

SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA 2020-2024

(skrajne daty)

Rok akademicki 2022/2023

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	Testowanie i wdrażanie systemów informatycznych
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	<i>Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Kolegium Nauk Przyrodniczych, Instytut Informatyki</i>
Kierunek studiów	<i>Informatyka i ekonometria</i>
Poziom studiów	<i>studia I stopnia</i>
Profil	<i>praktyczny</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr/y studiów	<i>rok III semestr 6</i>
Rodzaj przedmiotu	<i>przedmiot specjalnościowy</i>
Język wykładowy	<i>język polski</i>
Koordynator	<i>dr inż. Marcin Ochab</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	

* -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
6	10			10					2

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

PODSTAWOWA WIEDZA DOTYCZĄCA ALGORYTMÓW I STRUKTURY DANYCH, PROGRAMOWANIA OBIEKTOWEGO

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE**3.1 Cele przedmiotu**

C1	Zapoznanie studenta z metodykami testowania systemów informatycznych
----	--

C2	Zapoznanie z narzędziami służącymi testowaniu systemów informatycznych
C3	Zaznajomienie z narzędziami automatyzującymi testowanie
C4	Nabycie umiejętności tworzenia testów białoskrzynkowych
C5	Nabycie umiejętności korzystania z zalet programowania obiektowego oraz refaktoryzacji w celu poprawienia jakości oprogramowania i wydajniejszego testowania

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 cykl powstawania oprogramowania i rozumie kluczową rolę testowania w tym procesie.	K_W02
EK_02	Zna rolę serwera ciągłej integracji w procesie utrzymywania systemów informatycznych.	K_W02
EK_03	Zna i umie zastosować podstawowe narzędzia pozwalające testować oprogramowanie.	K_U05
EK_04	Potrafi dobrać odpowiednie narzędzia do przeprowadzenia odpowiedniego typu testów oprogramowania.	K_U09
EK_05	Umie samodzielnie wybrać i zaproponować odpowiednie do problemu narzędzia programistyczne.	K_U12
EK_06	Rozumie rolę dokumentacji oraz testów w powstawaniu złożonych systemów informatycznych tworzonych zespołowo i umie ją tworzyć.	K_U10

3.3 Treści programowe

A. Problematyka wykładu

Treści merytoryczne:
Kluczowa rola testów w tworzeniu oprogramowania i ich rodzaje
Testy jednostkowe na przykładzie JUnit oraz Mock
Mechanizmy umożliwiające działanie frameworków oferujących testy jednostkowe: asercje, adnotacje, refleksje oraz przykłady ich użycia
Testowanie interfejsów graficznych z użyciem Selenium
Idee test driven development (TDD) oraz behavior-driven development (BDD)
Przykłady testów wykorzystujących język naturalny korzystających z Behave
Rola serwera ciągłej integracji w procesie testowania na przykładzie Jenkinsa oraz Dockera
Rola refaktoryzacji w powstawaniu dobrego 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 ćwiczeń audytoryjnych, konwersatoryjnych, laboratoryjnych, zajęć praktycznych

Treści merytoryczne:
Tworzenie testów jednostkowych z wykorzystaniem JUnit
Wykorzystywanie mechanizmów asercji, adnotacji i refleksji
Testowanie oprogramowania opartego na wyjątkach
Refaktoryzacja kodu przy wykorzystaniu zalet programowania obiektowego
Wykorzystanie narzędzi Ant, GIT oraz GitHub w celu umożliwienia wdrożenia ciągłej integracji

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną

Ćwiczenia Laboratoryjne: praca w grupach (rozwiązywanie zadań, dyskusja).

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	zaliczenie	W, LAB
EK_02	kolokwium	LAB
EK_03	kolokwium	LAB
EK_04	kolokwium	LAB
EK_05	kolokwium	LAB
EK_06	kolokwium	LAB

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Efekt	Ocena	Kryteria oceny
EK_01	dst	Umie wytłumaczyć rolę testowania w cyklu wytwórczym oprogramowania
	db	Potrafi wskazać różnicę pomiędzy testami czarno- i białoskrzynkowymi oraz wyjaśnić ich rolę
	bdb	Potrafi wskazać różne kategorie testów i je porównać
EK_02	dst	Potrafi scharakteryzować rolę serwera ciągłej integracji
	db	Potrafi wymienić wymagane elementy pozwalające na wdrożenie ciągłej integracji
	bdb	Potrafi wymienić min. dwa suplementarne elementy serwera ciągłej integracji
EK_03	dst	Umie napisać test jednostkowy, wie gdzie jego tworzenie ma sens
	db	Umie napisać testy jednostkowe testujące wyjątki, czas obliczeń oraz wykorzystujące wspólne funkcje przygotowujące dane
	bdb	Umie napisać sparametryzowany test jednostkowy
EK_04	dst	Umie wymienić trzy narzędzia do wykonania testów i je scharakteryzować
EK_05	db	Umie przyporządkować odpowiednie narzędzia odpowiednim rodzajom testów
	bdb	Umie zaplanować wielopoziomą procedurę testowania dobierając odpowiednie oprogramowanie
EK_06	dst	Rozumie idee behavior-driven development (BDD)
	db	Umie uzasadnić potrzebę powstania testów wykorzystujących język naturalny
	bdb	Umie przedstawić przykładowy test wykorzystujący język naturalny

Zasady uzyskania oceny końcowej:

Zaliczenie laboratorium następuje na podstawie zaliczenia wszystkich efektów weryfikowanych przez planowane metody weryfikacji. Przy czym zakłada się, że każda metoda weryfikacji dostarcza osobne oceny dla każdego z weryfikowanych przez nią efektów uczenia się.

Zaliczenie wykładu może nastąpić tylko po zaliczeniu na ocenę pozytywną laboratorium

Kryteria oceny:

Student otrzymuje ocenę **niedostateczny**, gdy metody weryfikacji wykażą, iż co najmniej jeden z efektów nie został osiągnięty (średnia ocena dla tego efektu jest niższa niż 3.0);

Student otrzymuje ocenę **dostateczny**, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 3.0, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 3.75;

Student otrzymuje ocenę **dobry**, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 3.75, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 4.75;

Student otrzymuje ocenę **bardzo dobry**, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 4.75.

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	20
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	30
SUMA GODZIN	50
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:

Ciągłe dostarczanie oprogramowania: automatyzacja kompilacji, testowania i wdrażania / Jez Humble, David Farley ; [tł.: Dorota Konowrocka-Sawa]. - Gliwice: Helion, cop. 2015.

Java : obsługa wyjątków, usuwanie błędów i testowanie kodu / Stephen Stelting ; [tł. z ang. Adam Bochenek]. - Gliwice : "Helion", cop. 2005.

Testowanie i jakość oprogramowania : modele, techniki, narzędzia / Adam Roman. - Warszawa : Wydawnictwo Naukowe PWN SA, cop. 2015.

Testowanie oprogramowania : podręcznik dla początkujących / Rafał Pawlak. - Gliwice : Wydawnictwo Helion, cop. 2014.

Literatura uzupełniająca:

Glenford J. Myers i inni: Sztuka testowania oprogramowania, Helion 2005

Testowanie oprogramowania w praktyce : studium przypadków 2.0 / red. nauk. Adam Roman, Karolina Zmitrowicz ; [aut.] Wojciech Anzel, Maciej Chmielarz, Aleksandra Kornecka, Natalia Krawczyk, Ewa Marchewka, Rafał Nazwański, Adam Romanowicz, Aleksandra Pirek, Bartłomiej Prędko, Aleksandra Sasin, Adam Stankiewicz, Karolina Zmitrowicz. - Warszawa : Wydawnictwo Naukowe PWN, 2018.

Akceptacja Kierownika Jednostki lub osoby upoważnionej