

SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA 2021-2025

Rok akademicki 2023/2024

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 inżynierski</i>
Język wykładowy	<i>język polski</i>
Koordinator	<i>dr inż. Maksymilian Knap</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Maksymilian Knap, mgr inż. Ewa Żestawska</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	15			15					2

1.2. Sposób realizacji zajęć

zajęcia realizowane częściowo w formie tradycyjnej a częściowo z wykorzystaniem metod i technik kształcenia na odległość

1.3 Forma zaliczenia przedmiotu (z toku)

egzamin

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 Student:	Odniesienie do efektów kierunkowych ¹
EK_01	Zna proces inżynierii oprogramowania, jego etapy i potrafi je zastosować w projekcie informatycznym	K_Wo4
EK_02	Zna zagadnienia, w tym standardy, związane z procesem analizy oprogramowania	K_Wo7, K_W12
EK_03	Zna kilka metod zarządzania projektami i rozumie potrzebę ich stosowania	K_Wo7
EK_04	Potrafi tworzyć dokumentację techniczną, w tym diagramy UML oraz na podstawie dokumentacji potrafi zrealizować oprogramowanie	K_U11, K_U19, K_U21
EK_05	Potrafi testować tworzone systemy komputerowe, rozumie rolę testów jednostkowych	K_U18, K_U19
EK_06	Potrafi stosować wybraną metodę zarządzania projektami i określać niektóre ryzyka związane z realizacją projektu programistycznego	K_U07, K_U19, K_U21
EK_07	Jest gotów do realizacji od początku do końca prostego projektu programistycznego, działając w sposób przedsiębiorczy i w razie potrzeby rozstrzygając różne dylematy	K_Ko3, 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)

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 ćwiczeń laboratoryjnych

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

Ćwiczenia komputerowe: praca w grupie (w zespołach projektowych) i indywidualna.

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	Egzamin pisemny w formie testu	w
EK_02	Egzamin pisemny w formie testu, kolokwium	w, lab
EK_03	Egzamin pisemny w formie testu	w
EK_04	Kolokwium	lab
EK_05	Egzamin pisemny w formie testu, kolokwium	w, lab
EK_06	Obserwacja w trakcie zajęć	lab
EK_07	Obserwacja w trakcie zajęć	lab

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Egzamin będzie realizowany w formie testu.

Służy weryfikacji osiągnięcia efektów EK_01 – EK_03, EK_05. Próg zaliczenia zostaje ustalony na 50% możliwych poprawnych odpowiedzi + 1

Zaliczenie z laboratorium odbędzie się na podstawie:

- zrealizowanego projektu, w którym student rozwiązuje konkretny, zadany problem oraz
- kolokwium
- weryfikowanych na zajęciach laboratoryjnych kompetencji w zakresie efektu EK_07

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)	0
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	20
SUMA GODZIN	50
SUMARYCZNA LICZBA PUNKTÓW ECTS	2

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa: 1. Andrzej Jaskiewicz: <i>Inżynieria oprogramowania</i>, Helion, 1997 2. Bogdan Bereza-Jarociński, Bolesław Szomański: <i>Inżynieria oprogramowania. Jak zapewnić jakość tworzonej aplikacji</i>, Helion, 2008 3. Karl E Wieggers, Joy Beatty: <i>Specyfikacja oprogramowania. Inżynieria wymagań. Wydanie III</i>, Helion, 2014 4. Perdita Stevens: <i>UML. Inżynieria oprogramowania. Wydanie II</i>, Helion, 2007 Literatura uzupełniająca: 1. Computer Society: <i>Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0</i>, Pierre Bourque, Richard E. Fairley (Editors), IEEE, 2014 2. Project Management Institute: <i>A Guide to the Project Management Body of Knowledge (PMBOK® Guide)–Sixth Edition</i>, PMI, 2019
--

Akceptacja Kierownika Jednostki lub osoby upoważnionej