



**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>	
Programowanie obiektowe		11.3.0739	
<b>Nazwa jednostki prowadzącej przedmiot</b>			
Instytut Informatyki			
<b>Studia</b>			
<b>wydział</b>	<b>kierunek</b>	<b>poziom</b>	<b>pierwszego stopnia</b>
Wydział Matematyki, Fizyki i Informatyki	Informatyka	<b>forma</b>	niestacjonarne (zaoczne)
		<b>moduł</b>	wszystkie
		<b>specjalnościowy</b>	wszystkie
		<b>specjalizacja</b>	wszystkie
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b>			
dr Tomasz Borzyszkowski; mgr Maciej Dziemiańczuk			
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS</b>	
<b>Formy zajęć</b>		10 Celem przedmiotu jest zapoznanie studentów z zasadami programowania obiektowego i zdarzeniowego na przykładzie języka Java.	
Wykład, Ćw. laboratoryjne			
<b>Sposób realizacji zajęć</b>			
zajęcia w sali dydaktycznej			
<b>Liczba godzin</b>			
Ćw. laboratoryjne: 40 godz., Wykład: 40 godz.			
<b>Cykl dydaktyczny</b>			
2017/2018 zimowy			
<b>Status przedmiotu</b>		<b>Język wykładowy</b>	
obowiązkowy		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>- Projektowanie doświadczeń</li> <li>- Wykonywanie doświadczeń</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>- wykonanie pracy zaliczeniowej - projekt lub prezentacja</li> <li>- egzamin pisemny z pytaniami (zadaniami) otwartymi</li> <li>- kolokwium</li> </ul>	
		<b>Podstawowe kryteria oceny</b>	
		Ocena z laboratorium:	
		50% punkty z kolokwium	
		50% punkty z części projektowej	
		Egzamin pisemny.	
<b>Sposób weryfikacji założonych efektów kształcenia</b>			

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); szczególnie 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