

Programowanie poziomu systemu operacyjnego - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Programowanie poziomu systemu operacyjnego
Kod przedmiotu	11.3-WE-INF-D-ProgPozSystOper
Wydział	Wydział Nauk Inżynieryjno-Technicznych
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	drugiego stopnia z tyt. magistra inżyniera
Semestr rozpoczęcia	semestr zimowy 2021/2022

Informacje o przedmiocie	
Semestr	3
Liczba punktów ECTS do zdobycia	3
Typ przedmiotu	obieralny
Język nauczania	polski
Sylabus opracował	• dr inż. Grzegorz Łabiak

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
Laboratorium	30	2	18	1,2	Zaliczenie na ocenę

Cel przedmiotu

Celem przedmiotu jest praktyczna nauka programistycznych mechanizmów niskiego poziomu (funkcje API) udostępnianych przez system operacyjny Windows, typowymi dla programów działających na potrzeby systemu operacyjnego. Dodatkowo studenci zapoznawani są z takimi technikami programowania jak biblioteki statyczne (lib), dynamiczne (dll), OpenGL czy DirectX.

Wymagania wstępne

1. Podstawy programowania.

Zakres tematyczny

Architektura systemu operacyjnego Windows.

Programowy Interfejs Aplikacji – funkcje API.

Środowisko programu w warunkach systemu operacyjnego: aplikacja – zdarzenie – kolejka komunikatów.

Schemat programu w warunkach środowiska systemu operacyjnego: funkcja okienkowa, komunikat, pętla komunikatów.

Schemat obsługi komunikatu WM_PAINT, obszar roboczy okna, kontekst urządzenia graficznego.

Obiekty interfejsu urządzenia graficznego: pióro, pędzel, bitmapa, czcionka.

Zasoby. Tworzenie i posługiwanie się zasobami: menu, okna dialogowe, napisy, bitmapy.

Dynamiczne tworzenie menu i jego obsługa.

Tworzenie i obsługa własnych okien dialogowych.

Biblioteki statyczne (*.lib) i dynamiczne (*.dll).

Biblioteka OpenGL.

Technologia DirectX.

Metody kształcenia

wykład: wykład konwencjonalny

laboratorium: ćwiczenia laboratoryjne

projekt: metoda projektu

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Również zapoznawany jest z takimi technologiami niskopoziomowymi jak OpenGL i DirectX w stopniu umożliwiającym mu dalsze samokształcenie.	• K_U14	• bieżąca kontrola na zajęciach • sprawdzian	• Laboratorium
Student rozumie czym jest dla programisty system operacyjny, w szczególności system Windows, i jaka jest rola funkcji API.	• K_W09 • K_U14	• egzamin - ustny, opisowy, testowy i inne	• Laboratorium

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Poznaje rolę i zasady tworzenia i nabywa umiejętność tworzenia i wykorzystywania bibliotek statycznych (*.lib) i dynamicznych (*.dll).	<ul style="list-style-type: none"> • K_W09 • K_W11 • K_U14 	<ul style="list-style-type: none"> • bieżąca kontrola na zajęciach • sprawdzian 	<ul style="list-style-type: none"> • Laboratorium
Potrafi tworzyć proste aplikacje zbudowane z funkcji API w języku C/C++ na zasadach budowy oprogramowania systemowego (dla potrzeb funkcji systemu operacyjnego).	<ul style="list-style-type: none"> • K_U14 	<ul style="list-style-type: none"> • bieżąca kontrola na zajęciach 	<ul style="list-style-type: none"> • Laboratorium

Warunki zaliczenia

Wykład – warunkiem zaliczenia jest uzyskanie pozytywnej oceny z egzaminu przeprowadzonego w formie pisemnej lub ustnej.

Laboratorium – warunkiem zaliczenia jest uzyskanie pozytywnych ocen ze wszystkich ćwiczeń laboratoryjnych, przewidzianych do realizacji w ramach programu laboratorium.

Projekt – warunkiem zaliczenia jest uzyskanie pozytywnej oceny z realizacji zadania projektowego wskazanego przez prowadzącego zajęcia na początku semestru

Składowe oceny końcowej = wykład: 40% + laboratorium: 30% + projekt: 30%.

Literatura podstawowa

1. Charles Petzold, *Programowanie Windows*, Microsoft Press, 1999
2. Roland Waclawek, *Windows od kuchni*, Help, 1993
3. Wiktor Zychla, *Programowanie pod Windows*, wersja 0.99, Instytut Informatyki Uniwersytetu Wrocławskiego, Wrocław 2003

Literatura uzupełniająca

1. Wiktor Zychla, *Programowanie pod Windows. Zbiór zadań*, wersja 0.3, Instytut Informatyki Uniwersytetu Wrocławskiego, Wrocław 2006
2. Dave Shreiner, *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL(R), Version 3.0 and 3.1 (7th edition)*, Addison-Wesley, lipiec 2009
3. Robert Krupiński, *Aplikacje Direct3D*, Helion 2002
4. Jeffrey Richter, *Advanced Windows*, Microsoft Press, 1997

Uwagi

Zmodyfikowane przez prof. dr hab. inż. Andrzej Obuchowicz (ostatnia modyfikacja: 20-04-2021 08:48)

Wygenerowano automatycznie z systemu SyllabUZ