

Karta opisu zajęć - Sylabus			
Państwowa Wyższa Szkoła Techniczno-Ekonomiczna 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 2023/2024	
Nazwa kierunku studiów, poziom i profil kształcenia: Informatyka, studia pierwszego stopnia, profil praktyczny			
Język wykładowy: polski		Rodzaj zajęć: zajęcia specjalistyczne	
Rok studiów: 3		Semestr: 6	
Liczba punktów ECTS przypisana zajęciom: 4		Koordynator zajęć Imię, nazwisko, tytuł/stopień naukowy, adres e-mail:	
Jednostka organizacyjna: Instytut 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_W08, K_W10, K_W20
E_02	Praktyczne zastosowania aplikacji wielowarstwowych wykorzystujących mapowania ORM.	K_W08, K_W10, K_W20
Umiejętności - potrafi		
E_03	Zaprojektować i wykonać projekt aplikacji w języku Java obsługującej bazę danych z wykorzystaniem Hibernate.	K_U02, K_U05, K_U_09, K_U12, K_U14, K_U18, K_U20
E_04	Dokonać konfiguracji Hibernate oraz wygenerować pliki XML odwzorowujące klasy aplikacji.	K_U02, K_U05, K_U_09, K_U12, K_U14, K_U18, K_U20

E_05	Wykorzystać język HQL w celu pobierania danych z bazy	K_U02, K_U05, K_U_09, K_U12, K_U14, K_U18, K_U20		
E_06	Skonfigurować oraz wykorzystać architekturę Spring w aplikacji internetowej.	K_U02, K_U05, K_U_09, K_U12, K_U14, K_U18, K_U20		
Kompetencji społecznych - jest gotów do				
E_07	Podnoszenia swoich kwalifikacji zawodowych poprzez uczestnictwo w szkoleniach specjalistycznych oraz zdobywanie certyfikatów developerskich.	K_K01, 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 (w bazie danych) i model obiektowy (w aplikacji). 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

TP_02	Przedstawienie ogólnych zasad działania platformy programistycznej Hibernate. Omówienie zalet Hibernate w porównaniu z innymi podobnymi rozwiązaniami. Rola formatu XML – jako pośrednika pomiędzy relacyjną bazą danych a programem opartym na obiektach. Ogólna architektura Hibernate – gdzie Hibernate stanowi warstwę abstrakcji. Zasady konfiguracji aplikacji. Omówienie pojęcia i roli Klas trwałych. Stany obiektów aplikacji z punktu widzenia Hibernate.		wykład problemowy – prezentacja, dyskusja	Test, egzamin
TP_03	Omówienie zasad odwzorowania klas aplikacji przy użyciu adnotacji oraz plików XML. Przedstawienie zasad programistycznego „utrwalania obiektów”, „usuwania obiektów” i aktualizacji danych w relacyjnej bazie danych.		wykład problemowy – prezentacja, dyskusja	Test, egzamin
TP_04	Język HQL (Hibernate Query Language) – jako podstawowy sposób wyszukiwania i odczytu danych z bazy danych. Omówienie wspieranych asocjacji pomiędzy obiektami klas trwałych w Hibernate (1:1, 1:N, N:1). Możliwości kaskadowej propagacji operacji na obiekty zależne.		wykład problemowy – prezentacja, dyskusja	Test, egzamin

TP_05	Przedstawienie architektury szkieletowej Spring. Cechy Spring z punktu widzenia budowy aplikacji webowych. Omówienie wykorzystywanego w architekturze Spring mechanizmu wstrzykiwania zależności (ang. dependency injection). Wzorzec MVC w architekturze Spring.		wykład problemowy – prezentacja, dyskusja	Test, egzamin
TP_06	Programowanie aspektowe - Spring AOP – jako druga podstawowa technika wykorzystywana w architekturze Spring.		wykład problemowy – prezentacja, dyskusja	Test, egzamin
TP_07	Przetwarzanie transakcyjne wewnątrz aplikacji wykorzystującej architekturę Spring. Wykorzystanie właściwości JSP we współpracy z Spring Framework.		wykład problemowy – prezentacja, dyskusja	Test, egzamin
		Zajęcia praktyczne		
TP_08	Zajęcia praktyczne dotyczące przygotowania i konfiguracji środowiska Hibernate w Eclipse. Wstępne przygotowanie projektu aplikacji w języku Java realizującej zapis i odczyt z bazy danych poprzez wykorzystanie Hibernate. Konfiguracja projektu aplikacji dla bazy danych MySQL lub innego formatu bazy. Tworzenie kodu źródłowego konfiguracji połączenia z bazą.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów

TP_09	Rozbudowa aplikacji z poprzednich zajęć poprzez dodanie utworzonych 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	Ćwiczenia praktyczne z analizowania wygenerowanych plików XML odwzorowujących klasy aplikacji na poszczególne tabele w relacyjnej bazie danych. Analiza i edycja pliku konfiguracyjnego Hibernate w: „hibernate.cfg.xml”		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_11	Tworzenie w kodzie źródłowym klas pomocniczych wykorzystujących trzy najważniejsze interfejsy interfejsu programistycznego 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	Zajęcia praktyczne dotyczące pracy z obiektami Hibernate (tworzenie i zapisywanie nowych obiektów, odczyt i modyfikacja obiektów, usuwanie obiektów). Wstępne 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 dodanie w aplikacji nowej metody umożliwiającej odczyt z bazy danych za pomocą zapytania w języku HQL.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_14	Dalszy ciąg modyfikacji aplikacji – dodanie do aplikacji komponentu w postaci tabeli umożliwiającej wyświetlanie danych z bazy oraz ich ewentualne usuwanie poprzez uruchamianie odpowiednich metod – przygotowanie kodów źródłowych.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_15	Zajęcia praktyczne dotyczące przygotowania i konfiguracja środowiska Spring w Eclipse. Przygotowanie pierwszego projektu aplikacji w języku Java – konfiguracja Spring.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP_16	Przygotowanie kodów źródłowych niezbędnych klas komponentów JavaBean w aplikacji wykorzystującej architekturę Spring. Przygotowanie odpowiednich klas oraz komponentów JavaBean w celu wykorzystania mechanizmu „wstrzykiwania zależności”. Przygotowanie pliku konfiguracyjnego XML zawierającego powiązania między obiektami aplikacji.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów

TP_17	Modyfikacja tworzonych projektów poprzez zaimplementowanie nowych komponentów JavaBean oraz plików XML w celu wykorzystania wzorca projektowego Spring DAO w aplikacji odczytującej dane z bazy danych.		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	50
SUMA GODZIN:	95

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: 4	2
	Praca własna studenta		2

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