**Nazwa przedmiotu:**

Algorytmy i struktury danych (I)

**Koordynator przedmiotu:**

Roman PODRAZA

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne - podstawowe

**Kod przedmiotu:**

AISDI

**Semestr nominalny:**

1 / rok ak. 2012/2013

**Liczba punktów ECTS:**

5

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

128

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

**Język prowadzenia zajęć:**

polski

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

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

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 15h |
| Laboratorium: | 15h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Znajomość programowanie w C++

**Limit liczby studentów:**

120

**Cel przedmiotu:**

Praktyczne zapoznanie studentów z wybranymi typowymi strukturami danych i przykładowymi algorytmami operującymi na tych strukturach oraz wykorzystaniem struktur danych w kontekście projektowania obiektowego programów komputerowych.

**Treści kształcenia:**

Treść wykładu:
1. Przegląd struktur danych: typ danych, system typów danych, obiekt, struktura danych, liniowe struktury danych, drzewiaste struktury danych, grafowe struktury danych, koszt i rząd kosztu przetwarzania struktur (2h).
2. Liniowe struktury danych: listy jednokierunkowe i dwukierunkowe, pierścienie, stosy, kolejki i kolejki priorytetowe z operacjami przeglądania struktury, wyszukiwania, wstawiania i usuwania elementu. Biblioteka STL w C++. (6h).
3. Algorytmy sortowania: Insertion Sort, Selection Sort, Bubble Sort, Quicksort, Merge Sort, Heap Sort, Straight Radix Sort, Radix Exchenge Sort, Shell Sort (6h).
4. Algorytmy wyszukiwania: wyszukiwanie liniowe, binarne, interpolacyjne, z podziałem Fibonacciego, zastosowania funkcji mieszającej (ang. hashing) (2h).
5. Algorytmy rekurencyjne, zasada „dziel i rządź” (2h).
6. Drzewa: drzewa binarne BST, Binary Tree Sort, drzewa o dowolnej liczbie następników, drzewa zrównoważone (AVL, czerwono-czarne, 2-3-4, B, BB) z operacjami przeglądania struktury, wyszukiwania, wstawiania i usuwania elementu (8h).
7. Grafy: metody implementacji grafów, przeglądanie w głąb (DFS), przeglądanie wszerz (BFS), algorytm badania osiągalności węzłów, wyznaczenie wszystkich ścieżek, wyszukiwanie najkrótszej ścieżki w grafie, algorytm Dijkstry (4h).
Zakres ćwiczeń:
1. Szablony (ang. templates) w C++.
2. Operacje na listach jednokierunkowych.
3. Definiowanie typów danych z wykorzystaniem struktur liniowych oraz implementacje wybranych operacji.
4. Projektowanie i implementacja algorytmów rekurencyjnych.
5. Sprawdzian – struktury liniowe i rekurencja.
6. Definiowanie typów danych z wykorzystaniem drzew oraz implementacje wybranych operacji.
7. Wybrane algorytmy grafów.
Zakres laboratorium:
1. Zajęcia organizacyjne – instrukcje dotyczące środowiska programowania.
2. Implementacja określonego typu danych (wykazu) przy pomocy tablicy.
3. Implementacja określonego typu danych (wykazu) przy pomocy listy jednokierunkowej.
4. Implementacja określonego typu danych (wykazu) przy pomocy pierścienia dwukierunkowego.
5. Implementacja określonego typu danych (wykazu) przy pomocy drzewa BST.
6. Algorytmy z zastosowaniem rekurencji.
7. Implementacja grafu z wybranymi algorytmami.

**Metody oceny:**

Ocena egzaminu
Ocena sprawdzianu
Ocena laboratorium

**Egzamin:**

tak

**Literatura:**

[1] L. Nyhoff “ADTs, Data Structures, and Problem Solving with C++”, Prentice-Hall, 2005.
[2] M. A. Weiss, “Data Structures and Problem Solving Using C++”, Second Edition, Addison Wesley Longman, Inc., 2000.
[3] T. Budd, “Data Structures in C++ Using the Standard Template Library”, Addison Wesley Longman, Inc., 1998.
[4] S. Sengupta, C. Ph. Korobkin, “C++ Object-Oriented Data Structures”, Springer-Verlag, 1994.
[5] P. Wróblewski, “Algorytmy, struktury danych i techniki programowania”, Helion, 1996.
[6] L. Banachowski, K. Diks, W. Rytter, “Algorytmy i struktury danych”, WNT, Warszawa, 1996.
[7] N. Wirth, “Algorytmy + struktury danych = programy”, WNT, Warszawa, 1980.
[8] B. Stroustrup, “The C++ Programming Language”, Second Edition, Addison Wesley, 1991.

**Witryna www przedmiotu:**

https://studia.elka.pw.edu.pl

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt AISDI\_W01:**

Student, który zaliczył przedmiot, posiada podstawową wiedzę na temat: Liniowych struktur danych: - przeglądania struktury - wyszukiwania, wstawiania i usuwania elementu - sortowania - wydajności stosowania

Weryfikacja:

Ocena egzaminu Ocena sprawdzianu Ocena Lab. 2.-4.

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**

**Efekt AISDI\_W02:**

Student, który zaliczył przedmiot, posiada podstawową wiedzę na temat: Podstawowych drzewiastych struktur danych: - przeglądania struktury - wyszukiwania, wstawiania i usuwania elementu - wydajności stosowania

Weryfikacja:

Ocena egzaminu Ocena Lab. 6.

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**

**Efekt AISDI\_W03:**

Student, który zaliczył przedmiot, posiada podstawową wiedzę na temat: Formułowania prostych algorytmów rekursywnych

Weryfikacja:

Ocena egzaminu Ocena Lab. 5.-6.

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**

**Efekt AISDI\_W04:**

Student, który zaliczył przedmiot, posiada podstawową wiedzę na temat: Struktur grafowych: - sposobów implementacji - przeglądania

Weryfikacja:

Ocena egzaminu Ocena Lab. 7.

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**

### Profil ogólnoakademicki - umiejętności

**Efekt AISDI\_U01:**

Student, który zaliczył przedmiot, potrafi zaimplementować w języku C++: Funkcje realizujące - operacje przeglądania struktur danych - operacje wyszukiwania elementu w podstawowych struktur danych - operacje wstawiania elementu do podstawowych struktur danych - operacje usuwania elementu z podstawowych struktur danych

Weryfikacja:

Ocena egzaminu Ocena sprawdzianu Ocena Lab. 2.-5.

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**

**Efekt AISDI\_U02:**

Student, który zaliczył przedmiot, potrafi zaimplementować w języku C++: Proste algorytmy rekursywne

Weryfikacja:

Ocena egzaminu Ocena sprawdzianu Ocena Lab. 6.

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**