

Karta opisu zajęć - Sylabus

Państwowa Akademia Nauk Stosowanych im. ks. Bronisława Markiewicza w Jarosławiu

I. INFORMACJE PODSTAWOWE

Nazwa zajęć: **Programowanie aplikacji wielowarstwowych Java EE – technologie Hibernate i Spring**

Cykl kształcenia rozpoczynający się w roku akademickim
2022/2023

Nazwa kierunku studiów, poziom i profil kształcenia:

Informatyka, studia pierwszego stopnia, profil praktyczny

Język wykładowy: polski

Rodzaj zajęć: zajęcia kształcenia specjalistycznego

Rok studiów: III

Semestr: 6

Liczba punktów ECTS przypisana zajęciom: 4

Koordinator zajęć

Imię, nazwisko, tytuł/stopień naukowy, adres e-mail:

Jednostka organizacyjna: Wydział Inżynierii Technicznej

FORMA PROWADZENIA ZAJĘĆ I LICZBA GODZIN

Ogólna liczba godzin zajęć dydaktycznych na studiach stacjonarnych i niestacjonarnych z podziałem na formy:

Studia stacjonarne

Studia niestacjonarne

Wykład:

15

Wykład:

Ćwiczenia:

Ćwiczenia:

Laboratorium:

Laboratorium:

Lektorat:

Lektorat:

Projekt:

Projekt:

Zajęcia praktyczne:

30

Zajęcia praktyczne:

Seminarium:

Seminarium:

Zajęcia terenowe:

Zajęcia terenowe:

Praktyki zawodowe:

Praktyki zawodowe:

Inna forma (jaka):

Inna forma (jaka):

RAZEM:

45

RAZEM:

II. INFORMACJE SZCZEGÓŁOWE

Wymagania wstępne i dodatkowe: w zakresie Wiedzy i Umiejętności: Student powinien posiadać wiedzę i umiejętności w zakresie realizowanym w ramach przedmiotu „Programowanie obiektowe”, sem.3, oraz przedmiotu „Współczesne języki programowania”, sem4. wymagania w zakresie Kompetencji społecznych: Zrozumienie potrzeby ciągłego dokształcania się, umiejętność pozyskiwania informacji, umiejętność samokształcenia się.		
Cel (cele) kształcenia dla zajęć: Zapoznanie studentów z zakresu podstaw Framework’a Spring, oraz ORM Hibernate w budowie wielowarstwowych aplikacji internetowych.		
EFEKTY UCZENIA SIĘ OKREŚLONE DLA ZAJĘĆ I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA KIERUNKU STUDIÓW		
Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się UWAGA: Dzielimy efekty uczenia się określone dla zajęć na kategorie wiedzy, umiejętności i kompetencji społecznych. Określone dla zajęć efekty uczenia się nie muszą obejmować wszystkich trzech kategorii i zależą one od formy zajęć.		
Symbol efektów uczenia się określonego dla zajęć*	Treść efektu uczenia się. Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:	Odniesienie do efektów uczenia się określonych dla kierunku studiów (symbol efektów uczenia się)
Wiedzy - zna i rozumie		
E_01	Zasady mapowania relacyjno - obiektowego ORM.	K_W10, K_W20
E_02	Praktyczne zastosowania aplikacji wielowarstwowych wykorzystujących mapowania ORM.	K_W09, K_W10
Umiejętności - potrafi		
E_03	Zaprojektować i wykonać projekt aplikacji w języku Java obsługującej bazę danych z wykorzystaniem Hibernate.	K_U09, K_U18
E_04	Dokonać konfiguracji Hibernate oraz wygenerować pliki XML odwzorowujące klasy aplikacji.	K_U12, K_U18,
E_05	Skonfigurować oraz wykorzystać architekturę Spring w aplikacji internetowej.	K_U18, K_U20
Kompetencji społecznych - jest gotów do		

E_06	potrafi działać w grupie, ma świadomość odpowiedzialności za pracę własną i całego zespołu, rozumie potrzebę systematycznej pracy w celu zdobywania wyższych kompetencji zawodowych.			K_K03, K_K05
UWAGA! Zaleca się, aby w zależności od liczby godzin zajęć, liczba efektów uczenia się zawierała się w przedziale: 3-7, ale są to wartości umowne.				
TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO FORM ZAJĘĆ I METOD OCENIANIA				
Treści programowe (uszczegółowione, zaprezentowane z podziałem na poszczególne formy zajęć, tj. wykład, ćwiczenia, laboratoria, projekty, seminaria i inne):				
Symbol treści programowych	Opis treści programowych	Forma zajęć	Metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się *	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć #
		wykład		
TP_01	Omówienie ogólnych zasad O/RM - Mapowanie Obiektowo Relacyjne. Pojęcia model relacyjny i model obiektowy. Związki pomiędzy tabelami i obiektami. Tworzenie reprezentacji obiektowej dla istniejącego schematu relacyjnej bazy danych oraz tworzenie reprezentacji tabelowej na podstawie istniejących hierarchii klas obiektów.		wykład problemowy – prezentacja, dyskusja	Test, egzamin pisemny
TP_02	Przedstawienie ogólnych zasad działania platformy programistycznej Hibernate. Omówienie zalet Hibernate w porównaniu z innymi podobnymi rozwiązaniami. Rola formatu XML. Pojęcie Klas trwałych. Stany obiektów aplikacji z punktu widzenia Hibernate.		wykład problemowy – prezentacja, dyskusja	Test, egzamin pisemny

TP_03	Zasad odwzorowania klas przy użyciu adnotacji oraz plików XML. Przedstawienie zasad „utrwalania obiektów”, „usuwania obiektów” i aktualizacji danych w relacyjnej bazie danych.		wykład problemowy – prezentacja, dyskusja	Test, egzamin pisemny
TP_04	Język HQL (Hibernate Query Language). Omówienie wspieranych asocjacji pomiędzy obiektami klas trwałych w Hibernate. Możliwości kaskadowej propagacji operacji na obiekty zależne.		wykład problemowy – prezentacja, dyskusja	Test, egzamin pisemny
TP_05	Przedstawienie architektury szkieletowej Spring. Omówienie mechanizmu wstrzykiwania zależności (ang. dependency injection). Wzorec MVC w architekturze Spring.		wykład problemowy – prezentacja, dyskusja	Test, egzamin pisemny
TP_06	Programowanie aspektowe - Spring AOP – jako druga podstawowa technika wykorzystywana w architekturze Spring.		wykład problemowy – prezentacja, dyskusja	Test, egzamin pisemny
TP_07	Przetwarzanie transakcyjne wewnątrz aplikacji wykorzystującej architekturę Spring.		wykład problemowy – prezentacja, dyskusja	Test, egzamin pisemny
		Zajęcia praktyczne		

TP_08	Zajęcia praktyczne dotyczące przygotowania i konfiguracji środowiska Hibernate w dowolnym środowisku IDE. Konfiguracja projektu aplikacji dla bazy danych MySQL lub innego formatu bazy z wykorzystaniem Hibernate . Konfiguracja połączenia z bazą.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_09	Rozbudowa aplikacji poprzez dodanie klas trwałych według reguł POJO. Generowanie metod obsługujących klasy. Zasady automatycznego lub ręcznego generowania struktur bazodanowych na podstawie mapowań.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_10	Analizowanie plików XML odwzorowujących klasy na poszczególne tabele w relacyjnej bazie danych. Analiza i edycja pliku konfiguracyjnego Hibernate.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_11	Tworzenie klas pomocniczych wykorzystujących interfejsy: Hibernate SessionFactory, Session i Transaction. Rozbudowa aplikacji poprzez dodanie formularzy wprowadzania nowych danych.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_12	Praca z obiektami Hibernate (tworzenie i zapisywanie nowych obiektów, odczyt i modyfikacja obiektów , usuwanie obiektów). Testowanie aplikacji z wykorzystaniem przeglądarki internetowej.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów

TP_13	Język HQL. Modyfikacji pisanych wcześniej aplikacji poprzez zastosowanie języka HQL.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_14	Zajęcia praktyczne dotyczące przygotowania i konfiguracji środowiska Spring w dowolnym IDE języka Java. Przygotowanie projektu aplikacji w języku Java – konfiguracja Spring.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_15	Przygotowanie klas komponentów JavaBean w aplikacji wykorzystującej architekturę Spring oraz mechanizm „wstrzykiwania zależności”.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_16	Modyfikacja projektów poprzez zaimplementowanie nowych komponentów JavaBean oraz plików XML w celu wykorzystania wzorca projektowego Spring DAO.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów

Metody weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć, powinny być zróżnicowane w zależności od kategorii, tj. inne dla kategorii wiedza i inne dla kategorii umiejętności i kompetencje społeczne.

Dla wykładu:

* np. wykład podający, wykład problemowy, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy

np. egzamin ustny, test, prezentacja, projekt

Zaleca się podanie przykładowych zadań (pytań) służących weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć.

ZALECANA LITERATURA (w tym pozycje w języku obcym)

Literatura podstawowa (powinna być dostępna dla studenta w uczelnianej bibliotece):

1. McLaughlin B., Edelson J. tł. Piwko Ł.: „Java i XML”, wyd. Helion, 2007.
2. Walls C.: „Spring w akcji”, wyd. Helion, 2020,
3. Dai N., Mandel L., Ryman A.: „Tworzenie aplikacji WWW w języku Java”, wyd. Helion, 2008

Literatura uzupełniająca:

1. Sharma J., Sarin A.: „Spring Framework. Wprowadzenie do tworzenia aplikacji”, wyd. Helion, 2015
2. Bauer Ch., King G., Gregory G.: „Java persistence : programowanie aplikacji bazodanowych w Hibernate”, wyd. Helion, 2017.

III. INFORMACJE DODATKOWE

BILANS PUNKTÓW ECTS

OBCIĄŻENIE PRACĄ STUDENTA (godziny)			
Forma aktywności		Liczba godzin *	
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia		45	
Praca własna studenta		45	
SUMA GODZIN:		90	
OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)			
		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPISANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	Ogółem: 3	1,5
	Praca własna studenta		1,5
* godziny lekcyjne, czyli 1 godz. oznacza 45 min;			
OPIS PRACY WŁASNEJ STUDENTA:			
Praca własna studenta musi być precyzyjnie opisana, uwzględniając charakter praktyczny zajęć. Należy podać symbol efektu uczenia się, którego praca własna dotyczy oraz metody weryfikacji efektów uczenia się stosowane w ramach pracy własnej. Przykładowe formy aktywności: (1) przygotowanie do zajęć, (2) opracowanie wyników, (3) czytanie wskazanej literatury, (4) napisanie raportu z zajęć, (5) przygotowanie do egzaminu, opracowanie projektu.			
Samodzielna praca związana z czytaniem wskazanej literatury oraz przygotowanie do egzaminu (E_01 – E_02). Samodzielna realizacja ćwiczeń programistycznych na podstawie instrukcji, przygotowanie projektów indywidualnych oraz grupowych, których tematyka uzgodniona jest z prowadzącym zajęcia (E_03 – E_06).			
KRYTERIA OCENIANIA			

Zajęcia laboratoryjne kończą się zaliczeniem na ocenę, zaś wykład kończy się egzaminem.

Ocena kształtująca:

Forma i warunki zaliczenia wykładu:

- obecność na wykładach zgodna z Regulaminem Studiów
- Egzamin pisemny
- uzyskanie z egzaminu oceny pozytywnej – co najmniej dostatecznej (min. 51% punktów poprawnych odpowiedzi)

Forma i warunki zaliczenia zajęć praktycznych:

- zaliczenie na ocenę
- frekwencja (zgodnie z Regulaminem Studiów)
- przedłożenie przez studenta nauczycielowi prowadzącemu zajęcia wcześniej ustalonego projektu wraz z opisem we wskazanym przez nauczyciela terminie

Ocena podsumowująca:

Na ocenę dostateczną student zrealizował projekty programistyczne dotyczące opisywanych treści programowych przedmiotu w zakresie podstawowym i potrafi omówić zastosowane techniki. Ponadto w dostatecznym stopniu potrafi wykorzystać przekazaną wiedzę podczas wykładu w realizowanych projektach programistycznych oraz podczas egzaminu pisemnego.

Na ocenę dobrą student zrealizował kompletne projekty programistyczne dotyczące opisywanych treści programowych przedmiotu i potrafi omówić zastosowane techniki oraz uzasadnić ich wybór. Ponadto w dobrym stopniu potrafi wykorzystać przekazaną wiedzę podczas wykładu w realizowanych projektach programistycznych oraz podczas egzaminu pisemnego.

Na ocenę bardzo dobrą student zrealizował kompletne projekty programistyczne dotyczące opisywanych treści programowych przedmiotu z zastosowaniem ponadstandardowych rozwiązań i potrafi omówić zastosowane techniki oraz uzasadnić ich wybór. Ponadto w bardzo dobrym stopniu potrafi wykorzystać przekazaną wiedzę podczas wykładu w realizowanych projektach programistycznych oraz podczas egzaminu pisemnego.

Student podczas zajęć praktycznych wykonuje pod opieką nauczyciela zadania praktyczne, powiązane z treściami programowymi w celu opanowania umiejętności potrzebnych w realizacji indywidualnego lub grupowego projektu zaliczeniowego.

**INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA
KSZTAŁCENIA NA ODLEGŁOŚĆ**

Istnieje możliwość prowadzenia części zajęć (wykład) z wykorzystaniem metod i technik kształcenia na odległość z wykorzystaniem platformy Microsoft Teams zapewniającej synchroniczny, interaktywny sposób komunikowania się, jako formy uzupełniającej kształcenie prowadzone w siedzibie Uczelni.