

SYLABUS
DOTYCZY CYKLU KSZTAŁCENIA 2026/2027 - 2029/2030
Rok akademicki 2028/2029

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>inżynieria oprogramowania</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>sztuczna inteligencja</i>
Poziom studiów	<i>studia I stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr/y studiów	<i>rok III, semestr 5</i>
Rodzaj przedmiotu	<i>przedmiot kierunkowy</i>
Język wykładowy	<i>polski</i>
Koordinator	<i>dr inż. Przemysław Pardel</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Przemysław Pardel</i>

* - *opcjonalnie, zgodnie z ustaleniami w Jednostce*

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	15			25					3

1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

1.3 Forma zaliczenia przedmiotu (z toku)

wykład – zaliczenie bez oceny, laboratoria – zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

Znajomość dowolnego języka obiektowego. Umiejętność z zakresu tworzenia aplikacji z wykorzystaniem języka obiektowego. Podstawowe umiejętności z zakresu testowania oprogramowania.

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

3.1 Cele przedmiotu

C1	Zapoznanie studentów z zagadnieniami inżynierii oprogramowania
C2	Zapoznanie studentów z etapami cyklu rozwoju oprogramowania
C3	Zapoznanie studentów z podstawowymi zagadnieniami różnych metodologii zarządzania projektami
C4	Zapoznanie studentów z metodami analizy oprogramowania
C5	Zapoznanie studentów z metodami testowania, utrzymania i rozwoju oprogramowania

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych ¹
EK_01	Zna w zaawansowanym stopniu techniki oraz popularne i specjalistyczne narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z zakresu wytwarzania oprogramowania oraz wybranych zastosowań informatyki i potrafi je zastosować w projekcie informatycznym.	K_Wo7
EK_02	Zna zagadnienia cyklu życia systemów informatycznych i modeli AI, obejmujące przetwarzanie danych w chmurze, wdrażanie i utrzymanie modeli (MLOps) oraz zapewnianie bezpieczeństwa i niezawodności.	K_Wo9
EK_03	Potrafi właściwie zaprojektować oraz zrealizować obiekty informatyczne o komponencie programistycznym, a następnie dokonać weryfikacji i interpretacji rezultatów, oraz sporządzić dokumentację.	K_Uo9
EK_04	Potrafi opracować w języku polskim i angielskim raporty, prezentacje wyników i dokumentację techniczną dla odbiorcy naukowego lub biznesowego, dyskutować posługując się specjalistyczną terminologią	K_U11

3.3 Treści programowe

A. Problematyka wykładu

Wprowadzenie do inżynierii oprogramowania
Proces tworzenia oprogramowania - zagadnienia cyklu życia systemów informatycznych i modeli AI
Rola dokumentacji, planowania i kontroli jakości
Inżynieria wymagań
Analiza oprogramowania
Projektowanie oprogramowania
Testowanie oprogramowania

¹ W przypadku ścieżki kształcenia prowadzącej do uzyskania kwalifikacji nauczycielskich uwzględnić również efekty uczenia się ze standardów kształcenia przygotowującego do wykonywania zawodu nauczyciela.

B. Problematyka laboratoriów

Wprowadzenie i organizacja pracy zespołowej / OKR w praktyce
Modele cyklu życia oprogramowania – case study
Tworzenie wymagań dla analizowanego systemu informatycznego
Analiza wymagań – UML
Modelowanie domeny
Projektowanie architektury
Analiza funkcjonalności systemu
Projektowanie procesów biznesowych
Testowanie aplikacji
Dokumentacja techniczna projektowanego systemu

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną, dyskusja

Laboratorium: projekt, warsztaty

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_01 – EK_02	obecność na wykładzie i aktywne uczestniczenie w dyskusji, test wiedzy	wykład
EK_03 – EK_04	kolokwium, projekt, obserwacja w trakcie zajęć	laboratorium

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

- Zaliczenie przedmiotu odbywa się na podstawie dwóch składowych:
 - zaliczenia wykładu,
 - oceny z laboratorium.
- Zaliczenie laboratorium obejmuje:
 - projekt indywidualny,
 - kolokwium zaliczeniowe,
- Projekt indywidualny polega na opracowaniu rozwiązania konkretnego problemu z zakresu inżynierii oprogramowania oraz osobistej prezentacji wyników projektu.
- Student zalicza laboratorium, jeżeli:
 - uzyska ocenę co najmniej 3.0 z projektu oraz
 - (jeśli kolokwium zostało przeprowadzone) ocenę co najmniej 3.0 z kolokwium zaliczeniowego.
- Student zalicza przedmiot, jeśli:

- | |
|---|
| a. uzyskał zaliczenie z wykładu oraz
b. otrzymał z laboratorium ocenę co najmniej 3.0. |
|---|

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

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny z harmonogramu studiów	40
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	–
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	40
SUMA GODZIN	80
SUMARYCZNA LICZBA PUNKTÓW ECTS	3

* Należy uwzględnić, że 1 pkt ECTS odpowiada 25-30 godzin całkowitego nakładu pracy studenta.

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	–
zasady i formy odbywania praktyk	–

7. LITERATURA

Literatura podstawowa: • Ian Sommerville – Inżynieria oprogramowania (wyd. polskie, Helion) • Roger S. Pressman – Inżynieria oprogramowania. Podejście praktyczne (wyd. polskie, PWN) • Steve McConnell – Kod kompletny (Code Complete, wyd. polskie, Helion) • Frederick P. Brooks – Mityczny osobomiesiąc (wyd. polskie, Helion)
Literatura uzupełniająca: • Jeff Sutherland – Scrum. O zwinnym zarządzaniu projektami (wyd. polskie, Helion) • John Doerr – Rozmierz to! Jak Google, Bono i Gates definiują cele i osiągają sukcesy (wyd. polskie, MT Biznes) • Andrew Hunt, David Thomas – Pragmatyczny programista (wyd. polskie, Helion)