**Nazwa przedmiotu:**

Algorytmy i struktury danych

**Koordynator przedmiotu:**

prof. nzw. dr hab. inż. Barbara Putz

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Elektronika i Telekomunikacja

**Grupa przedmiotów:**

Przedmioty informatyki - obowiązkowe

**Kod przedmiotu:**

AISDZ

**Semestr nominalny:**

2 / rok ak. 2017/2018

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

Zajęcia kontaktowe z nauczycielem:
1. konsultacje mailowe z nauczycielem: 20 h
2. zajęcia stacjonarne na uczelni: 4 h
3. egzamin (w tym zaliczanie projektu): 2 h

Zajęcia bez kontaktu z nauczycielem:
1. praca z podręcznikiem: 90 h
2. praca wstępna i wykonanie 2 testów online: 10 h
3. opracowanie kilku etapów projektu: 40 h

Sumaryczna liczba godzin: 166

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

5

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 0h |
| Projekt:  | 30h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Znajomość podstaw programowania w języku C/C++, na poziomie obowiązkowego przedmiotu Programowanie.

**Limit liczby studentów:**

-

**Cel przedmiotu:**

Celem przedmiotu jest nauka zasad konstruowania algorytmów i doboru struktur danych, ze szczególnym uwzględnieniem dynamicznych listowych struktur danych.

**Treści kształcenia:**

Wprowadzenie: zagadnienia złożoności obliczeniowej algorytmów, notacja "duże O". Złożoność asymptotyczna, złożoność średnia i pesymistyczna.
Rekurencja. Realizacja wywołania rekurencyjnego, stos rekursji, warunek końca. Geometryczne przykłady ilustrujące zasadę rekurencji. Zagadnienia wydajności algorytmów rekurencyjnych.
Algorytmy sortowania: algorytmy proste (przez wybieranie, wstawianie, zamianę), sortowanie szybkie, sortowanie przez scalanie. Porównanie złożoności obliczeniowej.
Algorytmy przeszukiwania; przeszukiwanie danych: liniowe, binarne, z haszowaniem. Wyszukiwanie wzorca w tekście.
Listy jako przykład wykorzystania wskaźników i zmiennych dynamicznych. Zasady wykonywania operacji na listach: wstawianie i usuwanie elementów. Listy jednokierunkowe, dwukierunkowe i cykliczne.
Drzewa binarne i drzewa binarnego wyszukiwania: zasada definiowania, operacje wyszukiwania, wstawiania i usuwania elementów. Wykorzystanie drzew BST do sortowania danych.
Binarne drzewa prawie zrównoważone: drzewa AVL i drzewa czerwono-czarne. Operacje rotacji w procesie równoważenia drzew; zasady wstawiania i usuwania elementów.
Stosy i kolejki - implementowane w tablicach lub listach; kolejki priorytetowe jako implementacja sterty.
Grafy: reprezentacja macierzowa i listy sąsiedztwa. Najkrótsze ścieżki: metoda Floyda, algorytm Dijkstry. Minimalne drzewa rozpinające: algorytm Kruskala.
Algorytmy geometryczne (geometria obliczeniowa): poszukiwanie otoczki wypukłej, triangulacja Delaunaya. Struktura half-edge w reprezentacji brył.
Przegląd metod konstruowania algorytmów. Metody typu "dziel i zwyciężaj", programowanie dynamiczne, algorytmy zachłanne, algorytmy z powrotami, metody "zamiatania" płaszczyzny.
Kalkulator: przykład tworzenia rozbudowanego programu, od implementacji prostych działań poprzez operacje na macierzach aż do stworzenia rekurencyjnego parsera służącego do obsługi wyrażeń arytmetycznych z nawiasami i zmiennymi.

**Metody oceny:**

Zaliczenie przedmiotu odbywa się w języku C/C++ na podstawie sumy punktów uzyskanych z:
- dwu testów przeprowadzanych on-line (przez Internet); z każdego z nich można uzyskać maksymalnie 5 pkt. Testy odbywają się w ściśle określonych dniach, nie ma żadnej możliwości odrobienia ich w innym terminie.
- projektu realizowanego (jako aplikacja konsolowa) samodzielnie w ciągu semestru w kilku etapach, ograniczonych narzuconymi terminami - i zaliczanego podczas egzaminu.
- egzaminu pisemnego przeprowadzanego na uczelni.

UWAGA: wykonywanie testów on-line i projektu nie jest obowiązkowe, konieczny jest jedynie egzamin (cz. 1 i 2).

Egzamin trwa 120 minut i składa się z trzech części:
1. części testowej, trwającej 10 minut i zawierającej 15 pytań testowych (wybór jednej z 3 odpowiedzi).
2. części zadaniowej, trwającej 60 minut i wymagającej rozwiązania 2 zadań na papierze:
- zadanie polegające na napisaniu programu z zakresu list jednokierunkowych, czyli z zakresu lekcji 4.1-4.2, na poziomie zadań do lekcji 4;
- zadanie polegające na wykonaniu wraz z komentarzem rysunku ilustrującego działanie zadanego algorytmu (spośród kilkunastu podanych) na konkretnym przykładzie (z zakresu lekcji 1-8).
3. części projektowej, trwającej 50 minut i polegającej na zaliczaniu projektu przy komputerach (zaliczanie może wymagać umiejętności modyfikacji napisanego projektu).

**Egzamin:**

tak

**Literatura:**

1. Dawid Harel - Rzecz o istocie informatyki. Algorytmika. WNT, 2001.
2. Niklaus Wirth - Algorytmy+struktury danych=programy. WNT, 2002.
3. Piotr Wróblewski - Algorytmy, struktury danych i techniki programowania. Helion, 2010
4. Adam Drozdek - C++. Algorytmy i struktury danych. Helion, 2004.
5. R. Neapolitan, Kumarss Naimipour - Podstawy algorytmów z przykładami w C++ Helion, 2004.

**Witryna www przedmiotu:**

https://red.okno.pw.edu.pl/witryna/home.php, dostęp dla zalogowanych studentów

**Uwagi:**

-

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt [K\_W19]:**

ma uporządkowaną wiedzę ogólną obejmująca kluczowe zagadnienia z zakresu analizy i doboru algorytmów oraz technik programowania

Weryfikacja:

testy online, egzamin

**Powiązane efekty kierunkowe:** K\_W19

**Powiązane efekty obszarowe:** T1A\_W04

**Efekt [K\_W04] :**

ma szczegółową wiedzę z zakresu technik konstruowania algorytmów, ze szczególnym uwględnieniem dynamicznych struktur danych

Weryfikacja:

testy online, zaliczanie projektu, egzamin

**Powiązane efekty kierunkowe:** K\_W04

**Powiązane efekty obszarowe:** T1A\_W04, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt [K\_U15]:**

potrafi formułować zagadnienia w postaci algorytmicznej i zapisywać algorytmy w językach programowania

Weryfikacja:

zaliczanie zadań projektowych, egzamin

**Powiązane efekty kierunkowe:** K\_U15

**Powiązane efekty obszarowe:** T1A\_U14, T1A\_U15

**Efekt [K\_U20]:**

umie tworzyć proste konstrukcje i złożone algorytmy w sposób logiczny, zgodnie z regułami logiki matematycznej

Weryfikacja:

zadania projektowe (zaliczanie), egzamin

**Powiązane efekty kierunkowe:** K\_U20

**Powiązane efekty obszarowe:** T1A\_U09

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt [K\_K01]:**

ma nawyk ustawicznego kształcenia się i wyszukiwania nowych informacji (w podręczniku, w sieci) w zakresie konstruowania algorytmów

Weryfikacja:

konsultowanie i zaliczanie kilkustopniowego projektu

**Powiązane efekty kierunkowe:** K\_K01

**Powiązane efekty obszarowe:** T1A\_K01

**Efekt [K\_K06]:**

radzi sobie z rozwiązywaniem nowych, nietypowych zadań

Weryfikacja:

realizacja i zaliczanie projektu, testy online, egzamin

**Powiązane efekty kierunkowe:** K\_K06

**Powiązane efekty obszarowe:** T1A\_K06