

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

KARTA OPISU PRZEDMIOTU

Wydział		Informatyki	
Kierunek		Informatyka	
Specjalność		Java EE	
Semestr	VII	Program studiów, dla którego obowiązuje syllabus	2024/2025
Stopień studiów	I		

Nazwa przedmiotu	Programowanie aplikacji webowych			
Kod przedmiotu	PAW			
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 języka JAVA oraz programowania obiektowego.
Założenia i cele przedmiotu	Celem przedmiotu jest wprowadzenie studentów w proces tworzenia aplikacji webowych w środowisku Java Enterprise Edition, obejmujących integrację warstwy frontendowej i backendowej, zarządzanie stanem aplikacji, testowanie oraz optymalizację pod kątem wydajności i szybkości ładowania. Studenci nauczą się stosować architektoniczne wzorce w kontekście serwerowych frameworków JEE, korzystać z technologii takich jak Servlety, JavaServer Faces (JSF), JAX-RS, JPA oraz wdrażać testy jednostkowe i integracyjne z użyciem narzędzi dedykowanych dla ekosystemu Java. Ponadto studenci poznają techniki optymalizacji aplikacji tworzonych w ramach standardów Enterprise Edition oraz praktyki CI/CD.
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	W01.Architektoniczne wzorce projektowe (MVC, MVP, MVVM) oraz ich implementację przy	K_W06 K_W07	P6S_WG P6S_WG_INŻ

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

rozumie:	<p>użyciu technologii Java EE (np. JSF, JAX-RS) i ich wpływ na strukturę serwerowych aplikacji webowych.</p> <p>W02. Standardy i technologie Java EE (Servlety, JSF, JPA, JAX-RS), sposoby integracji warstwy frontendowej z backendem i bazą danych, zapewniające spójność danych oraz interfejsu użytkownika.</p> <p>W03. Metody zarządzania stanem aplikacji przy użyciu komponentów Java EE (CDI, stanowe i bezstanowe ejb-y) oraz frontendowych rozwiązań (Redux, Vuex, Context API) i ich wpływ na skalowalność oraz utrzymanie kodu.</p> <p>W04. Zasady testowania aplikacji webowych w kontekście JEE (testy jednostkowe z JUnit, integracyjne z Arquillian, testy end-to-end), narzędzia do ich realizacji oraz sposoby automatyzacji procesu testowania.</p> <p>W05. Techniki optymalizacji ładowania i renderowania stron oraz mechanizmy cache'owania w JEE (np. cachowanie zapytań JPA, integracja z CDN), minifikację kodu, lazy loading, a także narzędzia do profilowania aplikacji Java w środowisku serwerowym.</p>	<p>K_W10</p> <p>K_W20</p>	
UMIEJĘTNOŚCI – absolwent potrafi:	<p>U01. Zastosować wybrany wzorzec architektoniczny (np. MVC) w ramach środowiska Java EE (JSF, JAX-RS) w celu strukturyzacji aplikacji webowej.</p> <p>U02. Zintegrować frontend z backendem wykorzystując standardy i API Java EE (Servlety, JAX-RS) oraz zapewnić płynny przepływ danych między serwerem, warstwą persistence (JPA) a interfejsem użytkownika.</p> <p>U03. Wykorzystać narzędzia do zarządzania stanem aplikacji, takie jak CDI (Context and Dependency Injection) czy techniki frontendowe (Redux, Vuex), optymalizując przepływ danych i logikę biznesową w JEE.</p> <p>U04. Zaplanować i wykonać testy aplikacji webowych napisanych w JEE, automatyzując proces testowania (JUnit, Arquillian) w celu zwiększenia jakości i niezawodności oprogramowania.</p> <p>U05. Wdrożyć techniki optymalizacji wydajności (m.in. cachowanie, pooling zasobów, optymalizację zapytań JPA), monitorować czasy ładowania oraz reagować na problemy ze skalowalnością w środowisku Java EE.</p>	<p>K_U01</p> <p>K_U02</p> <p>K_U03</p> <p>K_U04</p> <p>K_U08</p> <p>K_U09</p> <p>K_U13</p> <p>K_U23</p> <p>K_U24</p>	<p>P6S_UW</p> <p>P6S_UW_INŻ</p> <p>P6S_UO</p> <p>P6S_KK</p> <p>P6S_UK</p>
KOMPETENCJE SPOŁECZNE – absolwent jest	K01. Pracy w zespole, przyjmując w nim różne role.	K_K04	P6S_UO

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

gotów do	<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>	<p>K_K05</p> <p>K_K06</p>	<p>P6S_KR</p> <p>P6S_KK</p>
----------	---	---------------------------	-----------------------------

Lp.	Tematyka zajęć	Liczba godzin
Forma zajęć – laboratorium		
1	Architektura aplikacji webowych w Java EE. Przegląd serwerów aplikacyjnych (WildFly, Payara, TomEE). Architektura z wykorzystaniem Servletów, JSF, JAX-RS, JPA. Wzorce MVC, MVP, MVVM i mikroserwisy (np. integracja z MicroProfile).	2
2	Integracja frontend i backend w JEE. Łączenie interfejsu użytkownika z serwerem aplikacji poprzez RESTful API (JAX-RS). Komunikacja z bazą danych poprzez JPA i kontrola transakcji (JTA). Obsługa asynchroniczna (JMS, WebSockets).	4
3	Zarządzanie stanem aplikacji. CDI (Context and Dependency Injection) w Java EE. Stanowe i bezstanowe EJB. Integracja z mechanizmami zarządzania stanem po stronie frontend (Redux, Vuex).	4
4	Testowanie aplikacji webowych w JEE. Testy jednostkowe (JUnit), integracyjne (Arquillian), end-to-end (Selenium, Cypress). Automatyzacja i integracja testów z procesem CI/CD (Maven, Jenkins).	4
5	Optymalizacja i wydajność w JEE. Techniki przyspieszania ładowania stron, minifikacja kodu, lazy loading. Strategie cachowania (HTTP Cache, ETag, cache drugiego poziomu w JPA). Profilowanie aplikacji (VisualVM, Flight Recorder), analiza logów, monitoring (MicroProfile Metrics). Zaliczenie.	4

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. R. Rhuan, P. Joao, <i>Java EE 8. Wzorce projektowe i najlepsze praktyki</i>, Helion, Warszawa 2019. 2. R. C. Martin, <i>Czysty kod. Podręcznik dobrego programisty</i>, Helion, Gliwice 2010. 3. J. Roszkowski, <i>Analiza i projektowanie strukturalne</i>, Helion, Gliwice, 2004. 4. A. Roman, <i>Testowanie i jakość oprogramowania. Modele, techniki, narzędzia</i>, Wydawnictwo Naukowe PWN, Warszawa 2015.
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. N. Wirth, <i>Algorytmy + struktury danych = programy</i>, WNT, Warszawa 2002 2. B. Eckel, <i>Thinking in Java. Edycja polska</i>, Helion, Gliwice 2006.

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

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