



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 I		11.3.0785	
Nazwa jednostki prowadzącej przedmiot			
Instytut Informatyki			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	stacjonarne
		moduł	wszystkie
		specjalnościowy	wszystkie
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr Paweł Pączkowski; mgr Radosław Ziemann; Cezary Walczak; prof. UG, dr hab. Paweł Żyliński; mgr Maciej Dziemiańczuk			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		5 Przedmiot w wymiarze 30h wykładu i 30h ćw. lab. + praca własna studenta	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Ćw. laboratoryjne: 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	
ćwiczenia laboratoryjne - programowanie		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	
		Podstawowe kryteria oceny	
		Ocena z ćwiczeń - na podstawie napisanych programów i wyników sprawdzianów 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			X				
K_W04			X					
K_W12								X
Umiejętności								
K_U01	X							
K_U03								X
K_U04							X	
K_U05			X					
K_U06			X					
K_U08			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, sposobami implementacji poznawanych algorytmów, analizą złożoności czasowej tych algorytmów i uzasadnieniem ich poprawności

Treści programowe

- Pojęcia wstępne: poprawność semantyczna, złożoność czasowa pesymistyczna i oczekiwana, notacja asymptotyczna
- 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 dowiązaniowych.
- Struktury danych dla operacji słownikowych (wstaw, usuń, szukaj): tablice z haszowaniem, drzewa poszukiwań binarnych.
- Analiza kosztu zamortyzowanego

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, K_W12: zna podstawowe zasady bezpieczeństwa i higieny pracy w zawodzie informatyka
K_U01 potrafi zastosować wiedzę matematyczną do

Wiedza

Student, który zaliczy przedmiot:

- ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie klasycznych algorytmów sortowania oraz algorytmów realizujących operacje słownikowe (wstaw, usuń, szukaj) i ich złożoności czasowej
- ma podstawową wiedzę na temat analizowania poprawności i złożoności algorytmów
- zna klasyczne struktury danych: stosy, listy, kolejki, kopce, drzewa, tablice z haszowaniem
- ma doświadczenie w programowaniu poznanych algorytmów

Umiejętności

Student, który zaliczy przedmiot:

- potrafi wyjaśnić, posługując się przykładem, działanie wybranych klasycznych algorytmów

<p>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_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</p>	<ul style="list-style-type: none"> • potrafi podać definicje wybranych 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 • potrafi zaprogramować poznane algorytmy posługując się ich opisem w postaci pseudokodu
<p>Kompetencje społeczne (postawy)</p> <p>Student potrafi operować pojęciami w zakresie algorytmów i struktur danych w sposób umożliwiający mu porozumienie się z innymi informatykami w tym zakresie</p>	
<p>Kontakt</p> <p>Pawel.Paczkowski@inf.ug.edu.pl</p>	