Politechnika Krakowska im. Tadeusza Kościuszki

Karta przedmiotu

obowiązuje studentów rozpoczynających studia w roku akademickim 2013/2014

Wydział Inżynierii Elektrycznej i Komputerowej

Kierunek studiów: Elektrotechnika — Profil: Ogólnoakademicki

Forma sudiów: niestacjonarne — Kod kierunku: Elek

Stopień studiów: II

Specjalności: Informatyczne systemy automatyki

1 Informacje o przedmiocie

Nazwa przedmiotu	Projektowanie i analiza algorytmów
Nazwa przedmiotu w języku angielskim	
Kod przedmiotu	WIEiK ELEKTROTECH oIIN PW15 13/14
Kategoria przedmiotu	Przedmioty specjalnościowe
Liczba punktów ECTS	4.00
Semestry	3

2 Rodzaj zajęć, liczba godzin w planie studiów

Semestr	Wykłady	Ćwiczenia	Laboratoria	Laboratoria komputerowe	Projekty	
3	15	0	0	10	5	0

3 Cele przedmiotu

Cel 1 Poznanie podstawowych pojęć i zagadnień wiązanych ze złożonością obliczeniową algorytmów.

Cel 2 Znajomość poznanego zbioru klasycznych metod programowania, podstaw wybranych zaawansowanych metod programowania.

Cel 3 Napisanie i uruchomienie prostych programów dla wybranych metod programowania. Udoskonalenie umiejętności programowania w języku C/C++ oraz indywidualnej pracy programistycznej.

Cel 4 Wykonanie wybranego zespołowego projektu programistycznego. Udoskonalenie umiejętności programowania w języku C/C++ oraz zespołowej pracy programistycznej.

4 Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

1 Zaliczony przedmiot : Programowanie w C++.

5 Efekty kształcenia

EK1 Wiedza Znajomość wybranych zagadnień: klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).

EK2 Wiedza Znajomość poznanego zbioru klasycznych metod programowania i obszaru zastosowania każdej z nich.

EK3 Umiejętności Napisanie i uruchomienie programów w języku C/C++ dla wybranych klasycznych metod programowania.

EK4 Kompetencje społeczne Praca w małym zespole, podział zadań, efektywna współpraca w osiąganiu wyznaczonego celu, dzielenie się wiedzą, wywiązywanie się z przyjętych obowiązków, kierowanie pracą zespołu

6 Treści programowe

Wykłady		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
W1	Wprowadzenie do algorytmów i ich złożoności obliczeniowej. Klasyfikacja problemów. Notacje asymptotyczne. Złożoność czasowa i pamięciowa. Klasyfikacja algorytmów. Hierarchia złożoności.	2
W2	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
W3	Idea programowania dynamicznego. Wyznaczanie wartości ciągów metodą programowania dynamicznego: silnia, liczby Fibonacciego, współczynniki dwumienne, liczby Stirlinga. Inne algorytmy programowania dynamicznego: problem podziału zbioru, dyskretny problem plecakowy, problem iloczynu łańcucha macierzy. Wymagania czasowe i pamięciowe algorytmów programowania dynamicznego.	2
W4	Idea algorytmu zachłannego. Własność optymalnej podstruktury. Elementy teorii matroidów. Klasy matroidów: podziałowe, macierzowe, grafowe, transwersalne. Przykłady klasycznych algorytmów zachłannych: algorytm Gaussa, algorytm Kruskala, algorytm Huffmana. Wady i zalety algorytmów zachłannych.	2

Wykłady		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
W5	Wyszukiwanie wyczerpujące w przestrzeni rozwiązań a idea algorytmu wyszukiwania z nawrotami. Przykład : problem komiwojażera (TSP). Przykłady klasycznych algorytmów z nawrotami: problem skoczka szachowego, problem ośmiu hetmanów. Gry i ich reprezentacje w postaci drzew gier. Ocena rozwiązań w węzłach. Technika odcinania alfa-beta gałęzi drzewa gry. Wady i zalety algorytmów z nawrotami.	2
W6	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. Ulepszenia i modyfikacje wyszukiwania lokalnego. Algorytm wyszukiwania z listą tabu (TS). Przykład : problem MST z ograniczeniami. Zalety i wady wyszukiwania lokalnego.	2
W7	Idea algorytmów iteracyjnych. Podstawowe typy algorytmów iteracyjnych. Wyszukiwanie z tabu. Symulowane wyżarzanie. Algorytmy ewolucyjne. Schematy zrównoleglenia programów iteracyjnych.	1
W8	Programowanie równoległe. Charakterystyka systemów równoległych. Organizacja obliczeń równoległych. Przykład obliczeń równoległych sito Eratostenesa. Podstawowe miary jakości obliczeń równoległych. Zależności pomiędzy miarami jakości. Przykład oceny jakości systemu równoległego. Bariera: inherentna sekwencyjność problemów.	2

Laboratoria komputerowe		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
K1	Generacja grafów w reprezentacji tablicowej o zadanych własnościach.	2
K2	Programowanie dynamiczne : problem plecakowy.	3
K3	Algorytmy zachłanne : kodowanie Huffmana.	3
K4	Wyszukiwanie lokalne : algorytmy GSAT i WALKSAT dla problemu Spełnialności.	2

Projekty		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
P1	Projekt zespołowy implementacji wybranego algorytmu dla problemu z klasy NP (TSP, kolorowanie grafu).	5

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	0
Konsultacje przedmiotowe	10
Egzaminy i zaliczenia w sesji	10
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	25
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	15
programowanie, testowanie programów	30
Sumaryczna liczba godzin dla przedmiotu wynikająca z całego nakładu pracy studenta	**90**
Sumaryczna liczba punktów ECTS dla przedmiotu	4.00

9 Sposoby oceny

Ocena formująca

F1 Projekt zespołowy

F2 Sprawozdanie z ćwiczenia laboratoryjnego

Ocena podsumowująca

P1 Średnia ważona ocen formujących

P2 Zaliczenie pisemne

Warunki zaliczenia przedmiotu

W1 Obecności na wykładach i laboratoriach komputerowych

Kryteria oceny

Efekt kształcenia 1	
Na ocenę 2.0	Brak znajomości przynajmniej 3 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).
Na ocenę 3.0	Znajomość 3 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).symptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).
Na ocenę 3.5	Znajomość 4 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).
Na ocenę 4.0	Znajomość 5 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).symptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).
Na ocenę 4.5	Znajomość 6 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).symptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).
Na ocenę 5.0	Znajomość 6 spośród 6 zagadnień związanych z programowaniem (klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne).symptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne) oraz wiedza wykraczająca poza program wykładu.
Efekt kształcenia 2	
Na ocenę 2.0	Brak znajomości teoretycznej przynajmniej 3 spośród 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne).
Na ocenę 3.0	Znajomość teoretyczna 3 spośród 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne) wraz z przykładami zastosowań.
Na ocenę 3.5	Znajomość teoretyczna 4 spośród 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne) wraz z przykładami zastosowań 3 z nich.
Na ocenę 4.0	Znajomość teoretyczna 4 spośród 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne) wraz z przykładami zastosowań.

Na ocenę 4.5	Znajomość teoretyczna 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne) wraz z przykładami zastosowań 4 z nich.
Na ocenę 5.0	Znajomość teoretyczna 5 klasycznych metod programowania (rekursja, programowanie dynamiczne, algorytmy zachłanne, algorytmy z nawrotami, wyszukiwanie lokalne) wraz z przykładami zastosowań.
Efekt kształcenia 3	
Na ocenę 2.0	Brak napisania i uruchomienia co najmniej 1 programów w języku C/C++ dla zbioru podstawowych metod programowania.
Na ocenę 3.0	Napisanie i uruchomienie co najmniej 1 w pełni funkcjonalnego programu w języku C/C++ dla zbioru podstawowych metod programowania.
Na ocenę 3.5	Napisanie i uruchomienie co najmniej 2 programów w języku C/C++ dla zbioru podstawowych metod programowania (w tym 1 w pełni funkcjonalnego).
Na ocenę 4.0	Napisanie i uruchomienie 2 w pełni funkcjonalnych programów w języku C/C++ dla zbioru podstawowych metod programowania.
Na ocenę 4.5	Napisanie i uruchomienie co najmniej 3 programów w języku C/C++ dla zbioru podstawowych metod programowania (w tym 2 w pełni funkcjonalnych).
Na ocenę 5.0	Napisanie i uruchomienie 3 w pełni funkcjonalnych programów w języku C/C++ dla zbioru podstawowych metod programowania.
Efekt kształcenia 4	
Na ocenę 2.0	Bierność lub niewywiązywanie się z przyjętych obowiązków lub destrukcyjny wpływ na pracę zespołu.
Na ocenę 3.0	Umiarkowana aktywność lub niewywiązanie się z części przyjętych obowiązków lub brak kreatywności lub brak współpracy w zespole.
Na ocenę 3.5	Zadowalająca aktywność, wywiązanie się z przyjętych obowiązków, przejawy kreatywności, poprawna współpraca w zespole w roli wykonawcy (łącznie)
Na ocenę 4.0	Dobra aktywność, wywiązanie się z przyjętych obowiązków, kreatywność, efektywna współpraca w zespole (łącznie)
Na ocenę 4.5	Dobra aktywność, wywiązanie się z przyjętych obowiązków, kreatywność, efektywna współpraca w zespole, transfer wiedzy do pozostałych członków zespołu (łącznie)
Na ocenę 5.0	Wyróżniająca się aktywność, wywiązanie się z przyjętych obowiązków, kreatywność, efektywna współpraca w zespole, transfer wiedzy do pozostałych członków zespołu, wykazanie się umiejętnościami kierowniczymi (łącznie)

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	Cel 1	W1	N1	P2
EK2	K_W02, K_W03	Cel 2	W2 W3 W4 W5 W6 W7 W8	N1 N2	F2 P1 P2
EK3	K_U01, K_U03, K_U11	Cel 3	K1 K2 K3 K4	N2	F2 P1
EK4	K_U01, K_U03, K_U11	Cel 4	P1	N3	F1 P1

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] Kernighan B.W., Pike R. — *Lekcja programowania*, Warszawa, 2002, WN-T

[3] Lipski W. — *Kombinatoryka dla programistów*, Warszawa, 2004, 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 inż. Zbigniew Kokosiński (kontakt: Zbigniew.Kokosinski@pk.edu.pl)

13 Zatwierdzenie karty przedmiotu do realizacji

(miejscowość, data) (odpowiedzialny za przedmiot) (dziekan)

Przyjmuję do realizacji (data i podpisy osób prowadzących przedmiot)

.....