

SYLABUS
DOTYCZY CYKLU KSZTAŁCENIA 2025/2026-2028/2029
ROK AKADEMICKI 2027/2028

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>Projekt informatyczny</i>
Kod przedmiotu*	
nazwa jednostki prowadzącej kierunek	<i>Instytut Informatyki, Wydział Nauk Ścisłych i Technicznych</i>
Nazwa jednostki realizującej przedmiot	<i>Instytut Informatyki, Wydział Nauk Ścisłych i Technicznych</i>
Kierunek studiów	<i>Informatyka i ekonometria</i>
Poziom studiów	<i>studia pierwszego stopnia</i>
Profil	<i>praktyczny</i>
Forma studiów	<i>studia stacjonarne</i>
Rok i semestr/y studiów	<i>rok III, sem. V</i>
Rodzaj przedmiotu	<i>przedmiot kierunkowy do wyboru</i>
Język wykładowy	<i>polski</i>
Koordynator	<i>dr inż. Bogusław Twaróg</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Bogusław Twaróg</i>

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt. ECTS
5						30			4

1.2. Sposób realizacji zajęć

☒ zajęcia w formie tradycyjnej

☐ zajęcia realizowane z wykorzystaniem metod i technik kształcenia na odległość

1.3. Forma zaliczenia przedmiotu (z toku) (egzamin, zaliczenie z oceną, zaliczenie bez oceny)

Zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

Podstawowe umiejętności z zaliczonych przedmiotów: algorytmy i struktury danych, programowanie obiektowe, bazy danych, inżynieria oprogramowania.

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE

3.1. Cele przedmiotu

C ₁	Przygotowanie studentów do realizacji pełnego cyklu projektu informatycznego od analizy wymagań, przez projektowanie, implementację, testowanie. Rozwijanie kompetencji w zakresie analizy i przetwarzania danych, typowych dla informatyki i ekonometrii, z naciskiem na zastosowanie ich w praktycznych rozwiązaniach.
C ₂	Rozwijanie umiejętności pracy zespołowej, w tym planowania zadań, podziału ról, korzystania z narzędzi do współpracy oraz komunikacji wewnątrz zespołu. Doskonalenie umiejętności programistycznych w kontekście realizacji projektu: tworzenie kodu, integracja modułów, korzystanie z bibliotek i narzędzi wspierających rozwój oprogramowania.
C ₃	Kształtowanie praktycznej znajomości metod projektowych wykorzystywanych w analizie i tworzeniu systemów informatycznych, ze szczególnym uwzględnieniem modeli danych, dokumentacji technicznej i dobrych praktyk programistycznych. Wzmacnianie umiejętności

	prezentacyjnych poprzez przygotowanie i przedstawienie wyników projektu w formie pisemnej i ustnej.
--	---

3.2. Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych
Ek_1	Potrafi zastosować posiadaną wiedzę do rozwiązywania nietypowych problemów informatycznych i ekonometrycznych, także w warunkach niepewności lub niepełnej specyfikacji; wyszukuje brakujące informacje w wiarygodnych źródłach, krytycznie je ocenia i syntetyzuje.	K_Uo2
Ek_2	Planuje i realizuje akwizycję danych, przetwarza pozyskane dane i interpretuje otrzymane wyniki, formułując wnioski adekwatne do problemów inżynierskich i ekonomicznych.	K_Uo5
Ek_3	Dobiera i stosuje metody analityczne, symulacyjne oraz eksperymentalne do formułowania i rozwiązywania zadań inżynierskich w obszarze informatyki i ekonometrii; wykorzystuje właściwe prawa, twierdzenia oraz narzędzia komputerowe.	K_Uo6
Ek_4	Projektuje obiekty informatyczne, takie jak aplikacje czy bazy danych, zgodnie ze specyfikacją opartą na podejściu systemowym; realizuje projekt (w całości lub w części), optymalizując dobór metod i narzędzi.	K_Uo7 K_U10
Ek_5	Potrafi integrować wiedzę informatyczną z wiedzą dziedzinową właściwą dla analizowanego obszaru zastosowań, uwzględniając zarówno aspekt techniczny, jak i pozatechniczny. Odpowiedzialnie uczestniczy w pracach projektowych, uwzględniając współpracę zespołową, otwartość na informację zwrotną oraz dbałość o jakość przygotowywanych rozwiązań.	K_Ko2

3.3. Treści programowe

A. Problematyka zajęć projektowych

- Wprowadzenie do projektu informatycznego - Charakterystyka projektów informatycznych i ekonometrycznych. Cykl życia projektu. - Podejście systemowe i rola specyfikacji. - Analiza problemu i pozyskiwanie informacji - Formułowanie problemu i identyfikacja wymagań. Praca z niepełną lub nieprecyzyjną specyfikacją. - Krytyczna analiza źródeł i poszukiwanie brakujących danych. - Akwizycja i przetwarzanie danych - Planowanie zbierania danych. Źródła danych: publiczne, instytucjonalne, własne. - Wstępne przetwarzanie, czyszczenie, transformacje. Wnioskowanie na podstawie danych. - Metody analityczne, symulacyjne i eksperymentalne - Dobór metod do rozwiązywanego problemu. Wykorzystanie modeli ekonometrycznych, metod statystycznych i symulacji komputerowych. - Narzędzia wspomagające obliczenia i eksperymenty. - Projektowanie obiektów informatycznych - Projektowanie aplikacji, baz danych, modułów analitycznych. - Diagramy UML, ERD, modele logiczne i fizyczne. Dobór architektury i technologii. - Realizacja projektu - Implementacja rozwiązań (częściowa lub pełna). Kontrola wersji i praca w repozytoriach. - Integracja modułów, optymalizacja kodu i narzędzi.
--

7. Testowanie i ocena rezultatów
• Testy funkcjonalne i niefunkcjonalne. Ocena jakości danych, wyników i kodu. • Dokumentacja techniczna i użytkowa.
8. Prezentacja rezultatów projektu
• Przygotowanie raportu końcowego. Prezentacja ustna i dyskusja wyników.

3.4. Metody dydaktyczne

Projekt: Projekt zespołowy lub indywidualny realizowany etapami. Konsultacje projektowe z prowadzącym. Prezentacje częściowe i końcowe.

4. METODY I KRYTERIA OCENY

4.1. Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, ćw, ...)
Ek_1	Ocena analizy problemu w dokumentacji. Ocena kompletności specyfikacji projektowej. Dyskusja podczas prezentacji (uzasadnianie wyborów).	projekt
Ek_2	Ocena etapów pośrednich projektu (plan zbierania danych, przetwarzanie). Sprawdzenie jakości danych i poprawności wniosków w dokumentacji. Test funkcjonalny modułów przetwarzających dane.	projekt
Ek_3	Ocena projektu technicznego (architektura, diagramy). Ocena implementacji (kod, baza danych, logika). Demonstracja działania podczas prezentacji.	projekt
Ek_4	Ocena realizacji projektu (poprawność metod i obliczeń). Przegląd kodu / modeli. Demonstracja działania podczas prezentacji. Ocena spójności projektu i dokumentacji. Obrona przyjętych rozwiązań podczas prezentacji. Ocena sposobu argumentacji.	projekt
Ek_5	Obserwacja pracy na zajęciach. Ocena wkładu indywidualnego (np. historia zmian w repozytorium). Terminowość etapów projektu.	projekt

4.2. Warunki zaliczenia przedmiotu (kryteria oceniania)

• Realizacja projektu 50% (Poprawność rozwiązania; adekwatność metod; jakość implementacji; działanie aplikacji/modelu; integracja danych i technologii) • Dokumentacja projektu 20% (Kompletność; logika; poprawne opisanie danych, metod i wyników; czytelny projekt techniczny) • Prezentacja końcowa i obrona 20% (Jasność prezentacji; umiejętność uzasadniania decyzji; poprawne wyjaśnianie działania rozwiązania) • Aktywność i postępy 10% (Terminowość; konsultacje; wkład w pracę zespołu; systematyczność)
5.0 = 90–100%
4.5 = 80–89%
4.0 = 70–79%
3.5 = 60–69%
3.0 = 50–59%
2.0 < 50%
Zaliczenie przedmiotu potwierdza stopień osiągnięcia przez studenta zakładanych efektów uczenia się.

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	30
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	20
Godziny nie kontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	50
SUMA GODZIN	100
SUMARYCZNA LICZBA PUNKTÓW ECTS	4

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa: 1. Ian Sommerville, Inżynieria oprogramowania. Wydawnictwo Naukowe PWN, 2020 2. Flasiński Mariusz, Zarządzanie projektami informatycznymi, Wyd. Naukowe PWN, 2013 3. Keeling Michael, Zostań architektem oprogramowania Zaprojektuj swoją karierę!, Wydawnictwo Naukowe PWN, 2019
Literatura uzupełniająca: 1. Schildt, H., Java. Kompendium programisty, wydanie IX, Helion, 2015 2. Monika Kośko (red.): Metody i zastosowania ekonometrii współczesnej. Wydawnictwo Uczelniane WSiE TWP, Olsztyn, 2013

Akceptacja Kierownika Jednostki lub osoby upoważnionej