

Załącznik nr 1.5 do Zarządzenia Rektora UR nr 61/2025

SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA 2025-2029

(skrajne daty)

Rok akademicki 2028/2029

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	Przedmiot obieralny: testowanie i wdrażanie systemów informatycznych
Kod przedmiotu*	
nazwa jednostki prowadzącej kierunek	Instytut Informatyki, Wydział Nauk Ścisłych i Technicznych
Nazwa jednostki realizującej przedmiot	Instytut Informatyki, Wydział Nauk Ścisłych i Technicznych
Kierunek studiów	Informatyka i ekonometria
Poziom studiów	studia pierwszego stopnia
Profil	praktyczny
Forma studiów	studia stacjonarne
Rok i semestr/y studiów	rok IV, sem. VII
Rodzaj przedmiotu	przedmiot kierunkowy do wyboru
Język wykładowy	polski
Koordynator	dr inż. Marcin Ochab
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
7	10			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	Nabywanie 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
EK_1	Zna cykl powstawania oprogramowania i rozumie kluczową rolę testowania w tym procesie.	K_Wo1, K_Wo2
EK_2	Zna rolę serwera ciągłej integracji w procesie utrzymywania systemów informatycznych.	K_Wo2
EK_3	Zna i umie zastosować podstawowe narzędzia pozwalające testować oprogramowanie	K_Uo1
EK_4	Umie samodzielnie wybrać i zastosować odpowiednie do problemu narzędzia integrując wiedzę informatyczną z dziedzinową	K_Uo2, K_Uo7
EK_5	Rozumie rolę dokumentacji oraz testów w powstawaniu złożonych systemów informatycznych, potrafi ją wyjaśnić otoczeniu stosując specjalistyczną terminologię	K_Uo2, K_U13

3.3 Treści programowe

A. Problematyka wykładów

1. Kluczowa rola testów w tworzeniu oprogramowania i ich rodzaje.
2. Testy jednostkowe na przykładzie JUnit oraz Mock.
3. Mechanizmy umożliwiające działanie frameworków oferujących testy jednostkowe: asercje, adnotacje, refleksje oraz przykłady ich użycia.
4. Testowanie interfejsów graficznych z użyciem Selenium.
5. Idee test driven development (TDD) oraz behavior-driven development (BDD).
6. Przykłady testów wykorzystujących język naturalny korzystających z Behave.
7. Rola serwera ciągłej integracji w procesie testowania na przykładzie Jenkinsa oraz Dockera.
8. Rola refaktoryzacji w powstawaniu dobrego oprogramowania.

B. Problematyka laboratorium

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

C. Problematyka zajęć projektowych

1. Ustalenie tematyki projektów
2. Dobór techniki testowania
3. Realizacja testowania
4. Dyskusja nad raportami z testów

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną

Laboratorium: praca w grupach (rozwiązywanie zadań, dyskusja)

Zajęcia projektowe: testowanie wybranego oprogramowania

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_1	kolokwium, obserwacja w trakcie zajęć, dyskusja	w, lab., proj.
EK_2	kolokwium, obserwacja w trakcie zajęć, dyskusja	w, lab., proj.
EK_3	kolokwium, obserwacja w trakcie zajęć, dyskusja	lab., proj.
EK_4	kolokwium, obserwacja w trakcie zajęć, dyskusja	lab., proj.
EK_5	kolokwium, obserwacja w trakcie zajęć, dyskusja	lab., proj.

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

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 sparymetryzowany test jednostkowy

EK_04

dst Umie wymienić trzy narzędzia do wykonania testów i je scharakteryzować

db Umie przyporządkować odpowiednie narzędzia odpowiednim rodzajom testów

bdb Umie zaplanować wielopoziomową procedurę testowania dobierając odpowiednie oprogramowanie

EK_05

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 i projektu 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:

niedostateczny, gdy co najmniej jeden z efektów nie został osiągnięty (średnia ocena dla tego efektu jest niższa niż 3.0);

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;
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;
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	30
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	2
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	18
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:

1. Ciągłe dostarczanie oprogramowania: automatyzacja kompilacji, testowania i wdrażania / Jez Humble, David Farley ; [tł.: Dorota Konowrocka-Sawa]. - Gliwice: Helion, cop. 2015.
2. Java : obsługa wyjątków, usuwanie błędów i testowanie kodu / Stephen Stelting ; [tł. z ang. Adam Bochenek]. - Gliwice : "Helion", cop. 2005.
3. Testowanie i jakość oprogramowania : modele, techniki, narzędzia / Adam Roman. - Warszawa : Wydawnictwo Naukowe PWN SA, cop. 2015.
4. Testowanie oprogramowania : podręcznik dla początkujących / Rafał Pawlak. - Gliwice : Wydawnictwo Helion, cop. 2014.

Literatura uzupełniająca:

1. Glenford J. Myers i inni: Sztuka testowania oprogramowania, Helion 2005
2. Testowanie oprogramowania w praktyce: studium przypadków 2.0 Wojciech Anzel i inni - Warszawa : Wydawnictwo Naukowe PWN, 2018.

Akceptacja Kierownika Jednostki lub osoby upoważnionej