

# POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

## KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2014/2015

Wydział Inżynierii Elektrycznej i Komputerowej

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma studiów: stacjonarne

Kod kierunku: Info

Stopień studiów: I

Specjalności: bez specjalności

### 1 INFORMACJE O PRZEDMIOCIE

NAZWA PRZEDMIOTU	Metody programowania
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	Programming Methods
KOD PRZEDMIOTU	WIEiK INFOR oIS PK11 14/15
KATEGORIA PRZEDMIOTU	Przedmioty kierunkowe
LICZBA PUNKTÓW ECTS	6.00
SEMESTRY	2

### 2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁADY	ĆWICZENIA	LABORATORIA	LABORATORIA KOMPUTERO- WE	PROJEKTY	
2	30	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 poznanego zbioru klasycznych metod programowania, podstaw wybranych zaawansowanych metod programowania, metod generacji obiektów kombinatorycznych i liczb losowych oraz obszarów zastosowań każdej z nich.

**Cel 4** Napisanie i uruchomienie programów dla klasycznych metod programowania, programowych generatorów obiektów kombinatorycznych i generatorów liczb losowych. Udoskonalenie umiejętności programowania w języku C/C++ oraz indywidualnej pracy programistycznej.

**Cel 5** 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 z pierwszego semestru: Wstęp do programowania.

## 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, hierarchia złożoności obliczeniowej, programowanie strukturalne, styl programowania, przenośność oprogramowania).

**EK2 Wiedza** Znajomość podstawowych i wybranych zaawansowanych struktur danych i ich zastosowań (listy, stos, kolejki priorytetowe, tablice, grafy, drzewa, kopce).

**EK3 Wiedza** Znajomość poznanego zbioru klasycznych metod programowania i obszaru zastosowania każdej z nich.

**EK4 Umiejętności** Napisanie i uruchomienie programów w języku C/C++ dla poznanego zbioru klasycznych metod programowania.

**EK5 Wiedza** Znajomość reprezentacji i metod generowania obiektów kombinatorycznych, metod generacji liczb losowych wraz z ich obszarami zastosowań.

**EK6 Umiejętności** Napisanie i uruchomienie programów generujących obiekty kombinatoryczne i liczby losowe.

**EK7 Wiedza** Znajomość podstaw wybranych zaawansowanych metod programowania (metody iteracyjne i równoległe) i obszaru zastosowania każdej z nich.

**EK8 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	Definicja programowania. Rys historyczny rozwoju programowania od starożytności do pierwszych komputerów. Dualizm programowo-sprzętowy. Języki i środowiska programowania. Hipoteza Sapira-Whorfa. Programowanie klasyfikacje. Wymagania jakościowe. Metody programowania.	2

WYKŁADY		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W2	Wprowadzenie do algorytmów i ich złożoności obliczeniowej. Klasyfikacja problemów. Notacje asymptotyczne. Złożoność czasowa i pamięciowa. Klasyfikacja algorytmów. Hierarchia złożoności. Inne kryteria klasyfikacji algorytmów. Obszary zastosowań algorytmiki.	2
W3	Podstawowe struktury danych. Tablica. Lista i jej warianty: kolejka, stos, kolejka podwójna. Zbiór. Graf. Drzewo. Kopiec. Kolejka priorytetowa. Struktury, unie. Abstrakcyjne typy danych.	2
W4	Klasyczne techniki programowania. Programowanie strukturalne. Wady instrukcji GO TO. Instrukcje strukturalne. Metody projektowania programów. Zasada abstrakcji - poziomy abstrakcji. Programowanie modularne. Graficzne przedstawianie programów. Schematy blokowe i diagramy strukturalne. Niestrukturalność. Metoda skrzynek zastępczych. Automatyzacja w tworzeniu programów CASE.	2
W5	Styl programowania a jakość programu. Nazwy. Wyrażenia i instrukcje. Jednolitość stylu, idiomy. Makroinstrukcje a funkcje. Liczby magiczne. Komentarze. Rola dobrego stylu. Zasady dobrego stylu.	2
W6	Przenośność oprogramowania. Tworzenie przenośnego oprogramowania. Język. Pliki nagłówkowe i biblioteki. Organizacja programu. Izolowanie. Wymiana danych - kolejność bajtów. Format CVF. Przenośność i uaktualnianie. Uniwersalne kodowanie.	2
W7	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
W8	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
W9	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
W10	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

WYKŁADY		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
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. Ulepszenia i modyfikacje wyszukiwania lokalnego. Algorytm wyszukiwania z listą tabu (TS). Przykład : problem MST z ograniczeniami. Zalety i wady wyszukiwania lokalnego.	2
W12	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.	2
W13	Generacja obiektów kombinatorycznych. Podstawowe typy obiektów: kombinacje, permutacje, nieporządki, podziały, drzewa 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.	2
W14	Idea algorytmów iteracyjnych. Podstawowe typy algorytmów iteracyjnych. Wyszukiwanie z tabu. Symulowane wyżarzanie. Algorytmy ewolucyjne. Schematy równoleglenia programów iteracyjnych.	2
W15	Programowanie równoległe i pojęcia pokrewne. 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. Języki równoległe. Przenośność. Dalszy rozwój systemów równoległych.	2

LABORATORIA KOMPUTEROWE		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
K1	Generacja krotek, kombinacji z powtórzeniami i kombinacji Generacja podziałów. Generacja wariacji i permutacji.	4
K2	Generacja grafów o zadanych własnościach.	2
K3	Proste algorytmy rekurencyjne.	4
K4	Programowanie dynamiczne : problem plecakowy.	4
K5	Algorytmy z powrotami: problem plecakowy, problem skoczka szachowego lub problem 8 hetmanów.	2
K6	Algorytmy zachłanne : kodowanie Huffmana.	4
K7	Wyszukiwanie lokalne : algorytmy GSAT i WALKSAT dla problemu Spełnialności.	2

LABORATORIA KOMPUTEROWE		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
<b>K8</b>	Porównanie generatorów liczb losowych. Statystyka Chi Kwadrat i inne metody testowe.	4
<b>K9</b>	Generacja grafów Gk. Kolorowanie odległościowe grafów.	4

## 7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Praca w grupach

## 8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
<b>Godziny kontaktowe z nauczycielem akademickim, w tym:</b>	
Godziny wynikające z planu studiów	60
Konsultacje przedmiotowe	0
Egzaminy i zaliczenia w sesji	4
<b>Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:</b>	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	30
Opracowanie wyników	15
Przygotowanie raportu, projektu, prezentacji, dyskusji	15
programowanie, testowanie programów	56
<b>SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA</b>	<b>180</b>
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	6.00

## 9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Test

F2 Projekt indywidualny

F3 Projekt zespołowy

F4 Sprawozdanie z ćwiczenia laboratoryjnego

#### OCENA PODSUMOWUJĄCA

P1 Średnia ważona ocen formujących

P2 Egzamin pisemny

#### WARUNKI ZALICZENIA PRZEDMIOTU

W1 Obecności na wykładach i laboratoriach komputerowych

#### OCENA AKTYWNOŚCI BEZ UDZIAŁU NAUCZYCIELA

B1 W ramach ocen formujących

#### KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1	
NA OCENĘ 2.0	Brak znajomości przynajmniej 6 spośród 10 zagadnień związanych z programowaniem (dualizm programowo-sprzętowy, hipoteza Sapira-Whorfa, klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne, styl programowania, przenośność oprogramowania).
NA OCENĘ 3.0	Znajomość 6 spośród 10 zagadnień związanych z programowaniem (dualizm programowo-sprzętowy, hipoteza Sapira-Whorfa, klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne, styl programowania, przenośność oprogramowania).
NA OCENĘ 3.5	Znajomość 7 spośród 10 zagadnień związanych z programowaniem (dualizm programowo-sprzętowy, hipoteza Sapira-Whorfa, klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne, styl programowania, przenośność oprogramowania).
NA OCENĘ 4.0	Znajomość 8 spośród 10 zagadnień związanych z programowaniem (dualizm programowo-sprzętowy, hipoteza Sapira-Whorfa, klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne, styl programowania, przenośność oprogramowania).
NA OCENĘ 4.5	Znajomość 9 spośród 10 zagadnień związanych z programowaniem (dualizm programowo-sprzętowy, hipoteza Sapira-Whorfa, klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne, styl programowania, przenośność oprogramowania).

NA OCENĘ 5.0	Znajomość 10 zagadnień związanych z programowaniem (dualizm programowo-sprzętowy, hipoteza Sapira-Whorfa, klasyfikacja problemów, notacje asymptotyczne, złożoność czasowa i pamięciowa, klasyfikacja algorytmów, hierarchia złożoności obliczeniowej, programowanie strukturalne, styl programowania, przenośność oprogramowania).
EFEKT KSZTAŁCENIA 2	
NA OCENĘ 2.0	Brak znajomości przynajmniej 4 podstawowych struktur danych i ich zastosowań (lista jedno- i dwukierunkowa, stos, tablica, graf).
NA OCENĘ 3.0	Znajomość 4 podstawowych struktur danych i ich zastosowań (lista jedno- i dwukierunkowa, stos, tablica, graf).
NA OCENĘ 3.5	Znajomość 5 podstawowych struktur danych i ich zastosowań (lista jedno- i dwukierunkowa, stos, kolejka priorytetowa, tablica, graf).
NA OCENĘ 4.0	Znajomość 6 podstawowych struktur danych i ich zastosowań (lista jedno- i dwukierunkowa, stos, kolejka priorytetowa, tablica, graf, drzewo binarne).
NA OCENĘ 4.5	Znajomość 7 podstawowych struktur danych i ich zastosowań (lista jedno- i dwukierunkowa, stos, kolejka priorytetowa, tablica, graf, drzewo binarne, kopiec).
NA OCENĘ 5.0	Znajomość 8 podstawowych struktur danych i ich zastosowań (lista jedno- i dwukierunkowa, stos, kolejka priorytetowa, tablica, graf, drzewo binarne, drzewo t-arne, kopiec).
EFEKT KSZTAŁCENIA 3	
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 4	
NA OCENĘ 2.0	Napisanie i uruchomienie mniej niż 3 programów w języku C/C++ dla zbioru podstawowych metod programowania lub mniej niż 2 w pełni funkcjonalnych.
NA OCENĘ 3.0	Napisanie i uruchomienie co najmniej 3 programów w języku C/C++ dla zbioru podstawowych metod programowania (w tym 2 w pełni funkcjonalnych).
NA OCENĘ 3.5	Napisanie i uruchomienie co najmniej 4 programów w języku C/C++ dla zbioru podstawowych metod programowania (w tym 3 w pełni funkcjonalnych).
NA OCENĘ 4.0	Napisanie i uruchomienie 4 w pełni funkcjonalnych programów w języku C/C++ dla zbioru podstawowych metod programowania.
NA OCENĘ 4.5	Napisanie i uruchomienie co najmniej 5 programów w języku C/C++ dla zbioru podstawowych metod programowania (w tym 4 w pełni funkcjonalnych).
NA OCENĘ 5.0	Napisanie i uruchomienie 5 w pełni funkcjonalnych programów w języku C/C++ dla zbioru podstawowych metod programowania.
EFEKT KSZTAŁCENIA 5	
NA OCENĘ 2.0	Brak znajomości reprezentacji i metod generowania co najmniej 3 obiektów kombinatorycznych (kombinacje, permutacje, podziały zbioru) oraz co najmniej 3 wybranych metod kongruencyjnych generacji liczb losowych.
NA OCENĘ 3.0	Znajomość reprezentacji i metod generowania obiektów kombinatorycznych (kombinacje, permutacje, podziały zbioru) oraz co najmniej 3 wybranych kongruencyjnych metod generacji liczb losowych.
NA OCENĘ 3.5	Znajomość reprezentacji i metod generowania obiektów kombinatorycznych (kombinacje, permutacje, podziały zbioru, drzewa binarne) oraz 4 kongruencyjnych metod generacji liczb losowych.
NA OCENĘ 4.0	Znajomość reprezentacji i metod generowania obiektów kombinatorycznych (kombinacje, permutacje, podziały zbioru, drzewa binarne, grafy) oraz metod generacji liczb losowych : 4 kongruencyjnych i 1 opartej na LFSR.
NA OCENĘ 4.5	Znajomość reprezentacji i metod generowania obiektów kombinatorycznych. (kombinacje, permutacje, podziały zbioru, drzewa binarne i t-arne, grafy) oraz 5 różnorodnych programowych i 2 sprzętowych metod generacji liczb losowych.
NA OCENĘ 5.0	Znajomość reprezentacji i metod generowania obiektów kombinatorycznych (kombinacje, permutacje, nieporządki, podziały zbioru, drzewa binarne i t-arne, grafy) oraz 5 różnorodnych programowych i 3 sprzętowych metod generacji liczb losowych.
EFEKT KSZTAŁCENIA 6	
NA OCENĘ 2.0	Napisanie i uruchomienie programów generujących mniej niż 3 obiekty kombinatoryczne (kombinacje, permutacje, podziały zbioru) lub mniej niż 1 program generujący liczby losowe.



NA OCENĘ 3.0	Napisanie i uruchomienie programów generujących 3 obiektów kombinatorycznych (kombinacje, permutacje, podziały zbioru) i 1 program generujący liczby losowe.
NA OCENĘ 3.5	Napisanie i uruchomienie programów generujących 4 obiektów kombinatorycznych (kombinacje, permutacje, podziały zbioru, drzewa binarne) i 1 program generujący liczby losowe z wbudowaną statystyką Chi Kwadrat.
NA OCENĘ 4.0	Napisanie i uruchomienie programów generujących 5 obiektów kombinatorycznych (kombinacje, permutacje, podziały zbioru, drzewa binarne, grafy) i 1 program generujący liczby losowe z wbudowaną statystyką Chi Kwadrat.
NA OCENĘ 4.5	Napisanie i uruchomienie programów generujących 6 obiektów kombinatorycznych (kombinacje, permutacje, podziały zbioru, drzewa binarne i t-arne, grafy) i 2 programy generujące liczby losowe z wbudowanymi statystykami.
NA OCENĘ 5.0	Napisanie i uruchomienie programów generujących 7 obiektów kombinatorycznych (kombinacje, permutacje, nieporządki, podziały zbioru, drzewa binarne i t-arne, grafy) i 2 programy generujące liczby losowe z wbudowanymi statystykami.
EFEKT KSZTAŁCENIA 7	
NA OCENĘ 2.0	Brak znajomości przynajmniej 1 iteracyjnej metody programowania z przykładami zastosowań.
NA OCENĘ 3.0	Znajomość teoretyczna 1 zaawansowanej metody programowania ( iteracyjnej) wraz z przykładami zastosowań.
NA OCENĘ 3.5	Znajomość teoretyczna 2 zaawansowanych metod programowania (1 iteracyjnej i 1 równoległej) wraz z przykładami zastosowań.
NA OCENĘ 4.0	Znajomość teoretyczna 3 zaawansowanych metod programowania (2 iteracyjnych i 1 równoległych) wraz z przykładami zastosowań.
NA OCENĘ 4.5	Znajomość teoretyczna 4 zaawansowanych metod programowania (2 iteracyjnych i 2 równoległych) wraz z przykładami zastosowań.
NA OCENĘ 5.0	Znajomość teoretyczna 5 zaawansowanych metod programowania (3 iteracyjnych i 2 równoległych) wraz z przykładami zastosowań.
EFEKT KSZTAŁCENIA 8	
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	Zadowolająca 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		Cel 1	W1 W2 W4 W5 W6	N1	F1 P1 P2
EK2		Cel 2	W3 K1 K2 K3 K4 K5 K6 K7 K8 K9	N1 N2	F1 F2 F3 P1 P2
EK3		Cel 3	W7 W8 W9 W10 W11	N1	F1 P1 P2
EK4		Cel 4	K3 K4 K5 K6 K7	N2	F2 F4
EK5		Cel 3	W12 W13	N1	F1 P1 P2
EK6		Cel 4	K1 K2 K8	N2	F2 F4
EK7		Cel 3	W14 W15	N1	F1 P2
EK8		Cel 5	K8 K9	N2 N3	F3 F4

## 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)

2 prof. dr hab. inż. Włodzimierz Samotyj (kontakt: vsamotyj@pk.edu.pl)

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

## 13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

---

(miejsowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

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