



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Nazwa przedmiotu		Kod ECTS	
Programowanie obiektowe		11.3.0739	
Nazwa jednostki prowadzącej przedmiot			
Instytut Informatyki			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	niestacjonarne (zaoczne)
		moduł	wszystkie
		specjalnościowy	wszystkie
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr Tomasz Borzyszkowski; mgr Maciej Dziemiańczuk			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		10 Celem przedmiotu jest zapoznanie studentów z zasadami programowania obiektowego i zdarzeniowego na przykładzie języka Java.	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Ćw. laboratoryjne: 40 godz., Wykład: 40 godz.			
Cykl dydaktyczny			
2016/2017 zimowy			
Status przedmiotu		Język wykładowy	
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"> - wykonanie pracy zaliczeniowej - projekt lub prezentacja - egzamin pisemny z pytaniami (zadaniami) otwartymi - kolokwium 	
		Podstawowe kryteria oceny	
		Ocena z laboratorium:	
		50% punkty z kolokwium	
		50% punkty z części projektowej	
		Egzamin pisemny.	
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	x
K_U05		x	x				
K_U06		x	x			x	x
K_U08						x	x
K_U15		x	x				
K_U16		x	x			x	x
K_U17		x	x			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 i Języki programowania.

B. Wymagania wstępne

Umiejętność konstruowania prostych instrukcji w języku C.

Cele kształcenia

Nauczenie studentów podstawowych koncepcji związanych z programowaniem obiektowym oraz umiejętności konstruowania programów obiektowych w języku Java.

Treści programowe

- Wprowadzenie pojęć podstawowych:** klasa, obiekt, pole, metoda; pola i metody: statyczne, publiczne oraz prywatne w języku Java; inicjalizacja i usuwanie obiektu oraz mechanizm tzw. zbieracza śmieci; przegląd instrukcji sterujących języka Java i ich porównanie z instrukcjami sterującymi języka C.
- Pakiety klas języka Java – ukrywanie implementacji:** struktura pakietu; importowanie pakietów; ustalanie praw dostępu do składowych pakietu; budowa interfejsów i ich implementacja.
- Dziedziczenie i polimorfizm:** dziedziczenie – składnia i zachowanie praw dostępu do dziedziczonych pól i metod; od abstrakcji do konkretności: klasy abstrakcyjne i finalne; porównanie własności pól finalnych i statycznych; porównanie własności metod przeciążonych i polimorficznych; przykłady wywołań funkcji polimorficznych.
- Przegląd klas implementujących typowe struktury danych:** pojęcie statycznych i dynamicznych struktur danych; przegląd własności i udostępnianych operacji na typie Collection języka Java i typach pochodnych (List, Set, BitSet, Maps, Vector, Stack i Hashtable); klasy implementujące typy wyliczeniowe i iteratory; polimorficzne metody umożliwiające sortowanie elementów przechowywanych w kolekcjach.
- Programowanie z wykorzystaniem wyjątków:** przegląd predefiniowanych wyjątków w języku Java; zasady tworzenia nowych wyjątków; zgłaszanie i wyłapywanie sytuacji wyjątkowych.
- Programowanie z wykorzystaniem wątków:** pojęcie wątku, zasobów współdzielonych i sekcji krytycznej; przegląd metod współdzielenia zasobów w języku Java: blokowanie zasobów i problem zakleszczenia wątków oraz kolejki priorytetowe i problem uczciwości w dostępie do zasobów; przykłady wykorzystania wątków języka Java do implementacji klasycznych problemów dostępu do zasobów krytycznych.
- Aplety i aplikacje graficzne:** tworzenie apletów – przegląd możliwości oferowanych przez bibliotekę AWT: elementy graficzne i sposoby ich rozmieszczania w oknie, rodzaje zdarzeń związanych z oknami biblioteki AWT i sposoby ich wyłapywania; tworzenie aplikacji – przegląd możliwości oferowanych przez biblioteki Java Beans i Java Swing.

Wykaz literatury

- Eckel B., Thinking in Java – edycja polska. Wydawnictwo HELION, Warszawa, 2010.
- Horstmann C. S., Cornell G., Core Java 2 - Podstawy. Helion, 2010.

3. Horstmann C. S., Cornell G., Core Java 2 - Techniki zaawansowane. Helion, 2010.

4. Campione M., Walrath K., Java Tutorial. Addison-Wesley, 2000.

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, 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); szczegółowo 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

Wiedza

Student zna podstawy programowania obiektowego oraz podstawowe konstrukcje języka Java.

Umiejętności

Student potrafi konstruować obiektowe rozwiązania prostych problemów programistycznych przy użyciu współczesnych narzędzi wspomagających wytwarzanie oprogramowania.

Kompetencje społeczne (postawy)

Student jest w stanie współpracować z zespołem programistów tworzących rozwiązania obiektowe problemów programistycznych.

Kontakt

t.borzyszkowski@ug.edu.pl