

SYLABUS
DOTYCZY CYKLU KSZTAŁCENIA 2023-2027
ROK AKADEMICKI 2025/2026

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>inżynieria oprogramowania</i>
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	<i>Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>informatyka</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>język polski</i>
Koordynator	<i>dr inż. Maksymilian Knap</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Maksymilian Knap, mgr inż. Ewa Żestawska</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					2

1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

1.3 Forma zaliczenia przedmiotu (z toku)

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 proces inżynierii oprogramowania, jego etapy i potrafi je zastosować w projekcie informatycznym	K_Wo7
EK_02	Zna zagadnienia związane z procesem analizy oprogramowania, potrafi tworzyć diagramy UML, potrafi zrealizować na ich podstawie oprogramowanie	K_Wo8
EK_03	Zna kilka metod zarządzania projektami i rozumie potrzebę ich stosowania	K_Wo7
EK_04	Potrafi tworzyć dokumentację techniczną	K_Uo2
EK_05	Potrafi testować tworzone systemy komputerowe, rozumie rolę testów jednostkowych	K_Uo2
EK_06	Rozumie znaczenie standardów w pracy informatyka i stosuje je	K_Ko5

3.3 Treści programowe

A. Problematyka wykładu

Wprowadzenie do zagadnień inżynierii oprogramowania, warstwy inżynierii oprogramowania
Działania związane z procesem tworzenia oprogramowania
Inżynieria wymagań (wymagania funkcjonalne i poza funkcjonalne)
Metody analizy oprogramowania
Projektowanie oprogramowania (wzorce projektowe)
Testowanie i utrzymanie oprogramowania (metodologia TDD – Test Driven Design)

B. Problematyka ćwiczeń laboratoryjnych

Tworzenie dokumentacji technicznej: RFI, RFP
Tworzenie wymagań dla analizowanego systemu informatycznego
Język UML: diagram przypadków użycia

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

Analiza struktury klas i ich wzajemnych relacji dla analizowanego systemu informatycznego
Język UML: diagram klas
Język UML: diagramy komponentów, pakietów
Analiza funkcjonalności systemu; Język UML: diagram sekwencji
Projektowanie procesów biznesowych; Język UML: diagram aktywności
Projektowanie testów automatycznych
Dokumentacja techniczna projektowanego systemu
Automatyzacja procesu tworzenia dokumentacji
Automatyzacja procesu wytwarzania oprogramowania

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną

Laboratorium: wykonywanie doświadczeń, projektowanie doświadczeń

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_03	test wiedzy	w
EK_04 - EK_05	kolokwium, projekt	lab
EK_06	projekt	lab

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Zaliczenie z wykładu będzie realizowane w formie testu. Próg zaliczenia zostaje ustalony na 50% poprawnych odpowiedzi + 1, dla każdego z efektów EK_01 – EK_03.
Zaliczenie z laboratorium odbędzie się na podstawie (a) zrealizowanego projektu, w którym student rozwiązuje konkretny, zadany problem (b) kolokwium.

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 planu studiów	40
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	20
SUMA GODZIN	60
SUMARYCZNA LICZBA PUNKTÓW ECTS	2

* 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:

Andrzej Jaskiewicz: *Inżynieria oprogramowania*, Helion, 1997

Bogdan Bereza-Jarociński, Bolesław Szomański: *Inżynieria oprogramowania. Jak zapewnić jakość tworzonej aplikacji*, Helion, 2008

Karl E Wieggers, Joy Beatty: *Specyfikacja oprogramowania. Inżynieria wymagań. Wydanie III*, Helion, 2014

Perdita Stevens: *UML. Inżynieria oprogramowania. Wydanie II*, Helion, 2007

Literatura uzupełniająca:

Computer Society: *Guide to the Software Engineering Body of Knowledge (SWEBOK(R))*: Version 3.0, Pierre Bourque, Richard E. Fairley (Editors), IEEE, 2014

Project Management Institute: *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)–Sixth Edition*, PMI, 2019