

OPIS MODUŁU KSZTAŁCENIA (przedmiot lub grupa przedmiotów)

Nazwa modułu/ przedmiotu Algorytmy i metody programowania			Przedmiot/y Algorytmy i struktury danych Metody programowania Języki i paradygmaty programowania		
Nazwa jednostki prowadzącej przedmiot Instytut Matematyki					
kierunek	specjalność	specjalizacja	semestr/y	poziom kształcenia/ profil kształcenia	forma studiów
Informatyka	Programowanie	-	1,2,4	SPS/praktyczny	stacjonarne/ niestacjonarne
Nazwisko osoby prowadzącej (osób prowadzących) dr inż. Zbigniew Ledóchowski, dr Piotr Sulewski					
Formy zajęć	Liczba godzin				Liczba punktów ECTS
	N (nauczyciel)		S (student)		
	studia stacjonarne	studia niestacjonarne	studia stacjonarne	studia niestacjonarne	
Algorytmy i struktury danych	50	30	75	95	5
(W)wykład	20	12	30	38	2
Przygotowanie do zaliczenia z oceną					
Studiowanie literatury					
(CL) ćwiczenia laboratoryjne	30	18	45	57	3
Przygotowanie do zajęć (w tym rozwiązywanie zadań domowych)					
Przygotowanie do kolokwium					
Metody programowania	45	27	45	63	3
(W)wykład	15	9	15	21	1
Przygotowanie do zaliczenia z oceną					
Studiowanie literatury					
(CL) ćwiczenia laboratoryjne	30	18	30	42	2
Przygotowanie do zajęć (w tym rozwiązywanie zadań domowych)					
Przygotowanie do kolokwium					
Języki i paradygmaty programowania	30	18	75	87	4
(W)wykład	10	6	20	24	1
Przygotowanie do zaliczenia z oceną					
Studiowanie literatury					
(CL) ćwiczenia laboratoryjne	20	12	55	63	3
Przygotowanie do zajęć (w tym rozwiązywanie zadań domowych)					
Przygotowanie do kolokwium					
Razem	125	75	195	245	12
Metody dydaktyczne					
<ul style="list-style-type: none"> (W)wykład: wykład problemowy wspomagany pokazem multimedialnym (CL) ćwiczenia laboratoryjne: ćwiczenia praktyczne - rozwiązywanie zadań z wykorzystaniem komputera, metoda problemowa 					

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne:

Matematyka dyskretna I, Logika i teoria mnogości

B. Wymagania wstępne: wiadomości i umiejętności z przedmiotów informatycznych na poziomie szkoły ponadgimnazjalnej

Cele modułu

- przekazanie podstawowych pojęć związanych z algorytmami oraz ich własnościami
- przekazanie wiedzy związanej z klasami złożoności obliczeniowej oraz kształtowanie umiejętności oceny złożoności obliczeniowej wybranych algorytmów, a także umiejętności optymalizowania algorytmów
- przekazanie wiedzy na temat najważniejszych struktur danych, w tym struktur dynamicznych oraz na temat ich implementacji w wybranym języku programowania
- przekazanie wiedzy na temat teoretycznych podstaw dla algorytmów i języków programowania
- dokonanie analizy algorytmów rozwiązania klasycznych problemów z wykorzystaniem poznanych struktur danych
- zapoznanie z najważniejszymi metodami rozwiązywania problemów algorytmicznych
- wytworzenie umiejętności optymalnego doboru struktur danych oraz metod programowania do rozwiązywanych problemów
- przekazanie wiedzy dotyczącej najważniejszych paradygmatów programowania
- wytworzenie umiejętności praktycznego wykorzystania paradygmatów programowania w rozwiązywaniu różnych problemów algorytmicznych

Algorytmy i struktury danych

1. Podstawowe pojęcia dotyczące algorytmów. Notacje algorytmów.
2. Typy algorytmów.
3. Własności algorytmów i sposoby ich analizy. Poprawność i złożoność obliczeniowa algorytmów. Analiza przykładowych algorytmów pod kątem ich złożoności obliczeniowej (sito Eratostenesa, rozkład liczby na czynniki pierwsze itp.) Klasy złożoności obliczeniowej.
4. Proste i złożone struktury danych. Struktury abstrakcyjne i odniesione do reprezentacji i implementacji.
5. Przegląd struktur danych. Tablice, wektory, stos.
6. Typ wskaźnikowy. Struktury dynamiczne. Listy, kolejki.
7. Przegląd algorytmów operujących na wybranych strukturach danych
 - a) algorytmy sortowania (sortowanie przez wstawianie, przez scalanie, sortowanie szybkie) i ocena ich złożoności
 - b) algorytm sortowania przez kopcowanie -wykorzystanie kopca binarnego
 - c) algorytm selekcji -algorytm Hoare'a
 - d) algorytm wyszukiwania liniowego i binarnego
 - e) algorytmy przetwarzające teksty-prefikso-sufiksy, algorytmy wyszukiwania wzorca (algorytm KMP)-wykorzystanie tekstowych struktur danych
8. Drzewa poszukiwań binarnych i ich zastosowania.
9. Struktury grafowe . Przegląd najważniejszych algorytmów grafowych. Algorytmy BFS i DFS. Algorytmy Dijkstry oraz Bellmana-Forda.

Metody programowania

1. Algorytmy rekurencyjne. Poprawność algorytmów rekurencyjnych. Rekurencja w programowaniu grafiki. Rekurencja, a iteracja.
2. Algorytmy z nawrotami – przeszukiwanie przestrzeni stanów, przycinanie rekurencji. Problem sumy podzbioru, problemy na szachownicy.
3. Metoda dziel i zwyciężaj i przykłady jej zastosowania:
 - a) wyszukiwanie binarne i interpolacyjne
 - b) binarne umieszczanie
 - c) sortowanie szybkie
 - d) problem znajdowania największego i najmniejszego elementu w zbiorze
4. Algorytmy zachłanne. Teoretyczne podstawy strategii zachłannej.
 - a) problem pakowania plecaka
 - b) problem wydawania reszty
 - c) algorytm Huffmana
 - d) analiza problemu szeregowania zadań

5. Podstawy programowania dynamicznego. Zasada optymalności Bellmana. Algorytmy wykorzystujące programowanie dynamiczne:
 - a) problem pakowania plecaka i wydawania reszty-porównanie ze strategią zachłanną
 - b) najdłuższy wspólny podciąg
 - c) mnożenie ciągu macierzy
6. Zastosowanie atrap i strażników w algorytmach wykorzystujących dynamiczne struktury danych.

Języki i paradygmaty programowania

1. Elementy teorii automatów i języków formalnych. Alfabet, słowo, składnia i semantyka. Gramatyki bezkontekstowe jako narzędzie definiowania składni języka. Notacja Chomskiego.
2. Wprowadzenie do paradygmatów programowania. Podprogramy, przeciążanie parametrów. Polimorfizm.
3. Programowanie imperatywne – zmienne, struktura blokowa, wiązania statyczne i dynamiczne, organizacja wywołań podprogramów, przydział pamięci dla programów.
4. Programowanie obiektowe – klasy jako abstrakcyjne typy danych, dziedziczenie klas, szablony i klasy rodzajowe, przykłady z różnych obiektowych języków programowania
5. Programowanie funkcyjne – funkcje jako model programowania, nadawanie typów, dopasowanie wzorca, funkcje wyższego rzędu, pakiety i moduły.
6. Elementy programowania w logice.

Efekty kształcenia

Wiedza

W_01 zna pojęcia dotyczące podstaw teoretycznych teorii algorytmów i języków programowania.

W_02 charakteryzuje klasy złożoności obliczeniowej algorytmów

W_03 opisuje różne struktury danych, w tym dynamiczne struktury danych

W_04 zna klasyczne algorytmy wykorzystujące różne struktury danych

W_05 opisuje i porównuje metody programowania

W_06 charakteryzuje paradygmaty programowania

Umiejętności

U_01 szacuje złożoność obliczeniową przykładowych algorytmów

U_02 projektuje rozwiązania problemów w postaci algorytmów o jak najmniejszej złożoności obliczeniowej.

U_03 konstruuje dowody poprawności wybranych algorytmów

U_04 implementuje różne struktury danych w wybranym języku programowania

U_05 rozwiązuje problemy algorytmiczne dobierając optymalne dla rozwiązania struktury danych i metody programowania

U_06 wykorzystuje przy rozwiązywaniu problemów algorytmicznych gotowe rozwiązania (algorytmy klasyczne)

U_07 stosuje przy implementacji algorytmów poznane paradygmaty programowania

Kompetencje społeczne

K_01 wykazuje kreatywność przy rozwiązywaniu problemów

K_02 pracuje w zespole

K_03 ma świadomość ograniczeń wynikających z niedostatecznej

Sposób zaliczenia oraz formy i podstawowe kryteria oceny/wymagania egzaminacyjne

A. Sposób zaliczenia

a) Przedmiot Algorytmy i struktury danych

W – zaliczenie z oceną

CL – zaliczenie z oceną

b) Przedmiot Metody programowania

W – zaliczenie z oceną

CL – zaliczenie z oceną

c) Przedmiot Języki i paradygmaty programowania

W – zaliczenie z oceną

CL – zaliczenie z oceną

B. Sposoby weryfikacji i oceny efektów

a) Przedmiot Algorytmy i struktury danych

(W)Wykład –test końcowy – efekty: W_01, W_02,W_03,W_04, K_03

Oceną zaliczenia wykładu jest ocena uzyskana za test końcowy -ocena A.

Maksymalnie w teście można uzyskać x punktów, a ocena A jest ustalana na podstawie następujących kryteriów.

wiedzy oraz umiejętności i rozumie potrzebę dalszego kształcenia

A ∈ [0% x, 50% x)	niedostateczna
A ∈ [50% x, 60% x)	dostateczna
A ∈ [60% x, 70% x)	dostateczna plus
A ∈ [70% x, 80% x)	dobra
A ∈ [80% x, 90% x)	db plus
A ∈ [90% x, 100% x]	bardzo dobra

(CL) Ćwiczenia laboratoryjne

- kolokwium nr 1 - efekty: U_01, U_02, U_03 (30%)

- kolokwium nr 2 - efekty: U_04, U_05, U_06, K_01 (70%)

Każda z form oceny CL jest punktowana, a suma punktów możliwych do uzyskania to y.

W nawiasach przy każdej z form oceniania CL podano jaki procent y można uzyskać maksymalnie z tej formy oceniania. Ocena B dla zaliczenia CL jest ustalana na podstawie uzyskanej przez studenta sumy punktów P według zasady:

P ∈ [0% y, 50% y)	niedostateczna
P ∈ [50% y, 60% y)	dostateczna
P ∈ [60% y, 70% y)	dostateczna plus
P ∈ [70% y, 80% y)	dobra
P ∈ [80% y, 90% y)	db plus
P ∈ [90% y, 100% y]	bardzo dobra

Końcowa ocena z zaliczenia przedmiotu jest wyliczona jako średnia ważona ocen otrzymanych za wykład (A) i ćwiczenia laboratoryjne (B), dla których wagami są przypisane im liczby punktów ECTS.

b) Przedmiot Metody programowania

(W) Wykład – test końcowy – efekty: W_05, K_01

Oceną zaliczenia W jest ocena uzyskana za test końcowy - ocena A.

Maksymalnie w teście można uzyskać x punktów, a ocena A jest ustalana na podstawie następujących kryteriów.

A ∈ [0% x, 50% x)	niedostateczna
A ∈ [50% x, 60% x)	dostateczna
A ∈ [60% x, 70% x)	dostateczna plus
A ∈ [70% x, 80% x)	dobra
A ∈ [80% x, 90% x)	db plus
A ∈ [90% x, 100% x]	bardzo dobra

(CL) Ćwiczenia laboratoryjne

- kolokwium nr 1 - efekty: U_05 (50%)

- kolokwium nr 2 - efekty: U_05, U_06, K_01 (50%)

Każda z form oceny CL jest punktowana, a suma punktów możliwych do uzyskania to y.

W nawiasach przy każdej z form oceniania CL podano jaki procent y można uzyskać maksymalnie z tej formy oceniania. Ocena B dla zaliczenia CL jest ustalana na podstawie uzyskanej przez studenta sumy punktów P według zasady:

$P \in [0\% y, 50\% y)$	niedostateczna
$P \in [50\% y, 60\% y)$	dostateczna
$P \in [60\% y, 70\% y)$	dostateczna plus
$P \in [70\% y, 80\% y)$	dobra
$P \in [80\% y, 90\% y)$	db plus
$P \in [90\% y, 100\% y]$	bardzo dobra

Końcowa ocena z zaliczenia przedmiotu jest wyliczona jako średnia ważona ocen otrzymanych za wykład (A) i ćwiczenia laboratoryjne (B), dla których wagami są przypisane im liczby punktów ECTS

c) Przedmiot Języki i paradygmaty programowania

(W) Wykład – test końcowy – efekty: W_06, K_01

Oceną zaliczenia W jest ocena uzyskana za test końcowy - ocena A.

Maksymalnie w teście można uzyskać x punktów, a ocena A jest ustalana na podstawie następujących kryteriów.

$A \in [0\% x, 50\% x)$	niedostateczna
$A \in [50\% x, 60\% x)$	dostateczna
$A \in [60\% x, 70\% x)$	dostateczna plus
$A \in [70\% x, 80\% x)$	dobra
$A \in [80\% x, 90\% x)$	db plus
$A \in [90\% x, 100\% x]$	bardzo dobra

(CL) Ćwiczenia laboratoryjne

- kolokwium - efekty: U_06, U_07 (30%)

- projekt zespołowy realizowany w domu – efekty: U_06, U_07, K_02, K_03 (70%)

Każda z form oceny CL jest punktowana, a suma punktów możliwych do uzyskania to y .

W nawiasach przy każdej z form oceniania CL podano jaki procent y można uzyskać maksymalnie z tej formy oceniania. Ocena B dla zaliczenia CL jest ustalana na podstawie uzyskanej przez studenta sumy punktów P według zasady:

$P \in [0\% y, 50\% y)$	niedostateczna
$P \in [50\% y, 60\% y)$	dostateczna
$P \in [60\% y, 70\% y)$	dostateczna plus
$P \in [70\% y, 80\% y)$	dobra
$P \in [80\% y, 90\% y)$	db plus
$P \in [90\% y, 100\% y]$	bardzo dobra

Końcowa ocena z zaliczenia przedmiotu jest wyliczona

jako średnia ważona ocen otrzymanych za wykład (A) i ćwiczenia laboratoryjne (B), dla których wagami są przypisane im liczby punktów ECTS

Ocena modułu jest średnią ważoną z ocen końcowych dla wszystkich przedmiotów. Wagami jest łączna liczba punktów ECTS dla każdego przedmiotu.

Matryca efektów kształcenia dla przedmiotu

Numer (symbol) efektu kształcenia	Odniesienie do efektów kształcenia dla programu	Odniesienie do charakterystyki drugiego stopnia PRK dla obszaru/ obszarów
W_01	K1_W17	P6S_WG
W_02	K1_W17	P6S_WG
W_03	K1_W20	P6S_WG
W_04	K1_W17, K1_W20	P6S_WG
W_05	K1_W19	P6S_WG
W_06	K1_W18	P6S_WG
U_01	K1_U21	P6S_UW
U_02	K1_U22, K1_U28	P6S_UW
U_03	K1_U22	P6S_UW
U_04	K1_U23, K1_U24	P6S_UW
U_05	K1_U24	P6S_UW
U_06	K1_U24, K1_U29	P6S_UW
U_07	K1_U26, K1_U27	P6S_UW
K_01	K1_K02, K1_08	P6S_KO
K_02	K1_K04, K1_K07, K1_K08	P6S_KR, P6S_KO
K_03	K1_K01, K1_08	P6S_KK, P6S_KO

Wykaz literatury

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

1. Banachowski L, Diks K, Rytter W. *Algorytmy i struktury danych*, WNT Warszawa 2003
2. Cormen T. Laiserson C., Rivest R., Stein C. *Wprowadzenie do algorytmów*, WNT Warszawa 2007
3. Foryś M., Foryś W., *Teoria automatów i języków formalnych*, AOW EXIT Warszawa 2005
4. Harel D., *Rzecz o istocie informatyki : algorytmika*, WNT, Warszawa, 2008
5. Knuth D.E., *Sztuka programowania komputerów tom 3*, WNT, Warszawa 2002
6. Moczurad M, *Wybrane zagadnienia z teorii rekursji*, Wydawnictwo UJ, 2002
7. Wirth N., *Algorytmy + struktury danych = programy*, WNT, Warszawa, 2001
8. Wojtuszkiewicz K. *Programowanie strukturalne i obiektowe* PWN Warszawa 2009/2010

B. Literatura uzupełniająca

1. Kluźniak F., Szpakowicz S., *Prolog*, WNT Warszawa, 1983
2. Wróblewski P., *Algorytmy : struktury danych i techniki programowania*, Helion Gliwice, 2003

Kontakt

dr inż. Zbigniew Ledóchowski, zbigniew.ledochowski@apsl.edu.pl