



**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 obiektowo-funkcyjne		11.3.1988	
<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>	stacjonarne
		<b>moduł</b>	wszystkie
		<b>specjalnościowy</b>	wszystkie
		<b>specjalizacja</b>	wszystkie
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b>			
dr Tomasz Borzyszkowski; dr Wiesław Pawłowski; dr inż. Tomasz Górski			
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS</b>	
<b>Formy zajęć</b>		4 Udział w zajęciach dydaktycznych, objętych planem studiów: 45h Praca własna studenta: 30h RAZEM: 75h	
Wykład, Ćw. laboratoryjne			
<b>Sposób realizacji zajęć</b>			
zajęcia w sali dydaktycznej			
<b>Liczba godzin</b>			
Wykład: 15 godz., Ćw. laboratoryjne: 30 godz.			
<b>Termin realizacji przedmiotu</b>			
2024/2025 letni			
<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>- Dyskusja</li> <li>- Projektowanie doświadczeń</li> <li>- Wykład konwersatoryjny</li> <li>- Wykład z prezentacją multimedialną</li> <li>- ćwiczenia laboratoryjne -- sporządzanie i uruchamianie programów komputerowych</li> </ul>		<b>Sposób zaliczenia</b>	
		Zaliczenie na ocenę	
		<b>Formy zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- zaliczenie ustne</li> <li>- ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru</li> </ul>	
		<b>Podstawowe kryteria oceny</b>	
		próg zaliczeniowy	składowa oceny końcowej
sporządzanie i uruchamianie programów komputerowych		50% łącznie z aktywnością	90%
aktywność na zajęciach		brak	10%
<b>Sposób weryfikacji założonych efektów uczenia się</b>			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	referat	raport	aktywność	obserwacja postawy i umiejętności
Wiedza							
K_W04			X				
Umiejętności							
K_U04			X				X
K_U08			X				X
K_U09			X				X
Kompetencje							
K_K02							X

**Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi**

**A. Wymagania formalne**

Brak wymagań formalnych

**B. Wymagania wstępne**

Brak wymagań wstępnych

**Cele kształcenia**

Podstawowym celem jest zapoznanie studentów z najważniejszymi koncepcjami funkcyjnego podejścia do programowania oraz połączenie ich z mechanizmami strukturalizacji wywodzącymi się z podejścia obiektowego. Pozwala to na poszerzenie i pogłębienie wiedzy już posiadanej przez studentów oraz ułatwia im dokonywanie doboru odpowiednich technik programowania w zależności od charakteru zadania. Projekty realizowane podczas zajęć laboratoryjnych służą praktycznemu ugruntowaniu koncepcji poznanych podczas wykładu

**Treści programowe**

1. 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.
2. Pakiety klas – ukrywanie implementacji: struktura pakietu; importowanie pakietów; ustalanie praw dostępu do składowych pakietu; budowa interfejsów i ich implementacja.
3. 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.
4. Przegląd klas implementujących typowe struktury danych: pojęcie statycznych i dynamicznych struktur danych; przegląd własności i udostępnianych operacji; klasy implementujące typy wyliczeniowe i iteratory; polimorficzne metody umożliwiające sortowanie elementów przechowywanych w kolekcjach.
5. Pojęcie lambda-funkcji i ich wykorzystanie do przetwarzania strumieni.
6. Programowanie z wykorzystaniem wyjątków: przegląd predefiniowanych wyjątków; zasady tworzenia nowych wyjątków; zgłaszanie i wyłapywanie sytuacji wyjątkowych.
7. Programowanie z wykorzystaniem wątków: pojęcie wątku, zasobów współdzielonych i sekcji krytycznej; przegląd metod współdzielenia zasobów: blokowanie zasobów i problem zakleszczenia wątków oraz kolejki priorytetowe i problem uczciwości w dostępie do zasobów; przykłady wykorzystania wątków do implementacji klasycznych problemów dostępu do zasobów krytycznych.
8. Przegląd wybranych wzorców projektowych.

**Wykaz literatury**

A. Literatura wymagana do ostatecznego zaliczenia zajęć (zdania egzaminu):

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

B. Literatura uzupełniająca:

Kierunkowe efekty uczenia się	Wiedza
K_W04 ma uporządkowaną, podbudowaną teoretycznie	ma uporządkowaną, podbudowaną teoretycznie wiedzę w zakresie programowania,

<p>wiedzę w zakresie programowania, algorytmów i złożoności, języków i paradygmatów programowania K_U04 potrafi tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz wzorców projektowych K_U08 ocenia przydatność różnych paradygmatów i narzędzi programistycznych do rozwiązywania problemów różnego typu K_U09 potrafi zgodnie z zadaną specyfikacją zaprojektować oraz zrealizować system informatyczny K_K02 potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania</p>	algorytmów i złożoności, języków i paradygmatów programowania
	<p><b>Umiejętności</b></p> <p>potrafi tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz wzorców projektowych ocenia przydatność różnych paradygmatów i narzędzi programistycznych do rozwiązywania problemów różnego typu potrafi zgodnie z zadaną specyfikacją zaprojektować oraz zrealizować system informatyczny</p>
	<p><b>Kompetencje społeczne (postawy)</b></p> <p>potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania</p>
<p><b>Kontakt</b></p> <p>tomasz.borzyszkowski@ug.edu.pl</p>	