

SYLABUS
DOTYCZY CYKLU KSZTAŁCENIA 2020-2024
ROK AKADEMICKI 2022/2023

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>

* -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			15					2

1.2. Sposób realizacji zajęć

wykład realizowany z wykorzystaniem metod i technik kształcenia na odległość; ćwiczenia 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

K_K05

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych
EK_01	W sposób uporządkowany zna najważniejsze zagadnienia inżynierii oprogramowania; posiada wiedzę szczegółową o procesie inżynierii oprogramowania, zna jego etapy. Wiedzę wykorzystuje w projektach.	K_Wo4
EK_02	Zna zagadnienia związane z procesem analizy oprogramowania, potrafi tworzyć diagramy UML, potrafi zrealizować na ich podstawie oprogramowanie. Zna kilka metod zarządzania projektami i rozumie potrzebę ich stosowania	K_Wo7
EK_03	Rozumie znaczenie i zna cechy dobrej dokumentacji technicznej zadania informatycznego	K_W12
EK_04	Potrafi opracować specyfikację problemu informatycznego oraz tworzyć dokumentację techniczną do wyspecyfikowanego zadania, używając do tego odpowiednich narzędzi i stosując standardy.	K_U11, K_U19, K_U21
EK_05	Potrafi samodzielnie i dzieląc się zadaniami w grupie testować tworzone systemy komputerowe, rozumie rolę testów jednostkowych	K_U18, K_U19
EK_06	Potrafi dostrzec różne aspekty ryzyka związanego z przedsięwzięciem informatycznym, szanse biznesowe i korzyści lub niebezpieczeństwa społeczne	K_U07, K_K03, K_K05
EK_07	Jest częściowo przygotowany do myślenia i działania w sposób przedsiębiorczy	K_K03

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
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	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	projekt, dyskusja w trakcie zajęć	LAB
EK_07	dyskusja w trakcie zajęć	LAB

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

<p>Zaliczenie z wykładu będzie realizowane w formie testu. Próg zaliczenia zostaje ustalony na 50% poprawnych odpowiedzi + 1</p> <p>Zaliczenie z laboratorium odbędzie się na podstawie</p> <ol style="list-style-type: none"> zrealizowanego i „obronionego” projektu, w którym student rozwiązuje zadany problem kolokwium udziału w dyskusji podczas zajęć
--

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)	
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

<p>Literatura podstawowa:</p> <p>Andrzej Jaskiewicz: <i>Inżynieria oprogramowania</i>, Helion, 1997</p> <p>Bogdan Bereza-Jarociński, Bolesław Szomański: <i>Inżynieria oprogramowania. Jak zapewnić jakość tworzonym aplikacjom</i>, Helion, 2008</p> <p>Karl E Wieggers, Joy Beatty: <i>Specyfikacja oprogramowania. Inżynieria wymagań. Wydanie III</i>, Helion, 2014</p> <p>Perdita Stevens: <i>UML. Inżynieria oprogramowania. Wydanie II</i>, Helion, 2007</p>
<p>Literatura uzupełniająca:</p> <p>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</p> <p>Project Management Institute: <i>A Guide to the Project Management Body of Knowledge (PMBOK® Guide)–Sixth Edition</i>, PMI, 2019</p>

Akceptacja Kierownika Jednostki lub osoby upoważnionej