

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

KARTA OPISU PRZEDMIOTU

Wydział		Informatyki	
Kierunek		Informatyka	
Specjalność		Programowanie	
Semestr	VI	Program studiów, dla którego obowiązuje sylabus	2025/2026
Stopień studiów	I		

Nazwa przedmiotu	Zaawansowane programowanie w chmurze			
Kod przedmiotu	ZPWC			
Łączna liczba godzin	18	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	laboratorium			
Język przedmiotu	polski			
Liczba punktów ECTS	2			

Prowadzący zajęcia	
Forma prowadzonych zajęć	Laboratorium
Wymiar zajęć	18 h
Stopień (tytuł) naukowy	
Imię	
Nazwisko	

Wymagania wstępne	Podstawowa znajomość języków programowania, takich jak Java lub C++. Umiejętność obsługi komputera i środowisk programistycznych. Znajomość podstawowych algorytmów i struktur danych.
Założenia i cele przedmiotu	Przedmiot koncentruje się na tworzeniu i wdrażaniu zaawansowanych rozwiązań w chmurze, ze szczególnym uwzględnieniem podejścia serverless, mikrousług, CI/CD oraz bezpieczeństwa. Studenci poznają różne platformy chmurowe (AWS, Azure, GCP), nauczą się modelować architekturę w chmurze, automatyzować wdrożenia i stosować zasady DevSecOps.
Metody dydaktyczne	<ol style="list-style-type: none"> 1. Prezentacje multimedialne. 2. Pokazy przykładowych rozwiązań problemów. 3. 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. Podstawowe koncepcje i usługi dostępne w popularnych środowiskach chmurowych (AWS, Azure, GCP) oraz różnice między IaaS, PaaS, a SaaS.	K_W06 K_W16 K_W19 K_W20	P6S_WG P6S_WG_INŻ

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

	<p>W02. Architektury mikrousług i modele serverless oraz rolę funkcji jako usługi (FaaS) w skalowalnych aplikacjach chmurowych.</p> <p>W03. Metody zabezpieczania i monitorowania aplikacji wdrożonych w chmurze, w tym stosowanie polityk bezpieczeństwa i analizy logów.</p> <p>W04. Zastosowanie narzędzi do orkiestracji kontenerów i zarządzania infrastrukturą w chmurze (np. Kubernetes, Docker w środowisku chmurowym).</p> <p>W05. Zasady automatyzacji procesów wdrożeń (CI/CD) oraz praktyki DevOps w chmurze.</p>	K_W23	
UMIEJĘTNOŚCI – absolwent potrafi:	<p>U01. Wdrażać i konfigurować aplikacje w chmurze, korzystając z usług chmurowych (Compute, Storage, Functions) dostarczanych przez AWS, Azure lub GCP.</p> <p>U02. Korzystać z kontenerów i orkiestratorów w celu skalowania i zarządzania aplikacjami chmurowymi.</p> <p>U03. Przeprowadzić testy wydajnościowe i bezpieczeństwa aplikacji działających w chmurze oraz zastosować mechanizmy automatyzacji tych testów.</p> <p>U04. Porównać różne rozwiązania chmurowe i wybrać optymalne pod kątem kosztów, wydajności i bezpieczeństwa.</p> <p>U05. Zaprojektować architekturę aplikacji chmurowej wykorzystującej mikrousługi i serverless, integrując ją z pipeline CI/CD.</p>	K_U01 K_U02 K_U03 K_U04 K_U08 K_U09 K_U24 K_U25	P6S_UW P6S_UW_INŻ P6S_UO P6S_KK P6S_UK
KOMPETENCJE SPOŁECZNE – absolwent jest gotów do	<p>K01. Pracy w zespole, przyjmując w nim różne role.</p> <p>K02. Krytycznej oceny możliwości urządzeń oprogramowania i systemów dostępnych na rynku IT.</p> <p>K03. Ciągłego samokształcenia się w celu dostosowywania się do dynamicznie zmieniających się technologii.</p>	K_K04 K_K05 K_K06	P6S_UO P6S_KR P6S_KK

Lp.	Tematyka zajęć	Liczba godzin
Forma zajęć – laboratorium		
1	Usługi chmurowe (AWS, Azure, GCP), modelowanie architektury w chmurze.	3
2	Serverless computing. Funkcje jako usługa (FaaS), tworzenie i zarządzanie funkcjami	4

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

	serverless.	
3	Programowanie mikrousług. Tworzenie i zarządzanie mikrousługami w środowiskach chmurowych.	5
4	Automatyzacja i CI/CD. Narzędzia do automatyzacji wdrożeń, Continuous Integration/Continuous Deployment.	3
5	Bezpieczeństwo aplikacji w chmurze. Zasady zabezpieczania aplikacji chmurowych, monitorowanie i audyt. Zaliczenie.	3

Forma i warunki zaliczenia przedmiotu	Wykonanie projektów. Częstkowe prezentacje, zdawanie raportów, obrona projektów.	
Metody weryfikacji efektów uczenia się		Nr efektu uczenia się z sylabusu
	Ocena projektów i częstkowych prezentacji.	W01-W05, U01-U05, K01-K03

Literatura podstawowa	<ol style="list-style-type: none"> 1. C. Dotson, <i>Bezpieczeństwo w chmurze</i>, Helion, Gliwice 2020. 2. M. Kief, <i>Infrastruktura jako kod. Dynamiczne systemy w epoce chmury</i>, APN Promise 2021. 3. R. C. Martin, <i>Czysty kod. Podręcznik dobrego programisty</i>, Helion, Gliwice 2010. 4. J. Roszkowski, <i>Analiza i projektowanie strukturalne</i>, Helion, Gliwice, 2004. 5. N. Wirth, <i>Algorytmy + struktury danych = programy</i>, WNT, Warszawa 2002
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. M. Krief, <i>DevOps w praktyce. Wdrażanie narzędzi Terraform, Azure DevOps, Kubernetes i Jenkins</i>, Helion, Gliwice 2023. 2. S. Prata, <i>Język C++. Szkoła programowania</i>. Wydanie VI, Helion, Gliwice 2019. 3. B. Eckel, <i>Thinking in Java. Edycja polska</i>, Helion, Gliwice 2006.

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.	22
Przygotowanie się do egzaminu / zaliczenia	-
Inne	-
ŁĄCZNY nakład pracy studenta w godz.	60
Liczba punktów ECTS	2