

**KAPITAŁ LUDZKI**
NARODOWA STRATEGIA SPÓJNOŚCIProjekt 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			
Faculty of Mathematics, Physics and Informatics			
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	
Wykład, Ćw. laboratoryjne		30 godz wykł + 30 godz lab + praca własna studenta	
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Ćw. laboratoryjne: 30 godz., Wykład: 30 godz.			
Cykl dydaktyczny			
2017/2018 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 częściowych otrzymywanych w trakcie trwania semestru 	
		Podstawowe kryteria oceny	
		Projekt indywidualny i projekt zespołowy.	
Sposób weryfikacji założonych efektów kształcenia			

zakładany efekt kształcenia	egzamin	kokokwium	projekt	referat	raport	aktywność w dyskusji	obserwacja i ocena postaw	
Wiedza								
K_W02			x			x		
K_W12						x	x	
Umiejętności								
K_U10			x				x	
K_U12								
Kompetencje								
K_K01			x					
K_K02						x	x	

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

- 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

- 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.

Efekty kształcenia

(obszarowe i kierunkowe)

K_W02: ma pogłębioną wiedzę na temat podstawowych paradygmatów programowania; zna również aktualne trendy w językach programowania;

K_W12: zna dobrze zasady bezpieczeństwa i higieny pracy w zawodzie informatyka;

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

K_U12: potrafi przedstawić wyniki badań w postaci samodzielnie przygotowanej rozprawy (referatu) zawierającej opis i uzasadnienie celu pracy, przyjęta metodologie, wyniki oraz ich znaczenie na tle innych podobnych badań;

K_K01: rozumie potrzeby dalszego kształcenia;

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

Wiedza

- 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

- 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, przyjęta metodologie, wyniki oraz ich znaczenie na tle innych podobnych badań

Kompetencje społeczne (postawy)

- 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

określić priorytety pracy;	
Kontakt	
w.bzyl@inf.ug.edu.pl	