

**AGH**AGH UNIVERSITY OF SCIENCE  
AND TECHNOLOGY

Nazwa modułu zajęć:	Informatyka w mechatronice				
Rok akademicki:	2019/2020	Kod:	RIME-2-102-SI-s	Punkty ECTS:	6
Wydział:	Inżynierii Mechanicznej i Robotyki				
Kierunek:	Inżynieria Mechatroniczna	Specjalność:	Systemy inteligentne		
Poziom studiów:	Studia II stopnia	Forma studiów:	Stacjonarne		
Język wykładowy:	Polski	Profil:	Ogólnoakademicki (A)	Semestr:	1
Strona www:	—				
Prowadzący moduł:	dr inż. Miękina Lucjan (miekina@agh.edu.pl)				

### Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

Celem przedmiotu jest zapoznanie studentów z wybranymi metodami i narzędziami informatycznymi, związanymi z językami Java i Scala w zastosowaniu do tworzenia oprogramowania systemów mechatronicznych. Oprócz typowo stosowanego podejścia obiektowego do programowania, wprowadzane są elementy programowania funkcyjnego. Zastosowania obejmują programowanie równoległe, komunikację sieciową z użyciem UDP i TCP, RMI, usługi sieciowe SOAP i REST, podstawy szkieletu aplikacji Spring Framework.

### Opis efektów uczenia się dla modułu zajęć

Kod MEU	Student, który zaliczył moduł zajęć zna i rozumie/potrafi/jest gotów do	Powiązania z KEU	Sposób weryfikacji i oceny efektów uczenia się osiągniętych przez studenta w ramach poszczególnych form zajęć i dla całego modułu zajęć
Wiedza: zna i rozumie			
M_W001	zna i rozumie zasady programowania obiektowego i funkcyjnego w języku Scala	IME2A_W02	Kolokwium
M_W002	zna i rozumie działanie, podstawy wykorzystania i sposoby konfiguracji SBT; zna i rozumie strukturę aplikacji webowej zbudowanej z użyciem szkieletu Play Framework; zna i rozumie sposób użycia szkieletu Slick.	IME2A_W02	Kolokwium

M_W003	zna i rozumie zasady programowania obiektowego w języku Java (klasy, obiekty, dziedziczenie, interfejsy, programowanie rodzajowe i elementy programowania równoległego); zna i rozumie elementy programowania funkcyjnego w Javie; zna i rozumie tworzenie aplikacji z zastosowaniem Spring Framework	IME2A_W02	Wynik testu zaliczeniowego, Wykonanie ćwiczeń laboratoryjnych, Wykonanie projektu, Kolokwium
M_W004	zna i rozumie metody programowania sieciowego i rozproszonego w Javie: UDP, TCP, RMI, usługi sieciowe SOAP i RESTful.	IME2A_W02	Wynik testu zaliczeniowego, Wykonanie ćwiczeń laboratoryjnych, Wykonanie projektu, Kolokwium
Umiejętności: potrafi			
M_U001	potrafi zbudować program w języku Java z wykorzystaniem klas, obiektów, interfejsów, typów rodzajowych i wątków, potrafi zastosować wybrane elementy programowania funkcyjnego w języku Java	IME2A_U09, IME2A_U12	Wykonanie ćwiczeń laboratoryjnych, Wykonanie projektu
M_U002	potrafi zbudować programy w języku Java wykorzystujące protokoły UDP i TCP, a także techniki programowania rozproszonego oparte o RMI, usługi sieciowe SOAP i RESTful, potrafi wykorzystać szkielet Spring Framework do budowy aplikacji	IME2A_U12	Wykonanie ćwiczeń laboratoryjnych, Wykonanie projektu
M_U003	potrafi zbudować program w języku Scala z wykorzystaniem SBT do zarządzania projektem, szkieletem Play do budowy aplikacji webowej i szkieletem Slick do współpracy z systemem bazodanowym (relacyjnym)	IME2A_U12	

### Liczba godzin zajęć w ramach poszczególnych form zajęć

Suma	Forma zajęć dydaktycznych										
	Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Prace kontrolne i przejściowe	Lektorat
68	28	0	26	14	0	0	0	0	0	0	0

### Matryca kierunkowych efektów uczenia się w odniesieniu do form zajęć i sposobu zaliczenia, które pozwalają na ich uzyskanie

Kod MEU	Student, który zaliczył moduł zajęć zna i rozumie/potrafi/jest gotów do	Forma zajęć dydaktycznych
---------	---	---------------------------

		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Prace kontrolne i przejściowe	Lektorat
Wiedza: zna i rozumie												
M_W001	zna i rozumie zasady programowania obiektowego i funkcyjnego w języku Scala	+	-	+	+	-	-	-	-	-	-	-
M_W002	zna i rozumie działanie, podstawy wykorzystania i sposoby konfiguracji SBT; zna i rozumie strukturę aplikacji webowej zbudowanej z użyciem szkieletu Play Framework; zna i rozumie sposób użycia szkieletu Slick.	+	-	+	+	-	-	-	-	-	-	-
M_W003	zna i rozumie zasady programowania obiektowego w języku Java (klasy, obiekty, dziedziczenie, interfejsy, programowanie rodzajowe i elementy programowania równoległego); zna i rozumie elementy programowania funkcyjnego w Javie; zna i rozumie tworzenie aplikacji z zastosowaniem Spring Framework	+	-	-	-	-	-	-	-	-	-	-
M_W004	zna i rozumie metody programowania sieciowego i rozproszonego w Javie: UDP, TCP, RMI, usługi sieciowe SOAP i RESTful.	+	-	-	-	-	-	-	-	-	-	-
Umiejętności: potrafi												
M_U001	potrafi zbudować program w języku Java z wykorzystaniem klas, obiektów, interfejsów, typów rodzajowych i wątków, potrafi zastosować wybrane elementy programowania funkcyjnego w języku Java	-	-	-	-	-	-	-	-	-	-	-
M_U002	potrafi zbudować programy w języku Java wykorzystujące protokoły UDP i TCP, a także techniki programowania rozproszonego oparte o RMI, usługi sieciowe SOAP i RESTful, potrafi wykorzystać szkielet Spring Framework do budowy aplikacji	-	-	+	+	-	-	-	-	-	-	-
M_U003	potrafi zbudować program w języku Scala z wykorzystaniem SBT do zarządzania projektem, szkieletem Play do budowy aplikacji webowej i szkieletem Slick do współpracy z systemem bazodanowym (relacyjnym)	-	-	+	+	-	-	-	-	-	-	-

## Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w zajęciach dydaktycznych/praktyka	68 godz
Przygotowanie do zajęć	28 godz
przygotowanie projektu, prezentacji, pracy pisemnej, sprawozdania	32 godz
Samodzielne studiowanie tematyki zajęć	28 godz
Egzamin lub kolokwium zaliczeniowe	2 godz
Sumaryczne obciążenie pracą studenta	158 godz
Punkty ECTS za moduł	6 ECTS

## Pozostałe informacje

### Szczegółowe treści kształcenia w ramach poszczególnych form zajęć (szczegółowy program wykładów i pozostałych zajęć)

#### Wykład

##### Zastosowania języka programowania Java

- geneza i cechy języka, zastosowania i główne konstrukcje
- programowanie równoległe
- programowanie sieciowe z użyciem protokołów UDP i TCP
- programowanie rozproszone z użyciem RMI i usług sieciowych SOAP i REST
- programowanie funkcyjne.

Wymienione zagadnienia są ilustrowane przykładami rozwiązań w architekturze klient-serwer.

##### Zastosowania języka programowania Scala

- wprowadzenie, instalacja i konfiguracja systemu
- Scala – wprowadzenie do języka
- SBT – narzędzie zarządzania projektami
- Play – struktura, szablony, podstawowe konstrukcje składniowe, formularze, walidacja
- Slick – praca z systemem bazodanowym (relacyjnym)
- Play – zadania zaawansowane – ajax, transfer plików, web streams

#### Ćwiczenia laboratoryjne

##### Zastosowania języka programowania Java

W trakcie ćwiczeń laboratoryjnych studenci:

- zapoznają się ze środowiskiem Eclipse i programowaniem w języku Java
- wspólnie dyskutują i rozwijają programy ilustrujące zastosowanie protokołów UDP i TCP w komunikacji sieciowej, technologii Java RMI i usług sieciowych w programowaniu systemów rozproszonych
- budują aplikacje z zastosowaniem Spring Framework.

##### Zastosowania języka programowania Scala

W drugiej części laboratoriów studenci pracują nad aplikacją webową, wykorzystując:

- język Scala
- SBT – narzędzie zarządzania projektami
- Play – szkielet aplikacji webowej
- Slick – szkielet realizujący współpracę z systemem bazodanowym (relacyjnym)

### **Ćwiczenia projektowe**

W trakcie ćwiczeń projektowych studenci realizują projekt z zakresu obejmującego pierwszą lub drugą część wykładu:

- projekt i implementacja aplikacji w konfiguracji klient-serwer
- projekt i implementacja aplikacji webowej o zadanej funkcjonalności

### **Metody i techniki kształcenia:**

Wykład: Treści prezentowane na wykładzie są przekazywane w formie prezentacji multimedialnej w połączeniu z klasycznym wykładem tablicowym wzbogaconymi o pokazy odnoszące się do prezentowanych zagadnień.

Ćwiczenia laboratoryjne: W trakcie zajęć laboratoryjnych studenci samodzielnie rozwiązują zadany problem praktyczny, dobierając odpowiednie narzędzia. Prowadzący stymuluje grupę do refleksji nad problemem, tak by otrzymane wyniki miały wysoką wartość merytoryczną.

Ćwiczenia projektowe: Studenci wykonują zadany projekt samodzielnie, bez większej ingerencji prowadzącego. Ma to wykształcić poczucie odpowiedzialności za pracę w grupie oraz odpowiedzialności za podejmowane decyzje.

### **Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu:**

Warunkiem zaliczenia ćwiczeń laboratoryjnych jest zamieszczenie w repozytorium poprawnego kodu będącego implementacją zadania określonego w instrukcji do ćwiczeń.

Warunkiem zaliczenia ćwiczeń projektowych jest zamieszczenie w repozytorium poprawnego kodu będącego implementacją zadania określonego w temacie projektu i uzasadnienie przyjętych rozwiązań.

### **Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa:**

Wykład:

– Obecność obowiązkowa: Nie

– Zasady udziału w zajęciach: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego.

Ćwiczenia laboratoryjne:

– Obecność obowiązkowa: Tak

– Zasady udziału w zajęciach: Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane kolokwium w formie ustnej lub pisemnej. Zaliczenie zajęć odbywa się na podstawie zaprezentowania rozwiązania postawionego problemu. Zaliczenie modułu jest możliwe po zaliczeniu wszystkich zajęć laboratoryjnych.

Ćwiczenia projektowe:

– Obecność obowiązkowa: Tak

– Zasady udziału w zajęciach: Studenci wykonują prace praktyczne mające na celu uzyskanie kompetencji zakładanych przez sylabus. Ocenie podlega sposób wykonania projektu oraz efekt końcowy.

### **Sposób obliczania oceny końcowej**

Średnia ważona z ocen:

- ćwiczeń laboratoryjnych dot. zastosowań języka Java – 25%
- ćwiczeń laboratoryjnych dot. zastosowań języka Scala – 25%
- ćwiczeń projektowych – 50%

### **Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach:**

samodzielna realizacja zaległych tematów i przedstawienie wyników do oceny.

### **Wymagania wstępne i dodatkowe, z uwzględnieniem sekwencyjności modułów**

- znajomość obiektowego podejścia do programowania
- umiejętność programowania obiektowego w języku Java
- znajomość sieci komputerowych i baz danych

### **Zalecana literatura i pomoce naukowe**

- Wykład
- Core Java Volume I - Fundamentals, Cay S. Horstmann, 10 Ed.
- Core Java Volume II - Advanced Features, Cay S. Horstmann, 10 Ed.
- Functional Programming in Java, Venkat Subramaniam, The Pragmatic Bookshelf, 2014

### **Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu**

Inżynieria oprogramowania. Skrypt AGH SU 1707, Wydawnictwa AGH, Kraków 2009

### **Informacje dodatkowe**

Brak