

KARTA KURSU (realizowanego w specjalności)**Inżynieria oprogramowania**

(nazwa specjalności)

Nazwa	Programowanie sieciowe
Nazwa w j. ang.	

Koordinator	mgr Patryk Mazurek	Zespół dydaktyczny
		Zespół dydaktyczny: mgr Patryk Mazurek mgr Michał Frontczak
Punktacja ECTS*	st. stacjonarne: 4 st. niestacjonarne: 4	

Opis kursu (cele kształcenia)

Kurs wprowadza studentów w zagadnienia programowania sieciowego w środowiskach rozproszonych. Obejmuje tworzenie aplikacji komunikujących się przez sieć z wykorzystaniem gniazd TCP/UDP, podstawy konfiguracji usług pocztowych oraz projektowanie systemów opartych na architekturze mikro serwisów. Zajęcia umożliwiają praktyczne poznanie technologii i narzędzi wykorzystywanych do budowy nowoczesnych, skalowalnych aplikacji sieciowych, niezależnie od wybranego języka programowania.

Warunki wstępne

Wiedza	Student zna podstawy projektowania i programowania w dowolnym języku programowania oraz zna podstawy protokołów sieciowych TCP/UDP
Umiejętności	Student posiada podstawowe umiejętności programowania obiektowego, potrafi analitycznie analizować napisany kod i rozwiązywać problemy. Student posiada podstawową wiedzę z zakresu sieci komputerowych.
Kursy	Programowanie Obiektowe, Programowanie w języku Java

Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Wiedza	Po zakończeniu kursu student:	
	W01: posiada zaawansowaną wiedzę z języków i technik programowania oraz środowisk ich wykorzystania	K_W06
	W02: posiada zaawansowaną wiedzę z zakresu systemów operacyjnych, sieci komputerowych i urządzeń sieciowych	K_W08
	W03: zna systemy bazodanowe, rozumie ich rolę i zasady Funkcjonowania	K_W09
	W04: w zaawansowanym stopniu zna usługi i technologie Internetowe	K_W10
	W05: Posiada zaawansowaną wiedzę na temat programowania obiektowego, systemowego i sieciowego.	S1_W02

Umiejętności	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
	Po zakończeniu kursu student:	
	U01: dobiera aplikacje, technologie, standardy i dostępne metody w celu rozwiązywania problemów informatycznych	K_U04
	U02: posiada umiejętność programowania oraz pracy w zespołach programistycznych	K_U05
	U03: Posiada umiejętności stosowania narzędzi programistycznych.	S1_U01
	U04: Potrafi tworzyć programy z wykorzystaniem różnych technologii (REST, Web Services, RPC) oraz modeluje przepływ danych w systemach informatycznych	S1_U04

Kompetencje społeczne	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
	Po zakończeniu kursu student:	
	K01: potrafi ocenić poziom swojej wiedzy i umiejętności (predyspozycje) i w związku z tym dostrzega konieczność kształcenia ustawicznego (podnoszenia kwalifikacji zawodowych).	K_K01
	K02: dostrzega potrzebę popularyzacji wiedzy z zakresu nowych technologii oraz dzielenia się wiedzą informatyczną w sposób otwarty i zrozumiały dla innych..	K_K02
	K03: Potrafi formułować opinie na temat zagadnień związanych z inżynierią oprogramowania.	S1_K02
	K04: Potrafi pracować zespołowo i rozumie konieczność systematycznej pracy nad projektami, które są realizowane wieloetapowo.	S1_K03

Studia stacjonarne

Organizacja							
Forma zajęć	Wykład (W)	Ćwiczenia w grupach					
		A	K	L	S	P	E
Liczba godzin				30			

Studia niestacjonarne

Organizacja							
Forma zajęć	Wykład (W)	Ćwiczenia w grupach					
		A	K	L	S	P	E
Liczba godzin				20			

Opis metod prowadzenia zajęć

Kurs ma formę zajęć laboratoryjnych, podczas których studenci poznają teoretyczne podstawy programowania sieciowego oraz zdobywają praktyczne umiejętności tworzenia aplikacji działających w środowisku sieciowym. W ramach laboratoriów realizowane są ćwiczenia obejmujące m.in. pracę z gniazdami TCP/UDP, konfigurację usług pocztowych oraz implementację mikro serwisów.

Formy sprawdzania efektów uczenia się

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Zadania problemowe
W01					X			X					
W02					X			X					
W03					X			X					
W04					X			X					
W05					X			X					
U01					X	X							
U02					X	X							
U03					X	X							
U04					X	X							
K01					X								
K02					X								
K03					X								
K04					X								

Kryteria oceny	Ocena dostateczna
	- Potrafi zaimplementować prostą aplikację klient – serwer w oparciu o klasy Socket i ServerSocket - Potrafi zaimplementować aplikację wykorzystującą dane Rest API, SOAP
	Ocena dobra
	- Potrafi zaimplementować i zasymulować działanie aplikacji opartej o architekturę klient – serwer z wykorzystaniem RestAPI, SOAP - Potrafi wykonać zdalny dostęp do bazy danych i związane z tym zadania - Pozostałe wymogi jak dla oceny dostatecznej
	Ocena bardzo dobra
	- Potrafi zaimplementować i zasymulować wybrany model Rest API - Pozostałe wymogi jak dla oceny dobrej i dostatecznej

Uwagi		
-------	--	--

Treści merytoryczne (wykaz tematów)

1. Programowanie wielowątkowe w języku Java
2. Klasy z Java SE API pozwalające na zapis i odczyt plików
3. Operacje na strumieniach
4. Omówienie wybranych protokołów sieciowych (np. Echo, HTTP, SMTP, POP3, IMAP)
5. Tworzenie programów w architekturze klient serwer w oparciu o gniazda TCP/UDP.
6. Zdalny dostęp do baz danych za pośrednictwem JDBC
7. Tworzenie programów oparciu o RestAPI, SOAP
8. Omówienie wybranych modeli mikro usług (np. RabbitMQ, MQTT, gRPC)

Wykaz literatury podstawowej

1. Cay S. Horstmann, Gary Cornell, Java Podstawy, wyd. 9, Helion 2014
2. Cay S. Horstmann, Gary Cornell, Java Techniki Zaawansowane, wyd. 9, Helion 2014
3. Wojciech Romowicz, Java Server Pages : oraz inne komponenty JavaPlatform, Helion 2001
4. Harold, Elliotte Rusty. Java: programowanie sieciowe, Wydawnictwo RM 2001

Wykaz literatury uzupełniającej

1. Eckel, Bruce, Thinking in Java Helion 2006
2. Ronnie Mita, Irakli Nadareishvili, Mikrouslugi Budowa i działanie – Przewodnik po budowaniu architektury mikrouslug. APN Promise 2021

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) – studia stacjonarne

Liczba godzin w kontakcie z prowadzącymi	Wykład	
	Konwersatorium (ćwiczenia, laboratorium itd.)	30
	Pozostałe godziny kontaktu studenta z prowadzącym	15
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	10
	Realizacja zadań domowych (problemowych) po zapoznaniu się z niezbędną literaturą przedmiotu	20
	Przygotowanie projektu lub prezentacji na podany temat (praca indywidualna lub w grupie)	25
	Przygotowanie do egzaminu/zaliczenia	
Ogółem bilans czasu pracy		100
Liczba punktów ECTS w zależności od przyjętego przelicznika		4

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) – studia niestacjonarne

Liczba godzin w kontakcie z prowadzącymi	Wykład	
	Konwersatorium (ćwiczenia, laboratorium itd.)	20
	Pozostałe godziny kontaktu studenta z prowadzącym	20
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	25
	Realizacja zadań domowych (problemowych) po zapoznaniu się z niezbędną literaturą przedmiotu	
	Przygotowanie projektu lub prezentacji na podany temat (praca indywidualna lub w grupie)	35
	Przygotowanie do egzaminu/zaliczenia	
Ogółem bilans czasu pracy		100
Liczba punktów ECTS w zależności od przyjętego przelicznika		4