



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez  
Unię Europejską w ramach  
Europejskiego Funduszu  
Społecznego

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



<b>Nazwa przedmiotu</b>		<b>Kod ECTS</b>	
JVM Internals		11.3.0775	
<b>Nazwa jednostki prowadzącej przedmiot</b>			
Instytut Informatyki			
<b>Studia</b>			
<b>wydział</b>	<b>kierunek</b>	<b>poziom</b>	<b>drugiego stopnia</b>
Wydział Matematyki, Fizyki i Informatyki	Informatyka	<b>forma</b>	stacjonarne
		<b>moduł</b>	wszystkie
		<b>specjalnościowy</b>	wszystkie
		<b>specjalizacja</b>	wszystkie
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b>			
dr Jakub Neumann			
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS</b>	
<b>Formy zajęć</b>		6	
Wykład, Ćw. laboratoryjne			
<b>Sposób realizacji zajęć</b>			
zajęcia w sali dydaktycznej			
<b>Liczba godzin</b>			
Wykład: 30 godz., Ćw. laboratoryjne: 30 godz.			
<b>Cykl dydaktyczny</b>			
2016/2017 letni			
<b>Status przedmiotu</b>		<b>Język wykładowy</b>	
fakultatywny (do wyboru)		polski	
<b>Metody dydaktyczne</b>		<b>Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne</b>	
<ul style="list-style-type: none"> <li>- Dyskusja</li> <li>- Projektowanie doświadczeń</li> <li>- Rozwiązywanie zadań</li> <li>- Wykonywanie doświadczeń</li> <li>- Wykład problemowy</li> <li>- Wykład z prezentacją multimedialną</li> </ul>		<b>Sposób zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- Zaliczenie na ocenę</li> <li>- Egzamin</li> </ul>	
		<b>Formy zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- egzamin ustny</li> <li>- ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru</li> <li>- wykonanie pracy zaliczeniowej - przeprowadzenie badań i prezentacja ich wyników</li> <li>- Wykład z kodowaniem na żywo fragmentów programów i eksperymentami</li> </ul>	
		<b>Podstawowe kryteria oceny</b>	
		Głównym kryterium zaliczenia będzie <b>systematyczna</b> praca na zajęciach i praktyczna znajomość i zrozumienie tematów omawianych na wykładach.	
<b>Sposób weryfikacji założonych efektów kształcenia</b>			

zakładany efekt kształcenia	Wykonywanie doświadczeń	Projektowanie doświadczeń	Rozwiązywanie zadań	Wykład z prezentacją multimedialną	Wykład problemowy	Dyskusja
Wiedza						
K_W02				x	x	x
K_W04				x	x	x
K_W05	x		x			
Umiejętności						
K_U02		x	x			
K_U06	x					
K_U07	x		x			
K_U12	x		x			
Kompetencje						
K_K01				x	x	x
K_K03				x	x	x

### Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

#### A. Wymagania formalne

Zaliczony przedmiot z kursu programowania obiektowego w języku Java na ocenę co najmniej dobrą. Wskazane zaliczenie kursu z technologii Java Enterprise Edition.

#### B. Wymagania wstępne

Dobra znajomość języka Java i zasad programowania obiektowego. Wskazana znajomość podstaw tworzenia aplikacji Java Enterprise Edition.

### Cele kształcenia

Celem przedmiotu jest zapoznanie studentów z problematyką szczegółów funkcjonowania aplikacji klasy przemysłowej na platformie Java Virtual Machine. Wśród omawianych zagadnień znajdują się tematy takie jak:

- kompilacja do bytecode
- algorytmy wykorzystywane przez GarbageCollector
- manipulacja bytecode i jego dynamiczne generowanie (instrumentacja)
- serializacja i klonowanie obiektów
- refleksja
- classloadery i ich hierarchia
- analiza zużycia pamięci
- debugowanie aplikacji

Ze względu na autorską tematykę wymagającą dużego doświadczenia praktycznego, przedmiot jest współtworzony i częściowo prowadzony przez doświadczonego specjalistę z firmy IT. Student będzie posiadał umiejętność szczegółowego analizowania działającej aplikacji klasy przemysłowej na platformie JVM, w szczególności pod kątem zużycia pamięci i jej wpływu na wydajność, będzie umiał rozpoznawać problemy związane z ładowaniem klas, generować i manipulować bytecode, debugować (również zdalnie) aplikacje.

### Treści programowe

- Kompilacja programów Java: bytecode, kod maszynowy, JIT, -client vs -server
- Classloader
- Manipulacja bytecode, dynamiczne generowanie kodu: ASM, cglib, JavaAssist
- Organizacja pamięci JVM
- Garbage Collector i analiza jego zachowań i wpływu na wydajność aplikacji
- Refleksje
- Serializacja i klonowanie obiektów
- Działanie debuggera i zdalne debugowanie
- Polecenia javap, jad
- JConsole
- Java Management Extensions
- Narzędzia typu MemoryAnalyzer
- Java 8: lambda wyrażenia oraz defender methods

### Wykaz literatury

- Dokumentacja techniczna języka Java oraz materiały (głównie tutoriale) firmy Oracle
- Opracowania, artykuły, blogi specjalistów dostępne w internecie

<p><b>Efekty kształcenia (obszarowe i kierunkowe)</b></p> <p>K_W02 ma pogłębioną wiedzę na temat podstawowych paradygmatów programowania; zna również aktualne trendy w językach programowania</p> <p>K_W04 zna formalne modele obliczeń a także ich własności i znaczenie w praktycznych zastosowaniach informatycznych, ma wiedzę na temat barier obliczalności i trudności obliczeń</p> <p>K_W05 Zna najważniejsze konstrukcje programistyczne oraz struktury danych</p> <p>K_U02 ma umiejętność projektowania abstrakcyjnych struktur danych i ich wydajnych implementacji</p> <p>K_U06 projektuje, analizuje pod kątem poprawności i złożoności obliczeniowej oraz programuje algorytmy z wykorzystaniem różnych technik programistycznych</p> <p>K_U07 potrafi zastosować znane algorytmy w konkretnych sytuacjach, potrafi efektywnie dobrać rodzaj i sposób wykonania algorytmu w zależności od postawionego problemu</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 potrzebę dalszego kształcenia</p> <p>K_K03 potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego rozumowania danego tematu lub odnalezieniu brakujących elementów rozumowania</p>	<p><b>Wiedza</b></p> <ul style="list-style-type: none"> <li>• K_W02 ma pogłębioną wiedzę na temat podstawowych paradygmatów programowania; zna również aktualne trendy w językach programowania szczególnie w odniesieniu do języków platformy JVM (np. Java, Scala, Groovy)</li> <li>• K_W04 zna formalne modele obliczeń a także ich własności i znaczenie w praktycznych zastosowaniach informatycznych, ma wiedzę na temat barier obliczalności i trudności obliczeń szczególnie w zakresie zagadnień związanych z algorytmami odśmiecania pamięci</li> <li>• K_W05 Zna najważniejsze konstrukcje programistyczne oraz struktury danych języków platformy JVM (np. Java, Scala, Groovy)</li> </ul> <p><b>Umiejętności</b></p> <ul style="list-style-type: none"> <li>• K_U02 ma umiejętność projektowania abstrakcyjnych struktur danych i ich wydajnych implementacji, w szczególności w kontekście budowy i działania Java Virtual Machine oraz algorytmów odśmiecania pamięci</li> <li>• K_U06 projektuje, analizuje pod kątem poprawności i złożoności obliczeniowej oraz programuje algorytmy z wykorzystaniem różnych technik programistycznych, w szczególności z uwzględnieniem zagadnień związanych z funkcjonowaniem Java Virtual Machine</li> <li>• K_U07 potrafi zastosować znane algorytmy w konkretnych sytuacjach, potrafi efektywnie dobrać rodzaj i sposób wykonania algorytmu w zależności od postawionego problemu, w szczególności w kontekście projektowania i przeprowadzenia eksperymentów praktycznych</li> <li>• 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 w szczególności dla zagadnień związanych z wydajnością Java Virtual Machine</li> </ul> <p><b>Kompetencje społeczne (postawy)</b></p> <ul style="list-style-type: none"> <li>• K_K01 rozumie potrzebę dalszego kształcenia, w szczególności zgłębiania zasad funkcjonowania Java Virtual Machine</li> <li>• K_K03 potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego rozumowania danego tematu lub odnalezieniu brakujących elementów rozumowania szczególnie w odniesieniu do zagadnień funkcjonowania Java Virtual Machine. Dyskusja i liczne przykłady kodowania na żywo wspierają i zachęcają Studenta do aktywnego uczestnictwa w zajęciach</li> </ul>
<p><b>Kontakt</b></p> <p><a href="mailto:jakub.neumann@inf.ug.edu.pl">jakub.neumann@inf.ug.edu.pl</a></p>	