



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	E-ID2S-12-s2
Nazwa przedmiotu	Programowanie Imperatywne, Obiektowe i Deklaracyjne
Nazwa przedmiotu w języku angielskim	Imperative, Object and Declarative programming
Obowiązuje od roku akademickiego	2019/20

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	II stopień
Profil studiów	ogólnoakademicki
Forma i tryb prowadzenia studiów	Stacjonarne
Zakres	Systemy informatyczne
Jednostka prowadząca przedmiot	Katedra Zastosowań Informatyki
Koordynator przedmiotu	prof., dr hab. inż. Volodymyr Ovsyak
Zatwierdził	Dziekan Wydziału Elektrotechniki Automatyki i Informatyki Dr hab. inż. Antoni Różowicz, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	kierunkowy
Status przedmiotu	nieobowiązkowy
Język prowadzenia zajęć	Polski
Usytuowanie modułu w planie studiów - semestr	II
Wymagania wstępne	Studia I-ego stopnia
Egzamin (TAK/NIE)	NIE
Liczba punktów ECTS	3

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	Inne
Liczba godzin w semestrze	30	0	0	15	0

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Ma podstawową wiedzę języka imperatywnego paradygmatu programowania mikroprocesorów.	NF2_W01
	W02	Ma podstawową wiedzę języka obiektowego programowania C# i języka markowania XAML Windows Presentation Foundation.	INF2_W02
Umiejętności	U01	Potrafi stworzyć oprogramowanie systemu informatycznego stosując język imperatywnego paradygmatu programowania.	INF2_U1
	U02	Potrafi stworzyć oprogramowanie systemu informatycznego wykorzystaniem języka obiektowego programowania C# platformy Microsoft Visual Studio .NET.	INF2_U19
	U03	Potrafi zaprogramować interfejs użytkownika systemu informatycznego i podstawowe elementy grafiki komputerowej wykorzystaniem języka markowania XAML.	INF2_U20
Kompetencje społeczne	K01	Zna podstawy składni i semantyki języków Asemblera mikroprocesorów ARM, C# i XAML platformy Microsoft Visual Studio .NET.	INF2_K04

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	1. Paradygmaty programowania. Paradygmaty imperatywnego, obiektowego i deklaratywnego programowania.
	2. Model architektury mikroprocesorów ARM. Rejestry, ALU, dekodery instrukcji, magistrale i pamięć.
	3. Model programowania. Typy danych. Tryby funkcjonowania mikroprocesora. Rejestry. Rejestry stanu programu. Tabela wektorów.
	4. Składnia kodu systemu informatycznego w języku Asemblera. Struktura modułów języka Asemblera. Nazwy rejestrów. Dyrektywy. Makra. Operacje. Literały.
	5. Stałe, pole literałów i operacje. Ładowanie stałych do rejestrów. Ładowanie adresów do rejestrów. Operacje logiczne i arytmetyczne.
	6. Pętle, skoki i tablice. Pętle while. Pętle for. Pętle do ... while. Znaczniki. Wykonanie warunkowe. Tablice.
	7. Składnia języka C#. Identyfikatory i słowa kluczowe. Literały, znaki interpunkcyjne. Komentarze.
	8. System typów danych. Typy predefiniowane. Typy własne. Składowe statyczne i składowe instancji. Konwersje typów. Typy wartościowe i typy referencyjne. Zmienne i parametry.
	9. Operatory. Metody. Klasy. Operatory logiczne i arytmetyczne. Metody statyczne i dynamiczne, wirtualne i abstrakcyjne. Parametry metod. Klasy częściowe i statyczne.
	10. Definicja XAML. Elementy i atrybuty. Przestrzeń nazw.
	11. Elementy właściwości. Konwertery typów. Rozszerzenie znaczników.
	12. Elementy potomne obiektów. Łączenie XAML z kodem obiektowym. Słowa kluczowe XAML.
projekt	Przedmiotem projektu jest stworzenie kodu systemu informatycznego zadanych graficzno-analitycznych wyrazów algorytmów w oparciu na wykorzystanie nowoczesnego języka obiektowego programowania C#, języka markowania i środowiska Windows Presentation Foundation.

*) zostawić tylko realizowane formy zajęć

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

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

A.**FORMA I WARUNKI ZALICZENIA**

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład	zaliczenie z oceną	Wiedza i umiejętności teoretyczne
projekt	zaliczenie z oceną	Sprawozdanie i kod działającego systemu

*) zostawić tylko realizowane formy zajęć

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
1.	Udział w zajęciach zgodnie z planem studiów	W	C	L	P	S	h
		30			15		
2.	Inne (konsultacje, egzamin)*	4			2		h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	51					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	2,04					ECTS
5.	Liczba godzin samodzielnej pracy studenta	24					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	0,96					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	15					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	0,88					ECTS
9.	Sumaryczne godzinowe obciążenie pracą studenta	75					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	3					

* wszelkie formy weryfikacji efektów, w tym egzaminy oraz nie więcej niż 2 godziny konsultacji dla każdej formy zajęć

LITERATURA

1. J.Albachani, B. Albachani. C# 6.0. Leksykon kieszonkowy. – Gliwice: Wydawnictwo "Helion", 2017.
2. A.Nathan. WPF 4.5. Księga eksperta. – Gliwice: Wydawnictwo "Helion", 2015.
3. J.Matulewski. NVVM i XAML w Visual Studio 2015. – Gliwice: Wydawnictwo "Helion", 2016.
4. R.Osherowe. Testy jednostkowe. Świat niezawodowych aplikacji. – Gliwice: Wydawnictwo "Helion", 2014.
5. J.Matulewski. NVVM i XAML w Visual Studio 2015. – Gliwice: Wydawnictwo "Helion", 2016.
6. W. Hohl. Asembler dla mikroprocesorów ARM. – Gliwice: Wydawnictwo "Helion", 2014.
7. A.Kempa. Wprowadzenie do WPF. Tworzenie aplikacji w WPF przy użyciu XAML i C#. – Gliwice: Wydawnictwo "Helion", 2017.

Uwaga: wykaz literatury winien uwzględniać aktualne i dostępne publikacje