

Uproszczony opis zajęć - Sylabus

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

I. INFORMACJE OGÓLNE

Nazwa zajęć			Kod zajęć:	
Specyfikacje i testowanie programów				
Nazwa kierunku studiów, poziom i profil kształcenia:		Informatyka I stopień, profil praktyczny		
Język wykładowy: polski	Rodzaj zajęć:		Przedmiot obieralny	
Rok studiów: III	Semestr: V	Liczba punktów ECTS przypisana zajęciom:		3
Instytut (Zakład) odpowiedzialny za zajęcia:		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:		Praktyki:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	45	RAZEM:	

II. INFORMACJE SZCZEGÓŁOWE

Przypisane do zajęć efekty uczenia się w zakresie wiedzy, umiejętności oraz kompetencji społecznych i odniesienie ich do efektów uczenia się dla określonego kierunku studiów, poziomu i profilu.

Symbol efektów uczenia się przypisane-go do zajęć	Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:	Odniesienie do efektów uczenia się dla określonego kierunku studiów, poziomu i profilu
	Wiedzy	
M_01	Student zna metody specyfikowania programów w językach imperatywnych. Ma wiedzę w zakresie 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	
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	K_U02, K_U03,

	grupie wyniki swojej pracy.	K_U04
	Kompetencji społecznych	
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

TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ PRZYPISANYCH DO ZAJĘĆ

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	Liczba godzin	Odniesienie do 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	1	M_01, M06
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	2	M_01, M06
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	1	M_01
TP_04	Aksjomaty testowania. Definicje błędu. Poziomy testowania oprogramowania. Testowanie w cyklu życia oprogramowania. Koszty błędów.	wykład podający, przykłady programów	1	M_02
TP_05	Modele programów. Testowanie metodą białej skrzynki. Standardy i reguły kodowania. Asercje w JUNIT. Analiza pokrycia kodu.	wykład podający, prezentacja wybranych urządzeń	1	M_02
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, przykłady programów	1	M_02
TP_07	Testy jednostkowe. Automatyzacja testów. Obszary testowania. Poprawność wyników. Warunki brzegowe. Odwrócenie relacji.	wykład podający	1	M_02
TP_08	Kontrola wyników na wiele sposobów. Wymuszanie warunków powstawania błędów. Charakterystyka efektywnościowa. Obiekty imitacji. Pułapki testowania. Testy ognia. Arytmetyka zmiennooprzecinkowa.	wykład podający, przykłady programów	1	M_02
TP_09	Środowiska testowania jednostkowego: JUNIT, testNG, <i>framework</i> Microsoft, NUnit – przykłady. Biblioteki do budowy obiektów imitacji.	wykład podający, przykłady programów	1	M_02
TP_10	Testowanie wydajności. Parametry wydajności. Metodyka testowania wydajnościowego. Automatyzacja przypadków testowych. Dostrajanie wydajności, wykonywanie <i>benchmarków</i> . Narzędzia do testowania wydajnościowego.	wykład podający, przykłady programów	1	M_02
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, przykłady programów	1	M_02
TP_12	Testowanie zwinne i ekstremalne. Specyfika i testowanie	wykład podający	1	M_02

	systemów obiektowych. Testowanie klas. Testowanie integracyjne. Testy systemowe i współdziałania. Testowanie własności klas. Testowanie użyteczności i dostępności.	cy		
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	1	M_02
TP_14	Zarządzanie testami. Wymagania stawiane testom. Metryki przydatne w testowaniu. Wariancja wysiłku.	wykład podający	1	M_01, M_02
		ćwiczenia		
		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 przykładów przy tablicy	2	M_03
TK_16	Wprowadzenie do specyfikacji i weryfikacji symbolicznej programów z pętlą. Formułowanie niezmiennika pętli.	Rozwiązywanie przykładów przy tablicy	2	M_03
TK_17	Specyfikacja i weryfikacja funkcji rekurencyjnych metodą indukcji matematycznej zupełnej.	Rozwiązywanie przykładów przy tablicy	2	M_03
TK_18	Wprowadzenie do środowiska testowego JUnit. Instalacja i konfiguracja oprogramowania. Pierwsze testy. Prezentacja przykładowych programów do przetestowania indywidualnego przez studentów.	Praca indywidualna przy komputerze	4	M_04
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 napotykaných przy testowaniu.	Praca indywidualna przy komputerze	2	M_04
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.	Praca indywidualna przy komputerze	2	M_04
TK_21	Indywidualna realizacja wybranych przypadków testowych i ich wykonanie. opracowanie raportu z wykonanego zadania – miniprojekt.	Praca indywidualna przy komputerze, konsultacje	10	M_04, M_05
TK_22	Prezentacja wyników testowania – dyskusja w grupie. Omówienie cech sporządzonej indywidualnej dokumentacji. Korekta błędów.	Prezentacje multimedialne	6	M_04, M_05
		seminarium		

III. INFORMACJE DODATKOWE

Odniesienie efektów uczenia się przypisanych do zajęć i treści programowych do form zajęć i metod oceniania

Symbol efektu uczenia się przypisanego do zajęć	Formy zajęć i metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć
WIEDZA		
M_01	Wykład podający, prezentacje działających programów, rozwiązywanie zadań przy współudziale studentów	test pisemny
M_02	Wykład podający, prezentacje działających programów, rozwiązywanie zadań przy współudziale studentów	test pisemny
UMIEJĘTNOŚCI		
M_03	Rozwiązywanie zadań przy tablicy	Ocena bieżąca podczas zajęć
M_04	Praca indywidualna	Ocena bieżąca programów
M_05	Praca indywidualna	Ocena sprawozdania i prezentacji
KOMPETENCJE SPOŁECZNE		
M_06	Wykład podający	test pisemny