



KARTA PRZEDMIOTU

Kod przedmiotu	studia stacjonarne:	
	studia niestacjonarne:	
Nazwa przedmiotu	Programowanie aplikacji dla systemu Windows	
Nazwa przedmiotu w języku angielskim	Software programming for Windows	
Obowiązuje od roku akademickiego	2022/23	

USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	I stopień
Profil studiów	ogólnoakademicki
Forma i tryb prowadzenia studiów	studia stacjonarne i niestacjonarne
Zakres	wszystkie specjalności
Jednostka prowadząca przedmiot	Katedra Informatyki, Elektroniki i Elektrotechniki
Koordynator przedmiotu	dr hab. inż. Filip Rudziński, prof. PŚk
Zatwierdził	Dziekan Wydziału Elektrotechniki, Automatyki i Informatyki dr hab. inż. Roman Deniziak, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów		przedmiot kierunkowy
Status przedmiotu		wybieralny
Język prowadzenia zajęć		polski
Usytuowanie w planie studiów - semestr	studia stacjonarne	semestr VII
	studia niestacjonarne	semestr VIII
Wymagania wstępne		Podstawy programowania 1 i 2 Programowanie obiektowe 1
Egzamin (TAK/NIE)		NIE
Liczba punktów ECTS		6

Forma prowadzenia zajęć		wykład	ćwiczenia	laboratorium	projekt	inne
Liczba godzin w semestrze	studia stacjonarne:	30		30	15	
	studia niestacjonarne:	18		18	9	

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty uczenia się	Odniesienie do efektów kierunkowych
Wiedza	W01	Student zna i rozumie techniki programowania graficznych interfejsów użytkownika dla aplikacji systemu Windows, pojęcia komunikatów Windows, zdarzeń, wątków i ich synchronizacji z GUI.	INF1_W27
	W02	Student zna i rozumie techniki programowania aplikacji wielowątkowych dla systemu Windows, w tym bazodanowych, sieciowych, gier.	INF1_W27
	W03	Student zna i rozumie techniki programowania bibliotek DLL i wtyczek aplikacji oraz metod integracji bibliotek DLL z aplikacjami Windows.	INF1_W27
Umiejętności	U01	Student potrafi projektować elementy graficznego interfejsu użytkownika dla systemu Windows.	INF1_U27
	U02	Student potrafi zaprogramować aplikację wielowątkową dla systemu Windows, w tym bazodanową, sieciową, grę komputerową.	INF1_U27
	U03	Student potrafi zaprogramować bibliotekę DLL i wykorzystać ją w aplikacji Windows.	INF1_U27
Kompetencje społeczne	K01	Student jest gotów do uznania znaczenia wiedzy z obszaru programowania aplikacji dla systemu Windows oraz potrzeby ciągłego jej poszerzania	INF1_K01
	K02	Student jest gotów do krytycznej oceny posiadanych umiejętności w zakresie programowania aplikacji dla systemu Windows i rozumie potencjalne skutki swojej działalności na podstawie niewystarczających umiejętności	INF1_K02

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	1. Wprowadzenie do programowania aplikacji zdarzeniowych (komunikaty Windows, zdarzenia i procedury ich obsługi, pojęcie wątku głównego i wątków pobocznych programu oraz ich synchronizacja). 2. Projektowanie graficznego interfejsu użytkownika (GUI) (podstawowe kontrolki: przyciski, pola edycyjne, listy rozwijane, pola wyboru, itp., własności, metody i zdarzenia kontrolki, integracja GUI aplikacji z powłoką systemu, synchronizacja procedur zdarzeń z wątkami pobocznymi aplikacji). 3. Aplikacje bazodanowe typu klient-serwer (programowanie GUI aplikacji klienta, narzędzia dostępowe do baz danych (łącza TCP/IP), kontrolki do prezentacji wyników zapytań SQL, zdalne wykonywanie zapytań SQL i procedur składowanych baz danych, programowanie transakcji). 4. Programowanie gier komputerowych z wykorzystaniem grafiki rastrowej 2D (obsługa myszy i klawiatury w trybie asynchronicznym, programowanie animowanych obiektów, programowanie detekcji kolizji i dynamiki ruchu obiektów, projektowanie wielowątkowego silnika gry). 5. Aplikacje sieciowe (programowanie serwerów i klientów TCP/IP, programowanie serwerów podstawowych usług sieciowych: FTP, HTTP, POP3, SMTP, itp.). 6. Biblioteki DLL (łączenie statyczne i dynamiczne bibliotek z aplikacją, programowanie wtyczek, haki systemu operacyjnego: podmiana domyślnych procedur kontroli klawiatury, myszy, itp.). 7. Integracja z wybranymi zewnętrznymi bibliotekami (OpenGL, OpenCV, OpenCL, itp., programowanie zaawansowanych algorytmów przetwarzania obrazów, w tym systemów z obszaru inteligencji obliczeniowej do detekcji i rozpoznawania obiektów w obrazach). 8. Projektowanie własnych kontrolki GUI (tworzenie nowych własności typowych kontrolki, tworzenie nowych typów zdarzeń, podmiana domyślnych procedur rysowania kontrolki).
laboratorium	1. Projekt i implementacja graficznego interfejsu użytkownika aplikacji dla systemu Windows. 2. Projekt i implementacja aplikacji klienta relacyjnej bazy danych SQL. 3. Projekt i implementacja gry komputerowej z wykorzystaniem grafiki rastrowej 2D. 4. Projekt i implementacja wybranej usługi sieciowej (FTP, HTTP, POP3, SMTP, itp.). 5. Projekt i implementacja biblioteki DLL. Integracja wybranych bibliotek DLL z aplikacjami Windows.
projekt	1. Projekt i implementacja aplikacji dla systemu Windows z wykorzystaniem wybranych narzędzi informatycznych przedstawionych w ramach wykładu.

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01					X	
W02					X	
W03					X	
U01				X	X	
U02				X	X	
U03				X	X	
K01					X	
K02					X	

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów ze sprawozdań i projektu.
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów ze sprawozdań z wykonanych zadań laboratoryjnych.
projekt	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów z projektu.

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS												
Lp.	Rodzaj aktywności	Obciążenie studenta										Jednostka
		studia stacjonarne					studia niestacjonarne					
1.	Udział w zajęciach zgodnie z planem studiów	W	C	L	P	S	W	C	L	P	S	h
		30		30	15		18		18	9		
2.	Inne (konsultacje, egzamin)	2		2	1		2		2	1		h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	80					50					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	3,2					2					ECTS
5.	Liczba godzin samodzielnej pracy studenta	70					100					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	2,8					4					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	45					27					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,8					1,08					ECTS
9.	Sumaryczne obciążenie pracą studenta	150					150					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	6										ECTS

LITERATURA

1. Paweł Borkowski: "Tworzenie aplikacji dla Windows", Helion, 2008.
2. Andrzej Daniluk: "C++Builder. Kompendium programisty", Helion, 2003.