

# KARTA KURSU (realizowanego w specjalności)

## Administracja Systemami Informatycznymi

Nazwa	<b>Programowanie Sieciowe</b>
Nazwa w j. ang.	Network Programming

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

### Opis kursu (cele kształcenia)

Celem kursu jest zapoznanie studentów z zagadnieniami tworzenia programów działających w rozproszonym środowisku sieciowym przy użyciu języka Java. W ramach kursu przedstawione zostaną również aspekty programowania wielowątkowego w języku Java. Kurs jest realizowany w języku polskim.

### Wymogi wstępne

Wiedza	Student zna podstawowe zagadnienia z algorytmiki (struktury danych i proste algorytmy) oraz składnię języka C++.
Umiejętności	Potrafi zapisywać podstawowe algorytmy i definiować struktury danych za pomocą języka C++.
Kursy	<u>Wymagane zaliczenie kursu:</u> <ul style="list-style-type: none"> <li>Podstawy programowania,</li> <li>Wstęp do programowania w języku Java</li> </ul>

### Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów dla specjalności (określonych w karcie programu studiów dla specjalności)
Wiedza	Po zakończeniu kursu	-
	W01: ma wiedzę na temat zagadnień związanych z programowaniem wielowątkowym w języku Java,	S1_W01
	W02: zna klasy wchodzące w skład Java SE API wspomagające tworzenie programów sieciowych w języku Java	S1_W02
	W03: orientuje się w wybranych protokołach sieciowych i sposobach ich wykorzystania w swoich programach,	S1_W02
	W04: zna podstawy wykorzystania gniazd TCP i UDP do tworzenia programów klienckich i serwerowych,	S1_W02
	W05: zna architekturę warstwy dostępu do relacyjnych baz danych JDBC.	S1_W03

Umiejętności	Efekt uczenia się dla kursu	Odniesienie do efektów dla specjalności (określonych w karcie programu studiów dla specjalności)
	Po zakończeniu kursu	
	U01: potrafi projektować i implementować programy wielowątkowe w języku Java,	S1_U06
	U02: tworzy z użyciem języka Java programy bazujące na modelu klient – serwer,	S1_U06, S1_U08
	U03: wykorzystuje gniazda TCP i UDP w swoich programach,	S1_U06
	U04: potrafi korzystać ze specyfikacji RFC w celu implementacji wybranych protokołów sieciowych.	S1_U06

Kompetencje społeczne	Efekt uczenia się dla kursu	Odniesienie do efektów dla specjalności (określonych w karcie programu studiów dla specjalności)
	Po zakończeniu kursu	
	K01: korzysta z różnych źródeł informacji w celu realizacji postawionych mu zadań programistycznych,	S1_K01, S1_K03
	K02: potrafi współpracować w grupie w celu realizacji postawionych mu zadań.	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 składa się z laboratoriów na których studenci tworzą programy korzystając z wybranych aspektów programowania sieciowego. Dodatkowo w ramach laboratoriów demonstrowane i omawiane są projekty realizowane przez studentów samodzielnie lub w grupach.

## 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	Inne
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								

Kryteria oceny	Ocena dostateczna
	<ul style="list-style-type: none"> <li>Potrafi zaimplementować prostą aplikację klient – serwer w oparciu o klasy Socket i ServerSocket</li> <li>Potrafi zaimplementować aplikację pobierającą dane przez Rest API, SOAP</li> </ul>
	Ocena dobra
	<ul style="list-style-type: none"> <li>Potrafi zaimplementować i zasymulować działanie aplikacji opartej o architekturę klient – serwer w oparciu o RestAPI, SOAP</li> <li>Potrafi wykonać zdalny dostęp do bazy danych i wykonać z tym związane zadania</li> <li>Pozostałe wymogi jak dla oceny dostatecznej</li> </ul>
	Ocena bardzo dobra
	<ul style="list-style-type: none"> <li>Potrafi zaimplementować i zasymulować wybrany model Rest API</li> <li>Pozostałe wymogi jak dla oceny dobrej i dostatecznej</li> </ul>

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)
5. Tworzenie programów klienckich w oparciu o gniazda TCP (klasa Socket)
6. Tworzenie wielowątkowych serwerów na bazie gniazd TCP (klasa ServerSocket)
7. Zdalny dostęp do baz danych za pośrednictwem JDBC
8. Programowanie w architekturze klient-serwer w oparciu o gniazda UDP
9. Programowanie w architekturze klient-serwer w oparciu o RestAPI, SOAP
10. Omówienie wybranych modeli RestAPI

## 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, Hellion 2001
4. Harold, Elliott Rusty. Java : programowanie sieciowe, Wydawnictwo RM 2001

## Wykaz literatury uzupełniającej

2. Eckel, Bruce, Thinking in Java Hellion 2006

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

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

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

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