



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Nazwa przedmiotu		Kod ECTS	
Testowanie aplikacji .NET		11.3.0749	
Nazwa jednostki prowadzącej przedmiot			
Faculty of Mathematics, Physics and Informatics			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	stacjonarne
		moduł	informatyka ogólna, tester programista
		specjalnościowy	
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr Tomasz Borzyszkowski			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		5 Celem zajęć jest przedstawienie studentom różnych aspektów testowania oprogramowania opartego na technologii .Net oraz nabycie umiejętności praktycznych w tym zakresie.	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Ćw. laboratoryjne: 45 godz., Wykład: 15 godz.			
Cykl dydaktyczny			
2017/2018 letni			
Status przedmiotu		Język wykładowy	
- fakultatywny (do wyboru) - obowiązkowy		polski	
Metody dydaktyczne		Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne	
<ul style="list-style-type: none"> - Projektowanie doświadczeń - Wykonywanie doświadczeń - Wykład z prezentacją multimedialną 		Sposób zaliczenia	
		<ul style="list-style-type: none"> - Zaliczenie na ocenę - Egzamin 	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - egzamin ustny - wykonanie pracy zaliczeniowej - projekt lub prezentacja - wykonanie pracy zaliczeniowej - wykonanie określonej pracy praktycznej 	
		Podstawowe kryteria oceny	
		Laboratorium: <ul style="list-style-type: none"> • 100% punkty z realizowanych projektów Egzamin: <ul style="list-style-type: none"> • 50% test wiedzy teoretycznej • 50% ocena realizowanych projektów 	
Sposób weryfikacji założonych efektów kształcenia			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	referat	raport	aktywność w dyskusji	obserwacja postawy studenta
Wiedza							
K_W02	x						
K_W03	x						
K_W04	x						
K_W08	x						
K_W12						x	x
Umiejętności							
K_U01	x		x				
K_U03			x			x	x
K_U04			x			x	x
K_U05			x			x	x
K_U06						x	x
K_U08						x	x
K_U15			x				
K_U16						x	x
K_U17						x	x
K_U18			x			x	x
K_U20			x			x	x

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Zaliczenie z przedmiotów: Wstęp do programowania, Języki programowania oraz Programowanie obiektowe.

B. Wymagania wstępne

Umiejętność konstruowania programów językach C# oraz Java.

Cele kształcenia

Nauczenie studentów podstawowych koncepcji związanych z testowaniem oprogramowania wykonanego przy użyciu technologii firmy Microsoft oraz umiejętności planowania i przeprowadzania testów konkretnych aspektów przykładowych aplikacji.

Treści programowe

1. Idee związane z Test Driven Development.
2. Testy jednostowe i ich tworzenie i automatyzacja w Visual Studio.
3. Tworzenie przypadków testowych i pojęcie pokrycia kodu oraz izolacji środowiska testowego. Wykorzystanie narzędzi Pex i Moles.
4. Testowanie aplikacji wykonanych w technologii ASP MVC.
5. Tworzenie testów wydajnościowych i obciążeniowych w Visual Studio.
6. Zastosowanie TDD w testowaniu aplikacji opartych na wzorcu MVVM.
7. Korzystanie z debugera w Visual Studio:
 - Śledzenie wykonania programu
 - Zapisywanie informacji o wykonaniu programu i jego analiza – wykorzystanie narzędzia IntelliTrace.

Wykaz literatury

1. Verifying Code by Using Unit Tests: <http://msdn.microsoft.com/en-us/library/dd264975.aspx>
2. Debugging in Visual Studio: <http://msdn.microsoft.com/en-us/library/sc65sadd.aspx>
3. Verifying Database Code by Using Unit Tests: [http://msdn.microsoft.com/en-us/library/dd172118\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/dd172118(v=vs.100).aspx)

Efekty kształcenia

(obszarowe i kierunkowe)

K_W02 ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie programowania, algorytmów i złożoności, architektury systemów komputerowych, systemów operacyjnych, technologii sieciowych, języków i paradygmatów programowania, baz danych, inżynierii oprogramowania, języków formalnych,
K_W03: zna podstawowe metody projektowania,

Wiedza

Student zna podstawy planowania, wytwarzania i przeprowadzania testów aplikacji opartych na technologiach firmy Microsoft.

Umiejętności

Student potrafi projektować, konstruować i analizować wyniki testów aplikacji wytworzonych przy użyciu technologii firmy Microsoft.

Kompetencje społeczne (postawy)

Student jest w stanie współpracować z zespołem programistów tworzących

<p>analizowania i programowania algorytmów, K_W04: zna podstawowe konstrukcje programistyczne oraz struktury danych, K_W08: ma ogólną wiedzę na temat różnych paradygmatów programowania i języków programowania (imperatywny, obiektowy, assembler); szczególnie zna metody projektowania i programowania obiektowego, K_W12: zna podstawowe zasady bezpieczeństwa i higieny pracy w zawodzie informatyka K_U01 potrafi zastosować wiedzę matematyczną do formułowania, analizowania i rozwiązywania prostych zadań związanych z informatyką, K_U03 potrafi pracować indywidualnie i w zespole informatyków, w tym także potrafi zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminów, K_U04 potrafi porozumiewać się przy użyciu różnych technik w środowisku zawodowym oraz w innych środowiskach, w tym w języku angielskim oraz z wykorzystaniem narzędzi informatycznych, K_U05 potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym, K_U06 projektuje, analizuje pod kątem poprawności i złożoności obliczeniowej oraz programuje algorytmy; wykorzystuje podstawowe techniki algorytmiczne i struktur danych, K_U08 posługuje się przyjętymi formatami reprezentacji różnego rodzaju danych stosownie do sytuacji, K_U15 Potrafi projektować oprogramowanie zgodnie z metodyką obiektową, potrafi stworzyć model obiektowy prostego systemu, K_U16 ocenia przydatność różnych paradygmatów i związanych z nimi środowisk programistycznych do rozwiązywania różnego typu problemów, K_U17 potrafi ocenić, na podstawowym poziomie, przydatność metod i narzędzi informatycznych, K_U18 potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny, K_U20 jest przygotowany do efektywnego uczestniczenia w inspekcji oprogramowania</p>	rozwiązania obiektowe problemów programistycznych.
---	--

Kontakt

t.borzyszkowski@inf.ug.edu.pl