

Programowanie komputerów 2 - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Programowanie komputerów 2
Kod przedmiotu	11.3-WK-liEP-PK2-W-S14_pNadGen3JVWA
Wydział	Wydział Matematyki, Informatyki i Ekonometrii
Kierunek	Informatyka i ekonometria
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. licencjata
Semestr rozpoczęcia	semestr zimowy 2017/2018

Informacje o przedmiocie	
Semestr	2
Liczba punktów ECTS do zdobycia	5
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	dr Florian Fabiś

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	30	2	-	-	Egzamin
Laboratorium	30	2	-	-	Zaliczenie na ocenę

Cel przedmiotu

Opanowanie umiejętności pisania programów z wykorzystaniem złożonych struktur danych. Umiejętność implementacji w programach algorytmów sortowania, selekcji i wyszukiwania. Znajomość w zakresie podstawowym problemu NP-zupełności (klasy P, NP, NP-zupełne). Zapoznanie studenta z podstawowymi pojęciami programowania zorientowanego obiektowego.

Wymagania wstępne

Programowanie komputerów 1.

Zakres tematyczny

Wykład

- Programowanie strukturalne w języku C++.** Struktury w C++. Zmienne dynamiczne: definicje, procedury tworzenia i usuwania zmiennych dynamicznych, operacje określone na elementach typu wskaźnikowego. Dynamiczne struktury danych: listy, stosy, kolejki. Podstawowe operacje na plikach w C++. Kontenery w C++
- Paradygmat programowania obiektowego.**
- Metody konstruowania efektywnych algorytmów.** Rekurencja, metoda „dziel i zwyciężaj”, algorytmy zachłanne.
- Podstawowe algorytmy wyszukiwania.** Wyszukiwanie liniowe, binarne, interpolacyjne. Samoorganizujące się pliki.
- Podstawowe algorytmy sortowania i selekcji.**
- Klasy złożoności obliczeniowej problemów.** Problemy decyzyjne i optymalizacyjne. Modele obliczeń: deterministyczna i niedeterministyczna maszyna Turinga. Klasy złożoności obliczeniowej problemów decyzyjnych: P, NP, NP-zupełne.

Laboratorium

- Pisanie i uruchamianie programów zawierających tablice struktur.
- Przetwarzanie plików.
- Stosowanie w programach zmiennych dynamicznych i tablic dynamicznych.
- Dynamiczne struktury danych (stosy, kolejki, listy).
- Pisanie programu będącego prostą bazą danych.

Metody kształcenia

Wykład: wykład problemowy.

Laboratorium: ćwiczenia laboratoryjne w pracowni komputerowej – pisanie i uruchamianie samo-dzielnie napisanych programów w C++, na zadane przez prowadzącego tematy, analiza tych programów i analiza stosowanych algorytmów. Nad niektórymi programami studenci będą pracowali w grupach 2–3 osobowych. Na ostatnich trzech ćwiczeniach każdy student pracuje nad samodzielnie zaprojektowaną aplikacją będącą prostą bazą danych, zawierającą wszystkie zdobyte na zajęciach umiejętności. Do aplikacji musi być dołączona dokumentacja zgodna z zadaną specyfikacją.

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbolne efektów	Metody weryfikacji	Forma zajęć
-------------	------------------	--------------------	-------------

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student zna podstawowe algorytmy sortowania i potrafi określić ich złożoność.	• K_W05	• egzamin - ustny, opisowy, testowy i inne	• Wykład
Student potrafi modelować fragment rzeczywistości przez definiowanie i korzystanie ze złożonych struktur danych.	• K_U28	• bieżąca kontrola na zajęciach • dyskusja	• Laboratorium
Student wie jak korzystać ze złożonych konstrukcji wybranego języka programowania do rozwiązywania zadań.	• K_W10	• egzamin - ustny, opisowy, testowy i inne • sprawdzian	• Wykład • Laboratorium
Student potrafi wykonywać podstawowe operacje na strukturach dynamicznych.	• K_U20	• bieżąca kontrola na zajęciach • sprawdzian	• Laboratorium
Student dobrze zna podstawowe pojęcia związane z programowaniem obiektowym.	• K_W10	• egzamin - ustny, opisowy, testowy i inne	• Wykład
Student potrafi tworzyć diagram klas.	• K_U25	• dyskusja • egzamin - ustny, opisowy, testowy i inne • sprawdzian	• Wykład • Laboratorium
Student potrafi definiować proste klasy i korzystać z nich.	• K_U28	• sprawdzian	• Laboratorium
Student potrafi zastosować złożone konstrukcje wybranego języka programowania do rozwiązywania zadań.	• K_U25	• egzamin - ustny, opisowy, testowy i inne • sprawdzian	• Wykład • Laboratorium
Student potrafi przedstawiać algorytmy za pomocą diagramów.	• K_U20	• egzamin - ustny, opisowy, testowy i inne • sprawdzian	• Wykład • Laboratorium

Warunki zaliczenia

Wykład. Egzamin weryfikujący efekty kształcenia w zakresie wiedzy i umiejętności. Egzamin składa się z dwóch części, pisemnej i ustnej. Warunkiem przystąpienia do części ustnej jest uzyskanie 30% punktów z części pisemnej. Uzyskanie 50% punktów z części pisemnej gwarantuje uzyskanie pozytywnej oceny.

Laboratorium. Ocena końcowa jest wystawiana na podstawie punktów uzyskanych na zajęciach. Punkty uzyskuje się za napisane na zajęciach sprawdziany, przedstawioną na ostatnich zajęciach aplikację, aktywność na zajęciach.

Na ocenę z przedmiotu składa się ocena z laboratorium (50%) oraz ocena z egzaminu (50%). Warunkiem przystąpienia do egzaminu jest pozytywna ocena z laboratorium. Warunkiem zaliczenia przedmiotu jest pozytywna ocena z egzaminu.

Literatura podstawowa

1. L. Banachowski, K. Diks, W. Rytter : Algorytmy i struktury danych, WNT, W-wa 1996.
2. J. Grebosz, Symfonia C++, Wydawnictwo Edition 2000, Kraków, 2010
3. A. Majczak, Programowanie strukturalne i obiektowe, Helion, Gliwice, 2010.
4. A. Stasiewicz, C++. Ćwiczenia praktyczne, Helion, Gliwice, 2008.

Literatura uzupełniająca

1. T.H. Cormen, Ch.E. Leiserson, R.L. Rivest: Wprowadzenie do algorytmów, WNT, 1997.
2. P. Wróblewski: Algorytmy, struktury danych i techniki programowania, wyd. II popr., Helion, 2001.

Uwagi

Zmodyfikowane przez dr Robert Dylewski, prof. UZ (ostatnia modyfikacja: 14-07-2017 13:27)

Wygenerowano automatycznie z systemu SylabUZ