzania konfiguracją oprogramowania; opracowywania planu przedsięwzięcia dotyczącego budowy oprogramowania.
<b>Metody dydaktyczne</b>	1. Wykład z elementami dyskusji.

## AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

<b>Wymagania wstępne</b>	Wiedza objęta programem nauczania informatyki, matematyki i fizyki z zakresu szkoły średniej.
	2. Prezentacje multimedialne. 3. Pokazy przykładowych rozwiązań problemów. 4. Rozwiązywanie zadań praktycznych.

Efekty uczenia się (odniesienie do charakterystyk poziomów Polskiej Ramy Kwalifikacji)		Odniesienie do efektów dla kierunku	Odniesienie do efektów uczenia się wg Polskiej Ramy Kwalifikacji
WIEDZA – absolwent zna i rozumie:	W01. Zasady modelowania w języku UML i odpowiednie do tego narzędzia. W02. Sposoby zarządzania projektami IT. W03. Elementy analizy obiektowej oraz zasad projektowania obiektowego. W04. Zasady specyfikowania interfejsów oraz modele planowania, odbioru i utrzymania oprogramowania w aspekcie systemów informacyjnych. W05. Metody optymalizacji modeli obiektowych. W06. Techniki testowania oprogramowania.	K_W06 K_W20 K_W21 K_W22	P6S_WG P6S_WG_INŻ
UMIEJĘTNOŚCI – absolwent potrafi:	U01. Zarządzać zespołem programistycznym. U02. Zorganizować projekt informatyczny oraz odpowiedni zespół. U03. Zaprojektować diagramy UML niezbędne do realizacji funkcji systemowych. U04. Specyfikować interfejsy oraz dobrać odpowiednie modele planowania, odbioru i utrzymania oprogramowania w aspekcie systemów informacyjnych. U05. Przeprowadzić procesy testowe aplikacji.	K_U14 K_U22	P6S_UW P6S_UW_INŻ
KOMPETENCJE SPOŁECZNE – absolwent jest gotów do	K01. Aktywnego i systematycznego doskonalenia się i aktywnego uczestnictwa w pracach grupy.	K_K03	P6S_UU

Treści programowe		
Lp.	Tematyka zajęć	Liczba godzin
<b>Forma zajęć – wykład</b>		
1	Wprowadzenie do inżynierii oprogramowania.	1
2	Organizacja projektu i komunikacja.	1
3	Analiza obiektowa, zagadnienia projektowania obiektowego.	1
4	Podsystemy i klasy, diagramy UML.	1
5	Specyfikacja interfejsów.	1
6	Optymalizacja modeli obiektowych.	1
7.	Wprowadzenie do testowania.	1

## AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

8	Wprowadzenie do zarządzania projektem informatycznym.	1
9	Modele planowania, odbioru i utrzymania oprogramowania w aspekcie systemów informacyjnych.	1

Treści programowe		
Lp.	Tematyka zajęć	Liczba godzin
<b>Forma zajęć – ćwiczenia</b>		
1	Przetarg na wykonanie oprogramowania – analiza dokumentacji.	1
2	Funkcjonalności – historyjki użytkownika, diagram przypadków użycia.	1
3	Funkcjonalności – scenariusze przypadków użycia.	1
4	UML – diagramy statyczne.	1
5	Diagramy UML przedstawiające działanie oprogramowania (diagramy dynamiczne).	1
6	Testowania funkcjonalne.	1
7	Testowanie spełnienia wymagań niefunkcjonalnych.	1
8	Specyfikacja interfejsów oprogramowania.	1
9	Scrum – tworzenie Backlogu Produktu i Backlogu Sprintu. Zaliczenie.	1

<b>Forma i warunki zaliczenia przedmiotu</b>		
<b>Metody weryfikacji efektów uczenia się</b>		<b>Nr efektu uczenia się z sylabusu</b>
	Aktywność na zajęciach – rozwiązywanie zestawów zadań	U01–U05
	Ocena projektu/prezentacji.	W01–W06

<b>Literatura podstawowa</b>	<ol style="list-style-type: none"> <li>1. J. Roszkowski, <i>Analiza i projektowanie strukturalne</i>, Helion, Gliwice, 2004.</li> <li>2. I. Sommerville, <i>Inżynieria oprogramowania</i>, WNT, Warszawa 2022.</li> </ol>
<b>Literatura uzupełniająca</b>	<ol style="list-style-type: none"> <li>1. M. Fowler., <i>UML w kropelce, wersja 2.0</i>, LTP, 2005.</li> </ol>

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	18
Przygotowanie się do zajęć	9
Studiowanie literatury	9
Udział w konsultacjach	2
Przygotowanie projektu / eseju / prezentacji itp.	10

**AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH**

<b>Nakład pracy studenta</b>	
Przygotowanie się do egzaminu / zaliczenia	9
Inne	-
<b>ŁĄCZNY nakład pracy studenta w godz.</b>	<b>57</b>
<b>Liczba punktów ECTS</b>	<b>2</b>