


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 Projekt współfinansowany przez
 Unię Europejską w ramach
 Europejskiego Funduszu
 Społecznego

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Nazwa przedmiotu		Kod ECTS	
Wstęp do programowania (OA)		11.3.2069	
Nazwa jednostki prowadzącej przedmiot			
Instytut Informatyki			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	stacjonarne
		moduł	wszystkie
		specjalnościowy specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr Janusz Dybizbański; dr inż. Anna Nenca; mgr Konrad Sołtys			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		6	
Wykład, Ćw. laboratoryjne		Przedmiot w wymiarze 30h wykładu, 45h ćw. laboratoryjnych + praca własna studenta 75h	
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin		Razem 150h	
Wykład: 30 godz., Ćw. laboratoryjne: 45 godz.			
Termin realizacji przedmiotu			
2023/2024 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"> - Wykład z prezentacją multimedialną - ćwiczenia laboratoryjne -- sporządzanie i uruchamianie programów komputerowych 		Sposób zaliczenia	
		Egzamin	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - wykonanie pracy zaliczeniowej - projekt lub prezentacja - egzamin pisemny z pytaniami (zadaniami) otwartymi - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru - kolokwium 	
		Podstawowe kryteria oceny	
		Ocena końcowa wyznaczana jest na podstawie poniższych kryteriów:	
		próg zaliczeniowy	składowa oceny końcowej
	egzamin	50%	50%
	kolokwium	50%	20%
	projekt	50%	10%
	projekty realizowane na zajęciach	0%	15%
	aktywność na zajęciach	0%	5%
Sposób weryfikacji założonych efektów uczenia się			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	sprawdzian	referat	raport	aktywność w dyskusji	obserwacja postawy
Wiedza								
K_W05	X	X	X					
K_W10								X
P_W1	X	X	X					
P_W2	X	X	X					
P_W3								X
Umiejętności								
K_U02							X	X
K_U06		X						X
P_U1		X					X	X
P_U2							X	X
P_U3		X					X	X
Kompetencje								
K_K01							X	X
K_K03							X	X
P_K1							X	X
P_K2							X	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 poznanie podstawowych technik programistycznych oraz zdobycie umiejętności projektowania, analizy i implementacji podstawowych algorytmów.

Treści programowe

- Algorytm a program. Proste programy w różnych sposobach zapisu (opis słowny, schemat blokowy, instrukcje języka programowania). Ręczna symulacja działania algorytmu.
- Schemat przetwarzania programu w języku wysokiego poziomu - etapy kompilacji, konsolidacji i wykonania.
- Deklaracje zmiennych. Proste typy danych oraz typy strukturalne: tablica, rekord. Zakres deklaracji i widoczności zmiennych, zmienne globalne.
- Instrukcje sterujące. Pojęcie poprawności częściowej i całkowitej programu.
- Zagnieżdżone pętle.
- Procedury i funkcje. Sposoby przekazywania parametrów do funkcji.
- Mechanizm rekursji i jego wykorzystanie. Wyrażanie pętli przez rekursję i przykład wyrażenia rekursji pętłą. Dowodzenie poprawności funkcji rekurencyjnych za pomocą indukcji matematycznej.
- Reprezentacja liczb całkowitych i zmiennoprzecinkowych w komputerze oraz błędy nimi spowodowane (przekroczenie zakresu i błędy zaokrąglenia).
- Wykorzystanie wskaźników i zarządzanie pamięcią na przykładzie prostych struktur danych.
- Informacja o teoretycznych ograniczeniach obliczeniowych.

Wykaz literatury

- [1] Griffiths David, Griffiths Dawn. *C. Rusz głową!* Wydawnictwo Helion.
- [2] Szepietowski A. *Podstawy informatyki*. Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 2000.
- [3] Kernighan B.W., Ritchie D.M. *Język ANSI C*. Wydawnictwa Naukowo-Techniczne, Warszawa 2003.

Kierunkowe efekty uczenia się

K_W05: ma ogólną wiedzę na temat różnych paradygmatów programowania i języków programowania;

Wiedza

Student:
- zna podstawowe narzędzia programistyczne

<p>szczegółowo zna metody i wzorce projektowania i programowania obiektowego</p> <p>K_W10: zna podstawowe zasady bezpieczeństwa i higieny pracy w zawodzie informatyka</p> <p>K_U02: potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania</p> <p>K_U06: potrafi projektować, tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz adekwatnych wzorców</p> <p>K_K01: zna ograniczenia własnej wiedzy i rozumie potrzebę dalszego uczenia się</p> <p>K_K03: potrafi i jest gotów formułować opinie na temat podstawowych zagadnień informatycznych</p>	<ul style="list-style-type: none"> - zna elementy programów komputerowych: zmienne, instrukcje proste i sterujące, funkcje - ma podstawy wiedzy o sposobach analizy poprawności i złożoności algorytmów - zna typowe algorytmy wyszukiwania i porządkowania - zna mechanizm rekursji - zna sposoby reprezentacji liczb w komputerze - zna podstawy technik programistycznych: dziel i rządź, programowanie dynamiczne - zna podstawowe struktury danych: stos, kolejka <p>Efekty przedmiotowe:</p> <p>P_W1 student umie tworzyć programy komputerowe i zna podstawowe techniki projektowania algorytmów (K_W05)</p> <p>P_W2 student zna mechanizm rekursji oraz podstawowe algorytmy i struktury danych (K_W05)</p> <p>P_W3 student zna podstawowe zasady korzystania z sieci i stanowisk komputerowych (K_W10)</p>
	<p>Umiejętności</p> <p>Student:</p> <ul style="list-style-type: none"> - umie przeprowadzić ręczną symulację działania prostego programu z pętlami lub rekursją - umie zaprojektować i zaimplementować program z pętlami (również zagnieżdżonymi) oraz funkcjami - umie uzasadnić, że napisany program działa zgodnie z (nieformalnymi) wymaganiami - potrafi ocenić liczbę operacji, wykonywanych przez program z pętlami - umie zaprojektować proste funkcje rekurencyjne - potrafi zapisywać i odczytywać liczby jako ciągi bitów, zarówno w przypadku stałojak i zmiennopozycyjnym <p>Efekty przedmiotowe:</p> <p>P_U1 student umie wyszukiwać i poprawiać błędy w istniejących programach komputerowych (K_U02, K_U06)</p> <p>P_U2 student umie ocenić i uzasadnić optymalność rozwiązań prostych problemów (K_U02)</p> <p>P_U3 student umie projektować proste algorytmy, w tym korzystające z techniki DP oraz 'dziel i zwyciężaj' (K_U06)</p>
	<p>Kompetencje społeczne (postawy)</p> <p>Student:</p> <ul style="list-style-type: none"> - potrafi formułować wymagania dotyczące algorytmów - rozumie konieczność dalszego kształcenia się <p>Efekty przedmiotowe:</p> <p>P_K1: student umie formułować opinie na temat podstawowych algorytmów i ograniczeń języków programowania (K_K01)</p> <p>P_K2: student rozumie konieczność rozwijania swojej wiedzy (K_K03)</p>
<p>Kontakt</p> <p>jdybiz@inf.ug.edu.pl</p>	