

Karta opisu zajęć - Sylabus

Państwowa Wyższa Szkoła Techniczno-Ekonomiczna im. ks. Bronisława Markiewicza w Jarosławiu

I. INFORMACJE PODSTAWOWE

Nazwa zajęć: Specyfikacje i testowanie programów	Cykl kształcenia rozpoczynający się w roku akademickim 2021/2022
Nazwa kierunku studiów, poziom i profil kształcenia: Informatyka I stopień, profil praktyczny	
Język wykładowy: polski	Rodzaj zajęć: zajęcia specjalistyczne
Rok studiów: III	Semestr: 5
Liczba punktów ECTS przypisana zajęciom: 3	Koordinator zajęć Imię, nazwisko, tytuł/stopień naukowy, adres e-mail:
Jednostka organizacyjna: Instytut Inżynierii Technicznej	

FORMA PROWADZENIA ZAJĘĆ I LICZBA GODZIN

Ogólna liczba godzin zajęć dydaktycznych na studiach stacjonarnych i niestacjonarnych z podziałem na formy:

Studia stacjonarne		Studia niestacjonarne	
Wykład:	15	Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:		Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:	30	Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki zawodowe:		Praktyki zawodowe:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	45	RAZEM:	

II. INFORMACJE SZCZEGÓŁOWE

Wymagania wstępne i dodatkowe:

- umiejętność budowania i realizowania algorytmów w języku imperatywnym,
- znajomość podstaw programowania obiektowego (język JAVA),
- znajomość i umiejętność stosowania logiki oraz dowodzenia twierdzeń,
- wiedza w zakresie kodowania liczb całkowitych i zmiennoprzecinkowych w komputerach.

Cel (cele) kształcenia dla zajęć:

Zasadniczym celem przedmiotu jest zapoznanie studentów z problematyką specyfikowania wymagań funkcjonalnych dla aplikacji tworzonych w językach imperatywnych i obiektowych. Na podstawie specyfikacji studenci zapoznają się z metodami formalnej weryfikacji (w zakresie podstawowym, bazując na metodzie Hoare-Floyda) i testowania prostych programów z wykorzystaniem specjalizowanych narzędzi tj. JUnit. Pozyskują teoretyczną wiedzę w zakresie różnych metod testowania bardziej złożonych programów.

EFEKTY UCZENIA SIĘ OKREŚLONE DLA ZAJĘĆ I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA KIERUNKU STUDIÓW

Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się

UWAGA:

Dzielimy efekty uczenia się określone dla zajęć na kategorie wiedzy, umiejętności i kompetencji społecznych. Określone dla zajęć efekty uczenia się nie muszą obejmować wszystkich trzech kategorii i zależą one od formy zajęć.

Symbol efektów uczenia się określonego dla zajęć*	Treść efektu uczenia się. Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:	Odniesienie do efektów uczenia się określonych dla kierunku studiów (symbol efektów uczenia się)
Wiedzy - zna i rozumie		
M_01	Student zna metody specyfikowania programów w językach imperatywnych. Ma wiedzę w zakresie stosowania metody Hoare-Floyda. Zna pojęcia: warunek wstępny, warunek końcowy, niezmiennik pętli. Student posiada elementarną wiedzę w zakresie weryfikacji symbolicznej programów. Rozumie ograniczenia metody i zakres jej stosowania. Posiada ogólne informacje o automatycznej weryfikacji i jej złożoności obliczeniowej.	K_W01, K_W06
M_02	Student zna metody definicji wymagań i testowania programów realizowanych w językach obiektowych. Zna i rozumie paradygmaty testowania.	K_W01
Umiejętności - potrafi		
M_03	Student umie specyfikować i przetestować symbolicznie proste programy stosując metodę Hoare-Floyda. Potrafi stosować zasadę indukcji matematycznej zupełnej i określić niezmiennik pętli.	K_U11, K_U21
M_04	Student potrafi stosować w praktyce narzędzie do testowania programów w języku obiektowym. W szczególności potrafi opracować plan testów i zrealizować klasę testującą.	K_U07, K_U08, K_U15
M_05	Potrafi zaplanować i zrealizować zadanie indywidualne oraz opracować i zaprezentować grupie wyniki swojej pracy.	K_U02, K_U03, K_U04
Kompetencji społecznych - jest gotów do		
M_06	Student rozumie znaczenie poprawności programów jako produktu użytkowego, rozumie jakie skutki niesie używanie błędnych programów. Student potrafi oszacować koszty testowania i ich wpływ na cykl produkcyjny programu.	K_K03

UWAGA!

Zaleca się, aby w zależności od liczby godzin zajęć, liczba efektów uczenia się zawierała się w przedziale: 3-7, ale są to wartości umowne.

TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO FORM ZAJĘĆ I METOD OCENIANIA

Treści programowe (uszczegółowione, zaprezentowane z podziałem na poszczególne formy zajęć, tj. wykład, ćwiczenia, laboratoria, projekty, seminaria i inne):

Symbol treści programowych	Opis treści programowych	Forma zajęć	Metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się *	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć #
		wykład		
TP_01	Testowanie programów – wprowadzenie. Przykłady najpoważniejszych skutków błędów. Rola testowania w tworzeniu oprogramowania. Ważniejsze definicje.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_02	Weryfikacja a testowanie programów. Znaczenie i tworzenie specyfikacji programów. Cechy specyfikacji. Definicje poprawności programów w językach imperatywnych. Zarys symbolicznej metody Hoare-Floyda.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_03	Symboliczna specyfikacja i weryfikacja programów – przykłady. Pojęcie niezmiennika pętli. Praktyczne uwagi wynikające z ograniczeń reprezentacji danych liczbowych w komputerach. Przykładowe programy weryfikatorów, ich ograniczenia i złożoność obliczeniowa. Metoda indukcji matematycznej zupełnej.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_04	Aksjomaty testowania. Definicje błędu. Poziomy testowania oprogramowania. Testowanie w cyklu życia oprogramowania. Koszty błędów.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne

TP_05	Modele programów. Testowanie metodą białej skrzynki. Standardy i reguły kodowania. Asercje w JUNIT. Analiza pokrycia kodu.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_06	Testowanie mutacyjne. Testowanie metodą czarnej skrzynki. Testowanie danych – warunki graniczne. Wartości specjalne i transcendentne. Metoda klas równoważności. Metoda Monte-Carlo i metody genetyczne.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_07	Testy jednostkowe. Automatyzacja testów. Obszary testowania. Poprawność wyników. Warunki brzegowe. Odwrócenie relacji.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_08	Kontrola wyników na wiele sposobów. Wymuszanie warunków powstawania błędów. Charakterystyka efektywnościowa. Obiekty imitacji. Pułapki testowania. Testy ognia. Arytmetyka zmiennoprzecinkowa.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_09	Środowiska testowania jednostkowego: JUNIT, testNG, <i>framework</i> Microsoft, NUnit – przykłady. Biblioteki do budowy obiektów imitacji.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_10	Testowanie wydajności. Parametry wydajności. Metodyka testowania wydajnościowego. Automatyzacja przypadków testowych. Dostrajanie wydajności, wykonywanie <i>benchmarków</i> . Narzędzia do testowania wydajnościowego.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_11	Testowanie regresyjne – typy testów, wybór momentu testowania. <i>Smoke test</i> . Testowanie doraźne. Metody testowania <i>ad-hoc</i> . Testowanie koleżeńskie, parami i badawcze. testowanie iteracyjne.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne

TP_12	Testowanie zwinne i ekstremalne. Specyfika i testowanie systemów obiektowych. Testowanie klas. Testowanie integracyjne. Testy systemowe i współdziałania. Testowanie własności klas. Testowanie użyteczności i dostępności.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_13	Testowanie estetyki i dostępności. Identyfikacja wymagań w zakresie zasobów. Ocena rozmiaru i wysiłku koniecznego do wykonania testów. Podział i planowanie zadań.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
TP_14	Zarządzanie testami. Wymagania stawiane testom. Metryki przydatne w testowaniu. Wariancja wysiłku.		Wykład podający, prezentacje multimedialne	zaliczenie końcowe ustne
		zajęcia praktyczne		
TK_15	Specyfikacja i weryfikacja symboliczna prostych programów w języku C. Instrukcja przypisania i wyboru. Określanie pre- i postwarunków jako elementów metody Hoare-Floyda.		rozwiązywanie zadań przy tablicy	indywidualna bieżąca ocena pracy
TK_16	Wprowadzenie do specyfikacji i weryfikacji symbolicznej programów z pętlą. Formułowanie niezmiennika pętli.		rozwiązywanie zadań przy tablicy	indywidualna bieżąca ocena pracy
TK_17	Specyfikacja i weryfikacja funkcji rekurencyjnych metodą indukcji matematycznej zupełnej.		rozwiązywanie zadań przy tablicy	indywidualna bieżąca ocena pracy
TK_18	Wprowadzenie do środowiska testowego JUnit. Instalacja i konfiguracja oprogramowania. Pierwsze testy. Prezentacja przykładowych programów do przetestowania indywidualnego przez studentów.		prezentacja multimedialna, praca indywidualna	indywidualna bieżąca ocena pracy

TK_19	Testowanie modułów przy pomocy JUnit. Zastosowanie praktyczne metod testujących środowiska JUnit. Dziedziczenie po klasie TestCase. Asercje JUnit i ich zastosowanie praktyczne. Omówienie problemów napotykanym przy testowaniu.		prezentacja multimedialna, praca indywidualna	indywidualna bieżąca ocena pracy
TK_20	Przygotowanie scenariuszy (przypadków testowych) dla testowanej przykładowej klasy. Dyskusja dotycząca możliwych wyjątków wejścia/wyjścia i wynikających z ograniczeń implementacji typów w języku JAVA.		prezentacja multimedialna, praca indywidualna	indywidualna bieżąca ocena pracy
TK_21	Indywidualna realizacja wybranych przypadków testowych i ich wykonanie. opracowanie raportu z wykonanego zadania – miniprojekt.		praca indywidualna	indywidualna ocena końcowa pracy (sprawozdania) pracy
TK_22	Prezentacja wyników testowania – dyskusja w grupie. Omówienie cech sporządzonej indywidualnej dokumentacji. Korekta błędów.		indywidualna prezentacja wyników realizowanych testów (sprawozdania)	indywidualna ocena końcowa pracy (sprawozdania) pracy

Metody weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć, powinny być zróżnicowane w zależności od kategorii, tj. inne dla kategorii wiedza i inne dla kategorii umiejętności i kompetencje społeczne.

Dla wykładu:

* np. wykład podający, wykład problemowy, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy

np. egzamin ustny, test, prezentacja, projekt

Zaleca się podanie przykładowych zadań (pytań) służących weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć.

ZALECANA LITERATURA (w tym pozycje w języku obcym)

Literatura podstawowa (powinna być dostępna dla studenta w uczelnianej bibliotece):

1. A. Roman, Testowanie i jakość oprogramowania: modele, techniki, narzędzia, PWN, 2017, Wyd 1 i 2.
2. B. Wiszniewski, B. Bereza-Jarociński: *Teoria i praktyka testowania programów*. Wydawnictwo PWN, 2006.
3. Andy Hunt, Dave Thomas: *JUnit. Pragmatyczne testy jednostkowe w Javie*. Helion 2006

Literatura uzupełniająca:

1. S. Stelting: *Java. Obsługa wyjątków, usuwanie błędów i testowanie kodu*. Wydawnictwo Helion, 2005
2. V. Massol: *JUnit in action*. wyd. Manning, Greenwich, 2012
3. R. Patton: *Testowanie oprogramowania*. Wydawnictwo MIKOM, 2002.

III. INFORMACJE DODATKOWE

BILANS PUNKTÓW ECTS			
OBCIĄŻENIE PRACĄ STUDENTA (godziny)			
Forma aktywności		Liczba godzin *	
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia		45	
Praca własna studenta		25	
SUMA GODZIN:		70	
OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)			
		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPISANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	Ogółem: 3	2
	Praca własna studenta		1
* godziny lekcyjne, czyli 1 godz. oznacza 45 min;			
OPIS PRACY WŁASNEJ STUDENTA:			
Praca własna studenta musi być precyzyjnie opisana, uwzględniając charakter praktyczny zajęć. Należy podać symbole efektu uczenia się, którego praca własna dotyczy oraz metody weryfikacji efektów uczenia się stosowane w ramach pracy własnej. Przykładowe formy aktywności: (1) przygotowanie do zajęć, (2) opracowanie wyników, (3) czytanie wskazanej literatury, (4) napisanie raportu z zajęć, (5) przygotowanie do egzaminu, opracowanie projektu.			
Czytanie wskazanej literatury (10 godzin lekcyjnych), przygotowanie sprawozdania z wykonanych testów(20 godzin lekcyjnych), przygotowanie do zaliczenia (10 godzin lekcyjnych)			
KRYTERIA OCENIANIA			
Ocena kształtująca: ocena przygotowania do zajęć ocena umiejętności stosowania zdobytej wiedzy podczas wykonywania zadań realizowanych podczas zajęć ocena aktywności podczas zajęć			
Ocena podsumowująca: ocena umiejętności rozwiązywania postawionych problemów ocena zadań praktycznych do samodzielnego wykonania ocena sprawozdania z realizowanych testów ocena końcowego zaliczenia ustnego			
INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA KSZTAŁCENIA NA ODLEGŁOŚĆ			