



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Nazwa przedmiotu		Kod ECTS	
Algorytmy i struktury danych		11.3.0780	
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 Paweł Pączkowski			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		10 Przedmiot w wymiarze 30h wykładu i 30h ćw. aud. + praca własna studenta	
Wykład, Ćw. audytoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Ćw. audytoryjne: 30 godz., Wykład: 30 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	
Rozwiązywanie zadań		Sposób zaliczenia	
		- Zaliczenie na ocenę - Egzamin	
		Formy zaliczenia	
		- egzamin pisemny z pytaniami (zadaniami) otwartymi - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru - kolokwium	
		Podstawowe kryteria oceny	
		Ocena z ćwiczeń - na podstawie wyników kolokwium i oceny napisanego programu Ocena z wykładu - wynik egzaminu pisemnego	
Sposób weryfikacji założonych efektów kształcenia			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	sprawdzian	referat	raport	aktywność w dyskusji	obserwacja
Wiedza								
K_W02	X	X						
K_W03	X							
K_Wo4	X							
Umiejętności								
K_U01	X							
K_U03								X
K_U04							X	
K_U05			X					
K_U06			X					
K_U08	X	X						
K_U16								X
K_U17			X					
K_U18			X					
K_U20							X	

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Matematyka Dyskretna, Języki Programowania

B. Wymagania wstępne

Umiejętność programowania, znajomość aparatu matematycznego na poziomie wykładu Matematyka Dyskretna

Cele kształcenia

Zapoznanie studentów z klasycznymi algorytmami i strukturami danych używanymi do efektywnego rozwiązania typowych zadań programistycznych, uzasadnienie poprawności poznawanych algorytmów i przeprowadzenie analizy złożoności czasowej tych algorytmów

Treści programowe

- Pojęcia wstępne: poprawność semantyczna, złożoność czasowa pesymistyczna i oczekiwana, notacja asymptotyczna złożoności
- Sortowanie przez porównania. Algorytmy o złożoności kwadratowej, o złożoności liniowo-logarytmicznej (heapsort), o średniej złożoności liniowo-logarytmicznej (quicksort). Twierdzenia o ograniczeniach dolnych złożoności czasowej pesymistycznej i oczekiwanej.
- Sortowanie w czasie liniowym.
- Podstawowe struktury danych: listy, stosy, kolejki, kolejki priorytetowe. Implementacje przy użyciu tablic i struktur dwojganiowych.
- Struktury danych dla operacji słownikowych (wstaw, usuń, szukaj): tablice z haszowaniem, drzewa poszukiwań binarnych, drzewa zrównoważone, B-drzewa.
- Metody konstruowania efektywnych algorytmów: metoda "dziel i zwyciężaj", programowanie dynamiczne (najdłuższy wspólny podciąg), algorytmy zachłanne (kody Huffmana).
- Przykłady algorytmów grafowych (najkrótsze ścieżki w grafie).

Wykaz literatury

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein -- Wprowadzenie do algorytmów, Wydawnictwo Naukowe PWN 2012.
- L. Banachowski, K. Diks, W. Rytter -- Algorytmy i struktury danych, WNT 2011.

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

Wiedza

Student:

- ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie klasycznych algorytmów sortowania oraz realizujących operacje słownikowe (wstaw, usuń, szukaj) i ich złożoności czasowej
- zna przykłady algorytmów reprezentujących podstawowe metody tworzenia efektywnych algorytmów: dziel i zwyciężaj, programowanie dynamiczne, strategia zachłanna
- ma podstawową wiedzę na temat analizowania poprawności i złożoności algorytmów
- zna klasyczne struktury danych: stosy, listy, kolejki, kopce, drzewa, tablice z haszowaniem

<p>K_U01 potrafi zastosować wiedzę matematyczną do formułowania, analizowania i rozwiązywania prostych zadań związanych z informatyką</p> <p>K_U03 potrafi pracować indywidualnie i w zespole informatyków, w tym także potrafi zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminów</p> <p>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</p> <p>K_U05 potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym</p> <p>K_U06 projektuje, analizuje pod kątem poprawności i złożoności obliczeniowej oraz programuje algorytmy; wykorzystuje podstawowe techniki algorytmiczne i struktur danych</p> <p>K_U08 posługuje się przyjętymi formatami reprezentacji różnego rodzaju danych stosownie do sytuacji</p> <p>K_U16 ocenia przydatność różnych paradygmatów i związanych z nimi środowisk programistycznych do rozwiązywania różnego typu problemów</p> <p>K_U17 potrafi ocenić, na podstawowym poziomie, przydatność metod i narzędzi informatycznych</p> <p>K_U18 potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny</p> <p>K_U20 jest przygotowany do efektywnego uczestniczenia w inspekcji oprogramowania</p>	<p>Umiejętności</p> <p>Student:</p> <ul style="list-style-type: none"> • potrafi wyjaśnić, posługując się przykładem, działanie wybranych klasycznych algorytmów • potrafi podać definicje wybranych klasycznych, powszechnie używanych struktur danych i zilustrować je przykładem (stosy, kolejki, kopce, drzewa, tablice z haszowaniem) • potrafi podać przykłady algorytmów o różnej czasowej złożoności obliczeniowej i potrafi ocenić złożoność czasową prostego algorytmu
	<p>Kompetencje społeczne (postawy)</p>
<p>Kontakt</p> <p>Pawel.Paczkowski@inf.ug.edu.pl</p>	