



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Nazwa przedmiotu		Kod ECTS	
Zaawansowane języki programowania		11.3.0786	
Nazwa jednostki prowadzącej przedmiot			
Instytut Informatyki			
Studia			
wydział	kierunek	poziom	drugiego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	stacjonarne
		moduł	wszystkie
		specjalnościowy	wszystkie
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr Włodzimierz Bzyl; mgr Mateusz Miotk; dr Piotr Arłukowicz			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		7 30 godz wykł + 30 godz lab + praca własna studenta	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Wykład: 30 godz., Ćw. laboratoryjne: 30 godz.			
Termin realizacji przedmiotu			
2019/2020 zimowy			
Status przedmiotu		Język wykładowy	
obowiązkowy		angielski	
Metody dydaktyczne		Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne	
<ul style="list-style-type: none"> - Projektowanie doświadczeń - Wykład problemowy - Wykład z prezentacją multimedialną 		Sposób zaliczenia	
		<ul style="list-style-type: none"> - Zaliczenie na ocenę - Egzamin 	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - wykonanie pracy zaliczeniowej - projekt lub prezentacja - egzamin pisemny z pytaniami (zadaniami) otwartymi - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru 	
		Podstawowe kryteria oceny	
		Projekt indywidualny i projekt zespołowy.	
Sposób weryfikacji założonych efektów kształcenia			
Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi			
A. Wymagania formalne			
Brak.			
B. Wymagania wstępne			
Rozumienie podstawowych konstrukcji programistycznych w językach obiektowych takich jak instrukcje warunkowe, pętle, metody, klasy, dziedziczenie.			
Cele kształcenia			
Celem przedmiotu jest zapoznanie słuchaczy z technikami refaktoryzacji kodu. Refaktoryzacje to techniki ulepszania struktury istniejącego kodu, które minimalizują ryzyko wprowadzenia błędów.			
Stałe refaktoryzacje są konieczne, bo z czasem kod zmienia się i staje się chaotyczny, co powoduje, że kodowanie zmienia się z procesu			

inżynierskiego w hakowanie.	
Treści programowe	
<ul style="list-style-type: none"> • Proces tworzenia aplikacji obiektowych. • Rola testów automatycznych. • Co to jest code smell ? • Katalog refaktoryzacji (przykładowo Extract Method, Feature Envy, Replace Conditional with Polymorphism). • Zasady projektowania SOLID: Single responsibility principle, Open/closed principle, Liskov substitution principle, Interface segregation principle, Dependency inversion principle. 	
Wykaz literatury	
<ul style="list-style-type: none"> • Robert C. Martin, Czysty kod. Podręcznik dobrego programisty. • Jay Fields, Shane Harvie, and Martin Fowler (with Kent Beck), Refactoring – Ruby Edition. • Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts, Erich Gamma, Refaktoryzacja. Ulepszanie struktury istniejącego kodu. 	
Kierunkowe efekty kształcenia	Wiedza
<p>K_W02: ma pogłębioną wiedzę na temat podstawowych paradygmatów programowania; zna również aktualne trendy w językach programowania;</p> <p>K_W12: zna dobrze zasady bezpieczeństwa i higieny pracy w zawodzie informatyka;</p> <p>K_U10: potrafi pracować zespołowo, również w grupach interdyscyplinarnych; rozumie konieczność systematycznej pracy nad wszelkimi projektami, które mają charakter długofalowy;</p> <p>K_U12: potrafi przedstawić wyniki badań w postaci samodzielnie przygotowanej rozprawy (referatu) zawierającej opis i uzasadnienie celu pracy, przyjętą metodologię, wyniki oraz ich znaczenie na tle innych podobnych badań;</p> <p>K_K01: rozumie potrzeby dalszego kształcenia;</p> <p>K_K02: potrafi pracować zespołowo, rozumie konieczność systematycznej pracy nad projektami, które mają charakter długofalowy; potrafi rozplanować prace w grupie, umie określić priorytety pracy;</p>	<ul style="list-style-type: none"> • ma pogłębioną wiedzę na temat podstawowych paradygmatów programowania; zna również aktualne trendy w językach programowania • zna dobrze zasady bezpieczeństwa i higieny pracy w zawodzie informatyka
	Umiejętności
	<ul style="list-style-type: none"> • potrafi pracować zespołowo, również w grupach interdyscyplinarnych; rozumie konieczność systematycznej pracy nad wszelkimi projektami, które mają charakter długofalowy • potrafi przedstawić wyniki badań w postaci samodzielnie przygotowanej rozprawy (referatu) zawierającej opis i uzasadnienie celu pracy, przyjętą metodologię, wyniki oraz ich znaczenie na tle innych podobnych badań
	Kompetencje społeczne (postawy)
	<ul style="list-style-type: none"> • rozumie potrzeby dalszego kształcenia • potrafi pracować zespołowo, rozumie konieczność systematycznej pracy nad projektami, które mają charakter długofalowy; potrafi rozplanować prace w grupie, umie określić priorytety pracy
Kontakt	
w.bzyl@inf.ug.edu.pl	