



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 Ruby		11.3.0782	
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 Włodzimierz Bzyl			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		5	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Ćw. laboratoryjne: 45 godz., Wykład: 15 godz.			
Cykl dydaktyczny			
2017/2018 zimowy			
Status przedmiotu		Język wykładowy	
obowiązkowy		polski	
Metody dydaktyczne		Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne	
Projektowanie doświadczeń		Sposób zaliczenia	
		- Zaliczenie na ocenę - Egzamin	
		Formy zaliczenia	
		ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymanych w trakcie trwania semestru	
		Podstawowe kryteria oceny	
		Umiejętność komunikacji w zespole, adaptacji do zmieniających się wymagań, planowania rozwiązań, systematyczność, dotrzymywanie terminów realizacji przyjętych zobowiązań.	
Sposób weryfikacji założonych efektów kształcenia			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	referat	raport	aktywność w dyskusji	obserwacja i ocena	
Wiedza								
K_W08			x			x	x	
K_W13						x		
Umiejętności								
K_U10			x					
K_U15						x		
Kompetencje								
K_K02			x				x	
K_K03			x			x		

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Brak.

B. Wymagania wstępne

Umiejętność tworzenia prostych skryptów przy użyciu wybranych przez studenta języków skryptowych (np. *Bash*, *Ruby*, *Python*).

Cele kształcenia

K_W08: ma ogólną wiedzę o aktualnych kierunkach rozwoju i nowoczesnych wariantach omawianych modeli opisu wiedzy

K_W13: ma pogłębioną wiedzę na temat aktualnie obowiązujących przepisów prawnych dotyczących działalności informatyka (działalność dydaktyczna, naukowa i zawodowa) oraz własności intelektualnej

K_U10: potrafi pracować zespołowo, również w grupach interdyscyplinarnych; rozumie konieczność systematycznej pracy nad wszelkimi projektami, które mają charakter długofalowy

K_U15: posiada umiejętność przygotowania wystąpień ustnych w zakresie informatyki

K_K02: potrafi pracować zespołowo, rozumie konieczność systematycznej pracy nad projektami, które mają charakter długofalowy; potrafi rozplanować prace w grupie, umie określić priorytety pracy

K_K03: potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego rozumowania danego tematu lub odnalezieniu brakujących elementów rozumowania

Treści programowe

- Wprowadzenie do języka Rubi.
 - Podstawowe elementy języka: pętle, funkcje.
 - Przegląd metod klas *Array* i *Hash*.
 - Piszemy proste skrypty.
- Ogólnie o testowaniu, dostępne narzędzia i technologie.
 - Instalacja potrzebnych narzędzi: Ruby Version Manager (RVM), edytor (Atom/Emacs/Vim).
 - Wprowadzenie do *RSpec* – wykorzystamy z gemów *Capybara* i *Factory Girl*.
- Testy jednostkowe, integracyjne.
 - Różnice między testami.
 - Znaczenie testów.
 - Pokrycie kodu testami.
- Refaktoryzacja kodu.
 - Na czym polega refaktoryzacja kodu?
 - Co to jest *bad smells in code* (w literaturze spotyka się też określenie *smoked code*)?
- Obiekty *doubles* (*mocks*, *stubs*).
 - Fake Objects*.
 - Do czego używamy *doubles*?
 - Główne zasady stosowania *doubles*.
 - Problemy, nadużycia, ograniczenia.
- Praca z *Legacy Code*, czyli z już istniejącym kodem / kodem nie pokrytym testami.
 - Refaktoryzacja *Legacy Code*.

Wykaz literatury

- Dokumentacja *RSpec*, <http://rspec.info/>
- Jay Fields, Shane Harvie, Martin Fowler with Kent Beck. *Refactoring – Ruby edition*, Pearson Education, 2009.
- A community-driven Ruby coding style guide*, <https://github.com/bbatsov/ruby-style-guide>.

Efekty kształcenia

Wiedza

<p>(obszarowe i kierunkowe)</p> <p>Podniesienie kompetencji w zakresie umiejętności wytwarzania oprogramowania dla klienta, którego wymagania zmieniają się w trakcie jego realizacji.</p>	<p>Student: na zasady wytwarzania oprogramowania w metodyce Agile.</p>
	<p>Umiejętności</p> <p>Student: potrafi pracować zespołowo; rozumie konieczność tworzenia testów rozumie konieczność korzystania z rozproszonego systemu kontroli wersji oprogramowania umie znajdować niezbędne informacje w podobnych projektach <i>open source</i></p>
	<p>Kompetencje społeczne (postawy)</p> <p>Student: rozumie potrzebę dalszego kształcenia potrafi pracować zespołowo, potrafi rozplanować swoją pracę, umie określić priorytety pracy</p>
<p>Kontakt</p> <p>matwb@ug.edu.pl</p>	