KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Nazwa przedmiotu

Języki programowania II

Kod ECTS

11.3.1995

Nazwa jednostki prowadzącej przedmiot

Instytut Informatyki

Studia

wydział	kierunek		
Wydział Matematyki, Fizyki i Informatyki	Informatyka	poziom	pierwszego stopnia
		forma	stacjonarne
		moduł specjalnościowy	wszystkie
		specjalizacja	wszystkie

Nazwisko osoby prowadzącej (osób prowadzących)

prof. UG, dr Jakub Neumann; mgr Aleksandra Tejszerska

Formy zajęć, sposób ich realizacji i przypisana im liczba godzin

Formy zajęć

Wykład, Ćw. laboratoryjne

Sposób realizacji zajęć

zajęcia w sali dydaktycznej

Liczba godzin

Wykład: 30 godz., Ćw. laboratoryjne: 30 godz.

Liczba punktów ECTS

4

Udział w zajęciach dydaktycznych, objętych planem studiów: 60h

Praca własna studenta: 40h

RAZEM: 100h

Termin realizacji przedmiotu

2023/2024 letni

Status przedmiotu

obowiązkowy

Język wykładowy

polski

Metody dydaktyczne

- Projektowanie doświadczeń
- Sporządzanie i uruchamianie programów komputerowych
- Wykład problemowy

Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne

Sposób zaliczenia

Zaliczenie na ocenę

Formy zaliczenia

- ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru
- kolokwium

Podstawowe kryteria oceny

	próg zaliczeniowy	składowa oceny końcowej
kolokwium x 2	60%	90% (po 45% za każde kolokwium)
wejściówki	30%	10%

Sposób weryfikacji założonych efektów uczenia się

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

Celem przedmiotu jest nauka i doskonalenie technik programowania, ze szczególnym uwzględnieniem technik programowania funkcyjnego i obiektowego. Praca z kodem asynchronicznym.

Treści programowe

- techniki programowania funkcyjnego: funkcje wyższego rzędu, funkcje zwracające funkcje, wyrażenia lambda, currying
- definicje funkcji reduce, map, filter, składanie funkcji
- tworzenie obiektów
- definiowanie klas, dziedziczenie
- asynchroniczność i typy danych związane z asynchronicznością
- definiowanie własnych typów danych, kontrola typów, hierarchia, interfejsy

Wykaz literatury

Kierunkowe efekty uczenia się

K_W04 ma uporządkowaną, podbudowaną teoretycznie 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

Wiedza

Student ma wiedzę o podstawach programowania obiektowego i funkcyjnego. Zna podstawy programowania obiektowego: wie na czym polega hermetyzacja danych i dziedziczenie. Wie czym są funkcje wyższego rzedu, w szczególności rozumie operacje reduce, map, filter, wie na czym polega składanie funkcji oraz zna pojęcie curry w odniesieniu do funkcji. Umie definiować własne typy danych, klasy. Wie na czym polega hermetyzacja danych i dziedziczenie. Rozumie zasady wnioskowania i pracy z kodem asynchronicznym.

Umiejętności

Student potrafi pisać programy z wykorzystaniem zasad programowania obiektowego i funkcyjnego, w szczególności tworzyć obiekty, definiować klasy, stosować dziedziczenie. Umie stosować funkcje wyższego rzędu w szczególności reduce, map, filter. Potrafi stosować operację składania funkcji. 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. Umie definiować własne klasy. Umie pracować z kodem asynchronicznym.

Kompetencje społeczne (postawy)

potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania

Kontakt

jakub.neumann@ug.edu.pl