

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2023/2024

Wydział Inżynierii Elektrycznej i Komputerowej

Kierunek studiów: Informatyka w Inżynierii Komputerowej

Profil: Ogólnoakademicki

Forma studiów: niestacjonarne

Kod kierunku: IwIK

Stopień studiów: I

Specjalności: bez specjalności

1 INFORMACJE O PRZEDMIOCIE

NAZWA PRZEDMIOTU	Metody programowania
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	
KOD PRZEDMIOTU	WIEiK INFOR_W_INZ_KOMP oIN PK22 23/24
KATEGORIA PRZEDMIOTU	Przedmioty kierunkowe
LICZBA PUNKTÓW ECTS	5.00
SEMESTRY	2

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁADY	ĆWICZENIA	LABORATORIA	LABORATORIA KOMPUTERO- WE	PROJEKTY	
2	15	0	0	30	0	0

3 CELE PRZEDMIOTU

Cel 1 Poznanie podstawowych pojęć i zagadnień związanych z programowaniem i złożonością obliczeniową algorytmów.

Cel 2 Ugruntowanie znajomości podstawowych struktur danych i ich zastosowań. Poznanie wybranych zaawansowanych struktur danych i ich zastosowań.

Cel 3 Znajomość zbioru klasycznych metod programowania, zbioru wybranych zaawansowanych metod programowania, metod generacji obiektów kombinatorycznych i liczb losowych oraz obszarów zastosowań każdej z nich.

Cel 4 Napisanie i uruchomienie programów implementujących algorytmy klasycznych metod programowania i wybranych zaawansowanych metod programowania, programowe generatory obiektów kombinatorycznych i generatory liczb losowych. Udoskonalenie umiejętności programowania w języku C/C++ oraz indywidualnej pracy programistycznej.

4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1 Zaliczony przedmiot z pierwszego semestru: Podstawy programowania w C/C++.

5 EFEKTY KSZTAŁCENIA

EK1 Wiedza Znajomość wybranych zagadnień związanych z programowaniem (dualizm programowo-sprzętowy, hipoteza Sapira-Whorfa, klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, programowanie strukturalne, programowanie obiektowe, programowanie wizualne, styl programowania, przenośność oprogramowania).

EK2 Wiedza Znajomość podstawowych i wybranych zaawansowanych struktur danych (tablica, lista, kolejki FIFO i LIFO (stos), kolejka priorytetowa, graf nieskierowany i skierowany, drzewo, kopiec), ich zastosowań oraz związku pomiędzy wyborem struktury danych a złożonością algorytmu.

EK3 Wiedza Znajomość poznanego zbioru klasycznych metod programowania (rekursja, programowanie dynamiczne, wyszukiwanie wyczerpujące, wyszukiwanie z powrotami, algorytmy zachłanne, wyszukiwanie lokalne, metody iteracyjne i obszaru zastosowania każdej z nich).

EK4 Wiedza Znajomość reprezentacji i metod generowania różnych klas obiektów kombinatorycznych, metod programowej i sprzętowej generacji liczb losowych wraz z ich obszarami zastosowań.

EK5 Umiejętności Napisanie i uruchomienie programów w języku C/C++ dla poznanego zbioru metod programowania oraz generatorów obiektów kombinatorycznych i liczb losowych.

EK6 Kompetencje społeczne Praca w małym zespole, podział zadań, efektywna współpraca w osiągnięciu wyznaczonego celu, dzielenie się wiedzą, wywiązywanie się z przyjętych obowiązków, kierowanie pracą zespołu.

6 TREŚCI PROGRAMOWE

LABORATORIA KOMPUTEROWE		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
K1	Generacja krotek, k-podzbiorów zbioru n-elementowego, podziałów zbioru, permutacji zbioru, drzew t-narych o n wierzchołkach wewnętrznych - w różnych reprezentacjach i porządkach liniowych.	6
K2	Generacja grafów o zadanych własnościach (grafy losowe, grafy R-MAT).	2
K3	Proste algorytmy rekurencyjne (problem do wyboru).	4
K4	Programowanie dynamiczne (problem do wyboru).	2

LABORATORIA KOMPUTEROWE		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
K5	Algorytmy zachłanne : kodowanie Huffmana, problem minimalnej bazy macierzy boolowskiej.	4
K6	Algorytmy z powrotami, algorytm MIN-MAX i odcinanie alfa-beta (problem do wyboru).	2
K7	Wyszukiwanie lokalne : algorytmy GSAT i WALKSAT dla problemu Spełnialności (SAT), algorytm dla problemu minimalnej bazy macierzy boolowskiej.	2
K8	Algorytmy iteracyjne.	2
K11	Laboratoria zaliczeniowe (2h) po każdym bloku tematach laboratoryjnych (6-10h).	6

WYKŁADY		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Definicja programowania. Rys historyczny rozwoju programowania. Dualizm programowo-sprzętowy. Języki i środowiska programowania. Hipoteza Sapira-Whorfa. Programowanie - klasyfikacje. Wymagania jakościowe. Wprowadzenie do algorytmów i ich złożoności obliczeniowej. Klasyfikacja problemów. Notacje asymptotyczne. Złożoność czasowa i pamięciowa. Metody programowania.	1
W2	Podstawowe struktury danych, ich reprezentacje, elementarne operacje i przykłady zastosowań. Tablica. Lista i jej warianty: kolejka FIFO, kolejka LIFO (stos), kolejka podwójna, kolejka priorytetowa. Zbiór. Graf nieskierowany i skierowany. Drzewo. Struktury, unie. Abstrakcyjne typy danych.	1
W3	Klasyczne techniki programowania. Programowanie strukturalne. Instrukcje strukturalne. Metody projektowania programów. Programowanie modularne. Programowanie wizualne. .	1
W4	Styl programowania a jakość programu. Nazwy. Wyrażenia i instrukcje. Jednolitość stylu, idiomy. Makroinstrukcje a funkcje. Liczby magiczne. Komentarze. Rola dobrego stylu. Zasady dobrego stylu. Przenośność oprogramowania. Tworzenie przenośnego oprogramowania. Język. Pliki nagłówkowe i biblioteki. Organizacja programu. Izolowanie. Wymiana danych - kolejność bajtów. Format CVF. Przenośność i uaktualnianie. Uniwersalne kodowanie.	1
W5	Pojęcie rekursji. Rekursja liniowa i drzewiasta. Rekurencyjne wyznaczanie wartości ciągów: silnia, liczby Fibonacciego, współczynniki dwumienne. Inne algorytmy rekurencyjne: wieże Hanoi, sortowanie MergeSort i QuickSort, trawersowanie drzew binarnych, Analiza zalet i wad rekursji. Eliminacja rekursji: iteracja, zastosowanie stosu. Rozwiązywanie rekurencji: metoda podstawiania, metoda iteracyjna, metoda rekurencji uniwersalnej.	2

WYKŁADY		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W6	Problem generacji liczb losowych. Liczby losowe a pseudolosowe. Zastosowania generatorów liczb losowych. Metody programowej generacji liczb losowych. Przykłady obliczeniowe. Statystyczna ocena jakości generatorów. Metody sprzętowej generacji liczb losowych.	1
W7	Generacja obiektów kombinatorycznych. Podstawowe typy obiektów: kombinacje, permutacje, podziały itp.. Reprezentacja obiektów kombinatorycznych za pomocą funkcji wyboru rodziny indeksowanej zbiorów. Porządki leksykograficzne na zbiorze funkcji wyboru. Generacja w porządku leksykograficznym. Algorytmy generacyjne. Zastosowania : wyszukiwanie wyczerpujące, testowanie algorytmów kombinatorycznych.	1
W8	Idea programowania dynamicznego. Wyznaczanie wartości ciągów metodą programowania dynamicznego: silnia, liczby Fibonacciego, współczynniki dwumienne, liczby Stirlinga. Własność optymalnej podstruktury. Inne algorytmy programowania dynamicznego: problem podziału zbioru, dyskretny problem plecakowy, problem iloczynu łańcucha macierzy. Wymagania czasowe i pamięciowe algorytmów programowania dynamicznego.	2
W9	Idea algorytmu zachłannego. Algorytm przybliżony lub dokładny (własność optymalnej podstruktury). Przykłady klasycznych algorytmów zachłannych: algorytm eliminacji Gaussa, algorytm Kruskala dla problemu MST, algorytm kodowania Huffmana. Wady i zalety algorytmów zachłannych.	1
W10	Wyszukiwanie wyczerpujące w przestrzeni rozwiązań a idea algorytmu wyszukiwania z nawrotami. Przykład : Problem plecakowy, problem komiwojażera (TSP). Przykłady klasycznych algorytmów z nawrotami: problem skoczka szachowego, problem ośmiu hetmanów. Wady i zalety algorytmów z nawrotami.	1
W11	Koncepcja wyszukiwania lokalnego. Przykład: problem spełnialności (SAT). Wyszukiwanie lokalne w optymalizacji. Przykłady: pokrycie wierzchołkowe grafu (VCP), minimalne drzewo rozpinające (MST), problem komiwojażera (TSP), Przykładowe operacje i definicje sąsiedztwa. Algorytm wyszukiwania z listą tabu (TS). Przykład : problem MST z ograniczeniami. Algorytm symulowanego wyżarzania (SA). Zalety i wady wyszukiwania lokalnego.	2
W12	Idea populacyjnych algorytmów iteracyjnych. Podstawowe typy iteracyjnych algorytmów optymalizacyjnych. Optymalizacja dyskretna - problem kolorowania grafu. Algorytmy ewolucyjne (EA).	1

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Praca w grupach

8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	45
Konsultacje przedmiotowe	2
Egzaminy i zaliczenia w sesji	0
konsultacje na odległość (zdalne)	3
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	15
Opracowanie wyników	15
Przygotowanie raportu, projektu, prezentacji, dyskusji	10
programowanie, testowanie programów	35
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	125
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	5.00

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Laboratorium : programy + sprawozdania z ćwiczeń laboratoryjnych

F2 Laboratorium (projekt zespołowy) : program + raport

OCENA PODSUMOWUJĄCA

P1 Średnia ważona z ocen formujących 1 i 2

WARUNKI ZALICZENIA PRZEDMIOTU

W1 Obecności na wykładach i laboratoriach komputerowych

W2 Zaliczenie laboratorium i projektu (ocena podsumowująca 1)

OCENA AKTYWNOŚCI BEZ UDZIAŁU NAUCZYCIELA

B1 W ramach ocen formujących

KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1	
NA OCENĘ 2.0	Poniżej 50 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z programowaniem wymaganą na ocenę 5.0
NA OCENĘ 3.0	50-59 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z programowaniem wymaganą na ocenę 5.0
NA OCENĘ 3.5	60-69 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z programowaniem wymaganą na ocenę 5.0
NA OCENĘ 4.0	70-79 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z programowaniem wymaganą na ocenę 5.0
NA OCENĘ 4.5	80-89 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z programowaniem wymaganą na ocenę 5.0
NA OCENĘ 5.0	90-100 % punktów uzyskanych za posiadaną wiedzę: Znajomość wybranych zagadnień związanych z programowaniem (dualizm programowo-sprzętowy, hipoteza Sapira-Whorfa, klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, programowanie strukturalne, programowanie obiektowe, programowanie wizualne, styl programowania, przenośność oprogramowania).
EFEKT KSZTAŁCENIA 2	
NA OCENĘ 2.0	Poniżej 50 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych ze strukturami danych wymaganą na ocenę 5.0
NA OCENĘ 3.0	50-59 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych ze strukturami danych wymaganą na ocenę 5.0
NA OCENĘ 3.5	60-69 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych ze strukturami danych wymaganą na ocenę 5.0
NA OCENĘ 4.0	70-79 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych ze strukturami danych wymaganą na ocenę 5.0
NA OCENĘ 4.5	80-89 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych ze strukturami danych wymaganą na ocenę 5.0
NA OCENĘ 5.0	90-100 % punktów uzyskanych za posiadaną wiedzę: Znajomość podstawowych i wybranych zaawansowanych struktur danych (tablica, lista, kolejki FIFO i LIFO (stos), kolejka priorytetowa, graf nieskierowany i skierowany, drzewo, kopiec), ich zastosowań oraz związku pomiędzy wyborem struktury danych a złożonością algorytmu.
EFEKT KSZTAŁCENIA 3	
NA OCENĘ 2.0	Poniżej 50 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z klasycznymi metodami programowania wymaganą na ocenę 5.0
NA OCENĘ 3.0	50-59 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z klasycznymi metodami programowania wymaganą na ocenę 5.0

NA OCENĘ 3.5	60-69 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z klasycznymi metodami programowania wymaganą na ocenę 5.0
NA OCENĘ 4.0	70-79 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z klasycznymi metodami programowania wymaganą na ocenę 5.0
NA OCENĘ 4.5	80-89 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z klasycznymi metodami programowania wymaganą na ocenę 5.0
NA OCENĘ 5.0	90-100 % punktów uzyskanych za posiadaną wiedzę: Znajomość poznanego zbioru klasycznych metod programowania (rekursja, programowanie dynamiczne, wyszukiwanie wyczerpujące, wyszukiwanie z powrotami, algorytmy zachłanne, wyszukiwanie lokalne, metody iteracyjne i obszaru zastosowania każdej z nich.
EFEKT KSZTAŁCENIA 4	
NA OCENĘ 2.0	Poniżej 50 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z reprezentacjami i metodami generowania obiektów kombinatorycznych oraz wybranymi metodami generacji liczb losowych wymaganą na ocenę 5.0
NA OCENĘ 3.0	50-59 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z reprezentacjami i metodami generowania obiektów kombinatorycznych oraz wybranymi metodami generacji liczb losowych wymaganą na ocenę 5.0
NA OCENĘ 3.5	60-69 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z reprezentacjami i metodami generowania obiektów kombinatorycznych oraz wybranymi metodami generacji liczb losowych wymaganą na ocenę 5.0
NA OCENĘ 4.0	70-79 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z reprezentacjami i metodami generowania obiektów kombinatorycznych oraz wybranymi metodami generacji liczb losowych wymaganą na ocenę 5.0
NA OCENĘ 4.5	80-89 % punktów uzyskanych za posiadaną wiedzę dotyczącą zagadnień związanych z reprezentacjami i metodami generowania obiektów kombinatorycznych oraz wybranymi metodami generacji liczb losowych wymaganą na ocenę 5.0
NA OCENĘ 5.0	90-100 % punktów uzyskanych za posiadaną wiedzę: Znajomość reprezentacji i metod generowania różnych klas obiektów kombinatorycznych, metod programowej i sprzętowej generacji liczb losowych wraz z obszarami ich zastosowań.
EFEKT KSZTAŁCENIA 5	
NA OCENĘ 2.0	Poniżej 50 % punktów uzyskanych za napisanie i uruchomienie programów w języku C/C++ dla poznanego zbioru klasycznych metod programowania wymaganych na ocenę 5.0
NA OCENĘ 3.0	50-59 % punktów uzyskanych za napisanie i uruchomienie programów w języku C/C++ dla poznanego zbioru klasycznych metod programowania wymaganych na ocenę 5.0

NA OCENĘ 3.5	60-69 % punktów uzyskanych za napisanie i uruchomienie programów w języku C/C++ dla poznanego zbioru klasycznych metod programowania wymaganych na ocenę 5.0
NA OCENĘ 4.0	70-79 % punktów uzyskanych za napisanie i uruchomienie programów w języku C/C++ dla poznanego zbioru klasycznych metod programowania na ocenę 5.0
NA OCENĘ 4.5	80-89 % punktów uzyskanych za napisanie i uruchomienie programów w języku C/C++ dla poznanego zbioru klasycznych metod programowania na ocenę 5.0
NA OCENĘ 5.0	90-100 % punktów uzyskanych za: Napisanie i uruchomienie programów w języku C/C++ dla poznanego zbioru klasycznych metod programowania oraz generatorów obiektów kombinatorycznych i liczb losowych.
EFEKT KSZTAŁCENIA 6	
NA OCENĘ 2.0	Bierność lub niewywiązywanie się z przyjętych obowiązków lub destrukcyjny wpływ na pracę zespołu. Spełnienie poniżej 50 % wymagań na ocenę 5.0
NA OCENĘ 3.0	Umiarkowana aktywność lub niewywiązanie się z części przyjętych obowiązków lub brak kreatywności lub brak współpracy w zespole. Spełnienie 50-59 % wymagań na ocenę 5.0
NA OCENĘ 3.5	Zadowalająca aktywność, wywiązywanie się z przyjętych obowiązków, przejawy kreatywności, poprawna współpraca w zespole w roli wykonawcy (łącznie). Spełnienie 60-69 % wymagań na ocenę 5.0
NA OCENĘ 4.0	Dobra aktywność, wywiązywanie się z przyjętych obowiązków, kreatywność, efektywna współpraca w zespole (łącznie). Spełnienie 70-79 % wymagań na ocenę 5.0
NA OCENĘ 4.5	Dobra aktywność, wywiązywanie się z przyjętych obowiązków, kreatywność, efektywna współpraca w zespole, transfer wiedzy do pozostałych członków zespołu (łącznie). Spełnienie 80-89 % wymagań na ocenę 5.0
NA OCENĘ 5.0	Wyróżniająca się aktywność, wywiązywanie się z przyjętych obowiązków, kreatywność, efektywna współpraca w zespole, transfer wiedzy do pozostałych członków zespołu, wykazanie się umiejętnościami kierowniczymi (łącznie). Spełnienie 90-100 % wymagań.

10 MACIERZ REALIZACJI PRZEDMIOTU

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK1	K_W03 K_W06 K_W07 K_W08 K_W09 K_W17	Cel 1	W1 W2 W4 W5 W6	N1	F1 P1
EK2	K_W03 K_W06 K_W07 K_W08 K_W09	Cel 2	K1 K2 K3 K4 K5 K6 K7 K8 W3	N1 N2	F1 F2 P1
EK3	K_W03 K_W06 K_W07 K_W08 K_W09	Cel 3	K3 K4 K5 K6 K7 W3 W7 W8 W9 W10 W11	N1 N2	F1 F2 P1
EK4	K_W01 K_U01 K_U06 K_U08 K_U09	Cel 3	W12	N1	F1 F2
EK5	K_U01 K_U04 K_U05 K_U06 K_U08 K_U11 K_U12	Cel 4	K1 K2 K8 K11 W2 W3 W4 W5 W8 W9 W10 W11	N2	F2
EK6	K_K01 K_K02 K_K03 K_K04 K_K06	Cel 4	K11 W12	N3	F2

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

- [1] | Cormen T.H., Leiserson C.E., Rivest R.L., C. Stein — *Wprowadzenie do algorytmów*, Warszawa, 2007, WN-T
- [2] | Dasgupta S., Papadimitriou C., Vazirani U. — *Algorytmy*, Warszawa, 2010, PWN
- [3] | Lipski W. — *Kombinatoryka dla programistów*, Warszawa, 2004, WN-T
- [4] | Kernighan B.W., Pike R. — *Lekcja programowania*, Warszawa, 2002, WN-T

LITERATURA UZUPEŁNIAJĄCA

- [1] | Sedgewick R. — *Algorytmy w C++. Sortowanie i wyszukiwanie*, Warszawa, 1999, Wydawnictwo RM
- [2] | Sedgewick R. — *Algorytmy w C++. Algorytmy grafowe*, Warszawa, 2003, Wydawnictwo RM

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr hab. inż. Prof PK Zbigniew Kokosiński (kontakt: zk@pk.edu.pl)

OSOBY PROWADZĄCE PRZEDMIOT

1 dr hab. inż. Zbigniew Kokosiński (kontakt: zk@pk.edu.pl)

2 mgr inż. Grzegorz Nowakowski (kontakt: gnowakowski@pk.edu.pl)

3 mgr inż. Piotr Kędra (kontakt: piotr.kedra@pk.edu.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....
.....
.....