



KARTA MODUŁU / KARTA PRZEDMIOTU

Kod modułu	E-2IZ2-1004-s3
Nazwa modułu	Programowanie Imperatywne, Obiektowe i Deklaracyjne
Nazwa modułu w języku angielskim	Imperative, Object and Declarative programming
Obowiązuje od roku akademickiego	2013/2014 (aktualizacja 2017/2018)

A. USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	II stopień (I stopień / II stopień)
Profil studiów	ogólnoakademicki (ogólno akademicki / praktyczny)
Forma i tryb prowadzenia studiów	Niestacjonarne (stacjonarne / niestacjonarne)
Specjalność	Systemy informatyczne
Jednostka prowadząca moduł	Katedra Systemów Informatycznych Zakład Zastosowań Informatyki
Koordynator modułu	prof., dr hab. inż. Volodymyr Ovsyak
Zatwierdził:	Dziekan WEAiI Dr hab. inż. Antoni Różowicz, prof. PŚk

B. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	kierunkowy (podstawowy / kierunkowy / inny HES)
Status modułu	nieobowiązkowy (obowiązkowy / nieobowiązkowy)
Język prowadzenia zajęć	polski
Usytuowanie modułu w planie studiów - semestr	II
Usytuowanie realizacji przedmiotu w roku akademickim	Semestr zimowy (semestr zimowy / letni)
Wymagania wstępne	Studia I-ego stopnia (kody modułów / nazwy modułów)
Egzamin	nie (tak / nie)
Liczba punktów ECTS	4

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	inne
w semestrze	18			18	



C. EFEKTY KSZTAŁCENIA I METODY SPRAWDZANIA EFEKTÓW KSZTAŁCENIA

Cel modułu	Otrzymanie wiedzy i umiejętności wykonania programowania systemów informatycznych językami imperatywnego, obiektowego i deklaratywnego paradygmatów programowania.
-------------------	--

Symbol efektu	Efekty kształcenia	Forma prowadzenia zajęć (w/ć/l/p/inne)	odniesienie do efektów kierunkowych	odniesienie do efektów obszarowych
W_01	Ma podstawową wiedzę języka imperatywnego paradygmatu programowania mikroprocesorów.	W/L	K_W01	T2A_W01, T2A_W02
W_02	Ma podstawową wiedzę języka obiektowego programowania C# i języka markowania XAML Windows Presentation Foundation.	W/I/P	K_W02	T2A_W03 T2A_W07
U_01	Potrafi stworzyć oprogramowanie systemu informatycznego stosując język imperatywnego paradygmatu programowania.	L/P	K_U04	T2A_U04
U_02	Potrafi stworzyć oprogramowanie systemu informatycznego wykorzystaniem języka obiektowego programowania C# platformy Microsoft Visual Studio .NET.	L/P	K_U08	T2A_U09
U_03	Potrafi zaprogramować interfejs użytkownika systemu informatycznego i podstawowe elementy grafiki komputerowej wykorzystaniem języka markowania XAML.	L/P	K_U11	T2A_U12,T2A_U18
K_01	Zna podstawy składni i semantyki języków Asemblera mikroprocesorów ARM, C# i XAML platformy Microsoft Visual Studio .NET.	P	K_K02	T2A_K03

Treści kształcenia:

5. Treści kształcenia w zakresie wykładu

Nr wykładu	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1	Paradygmaty programowania. Paradygmaty imperatywnego, obiektowego i deklaratywnego programowania.	W_01
2	Model architektury mikroprocesorów ARM. Rejestry, ALU, dekodery instrukcji, magistrale i pamięć.	W_01, U_01
3	Model programowania. Typy danych. Tryby funkcjonowania mikroprocesora. Rejestry. Rejestry stanu programu. Tabela wektorów.	W_02, U_01, U_02, U_03
4	Składnia kodu systemu informatycznego w języku Asemblera. Struktura modułów języka Asemblera. Nazwy rejestrów. Dyrektywy. Makra. Operacje. Literały.	W_02, U_01, U_02, U_03
5	Stałe, pole literałów i operacje. Ładowanie stałych do rejestrów. Ładowanie adresów do rejestrów. Operacje logiczne i arytmetyczne.	W_02, U_01, U_02, U_03
6	Pętle, skoki i tablice. Pętla while. Pętla for. Pętla do ... while. Znaczniki. Wykonanie warunkowe. Tablice.	W_02, U_02, U_03
7	Składnia języka C#. Identyfikatory i słowa kluczowe. Literały, znaki interpunkcyjne. Komentarze.	W_02, U_02, U_03
8	System typów danych. Typy zdefiniowane. Typy własne. Składowe statyczne i składowe instancje. Konwersje typów. Typy wartościowe i typy referencyjne. Zmienne i parametry.	W_02, U_02, U_03
9	Operatory. Metody. Klasy. Operatory logiczne i arytmetyczne. Metody statyczne i dynamiczne, wirtualne i abstrakcyjne. Parametry metod. Klasy	W_02, U_02, U_03



	częściowy i statyczne.	
10	Definicja XAML. Elementy i atrybuty. Przestrzeń nazw.	W_02, U_02, U_03
11	Elementy właściwości. Konwertery typów. Rozszerzenie znaczników.	W_02, U_02, U_03
12	Elementy potomne obiektów. Łączenie XAML z kodem obiekowym. Słowa kluczowe XAML.	U_03, K_01

3. Charakterystyka zadań projektowych

Treść projektu.

Przedmiotem projektu jest stworzenie kodu systemu informatycznego zadanych graficzno-analitycznych wyrazów algebry algorytmów w oparciu na wykorzystanie nowoczesnego języka obiektowego programowania C#, języka markowania i środowiska Windows Presentation Foundation. Kompletny i działający kod systemu informatycznego będzie stanowił podstawę zaliczenia projektu. Projekty zaliczeniowe wykonywane indywidualnie. Przedmiot i zakres projektu musi być uzgodniony z prowadzącym zajęcia.

Projekt ma obejmować oprogramowanie funkcjonalnej części systemu informatycznego napisane w języku obiektowego programowania C# platformy Microsoft Visual Studio .NET. Interfejs użytkownika systemu i graficzno-analityczny wyrazy algebry algorytmów zawiera oprogramowanie napisane w języku deklaratywnego programowania XAML.

Projekt powinien być realizowany w ramach planowanych zajęć (systematyczność pracy studenta będzie elementem oceny projektu)..

EFEKTY: U_01, U_02, U_03, K_01

Metody sprawdzania efektów kształcenia

Symbol efektu	Metody sprawdzania efektów kształcenia (sposób sprawdzenia, w tym dla umiejętności – odwołanie do konkretnych zadań projektowych, laboratoryjnych, itp.)
W_01	test wiedzy sprawdzający znajomość składni i semantyki języka Asemblera mikroprocesorów ARM.
W_02	test wiedzy sprawdzający znajomość składni i semantyki języka obiektowego programowania C# i języka markowania XAML.
U_01	test sprawdzający umiejętność wykorzystania języka Asemblera mikroprocesorów ARM.
U_02	test sprawdzający umiejętność wykorzystania składni i semantyki języka obiektowego programowania C#.
U_03	test sprawdzający umiejętność samodzielnego i twórczego wykorzystania składni i semantyki języka markowania XAML.
K_01	test na wykorzystanie platformy Microsoft Visual Studio .NET i środowiska Windows Presentation Foundation w praktyce.



D. NAKŁAD PRACY STUDENTA

Bilans punktów ECTS		
	Rodzaj aktywności	obciążenie studenta
1	Udział w wykładach	18
2	Udział w ćwiczeniach	
3	Udział w laboratoriach	
4	Udział w konsultacjach (2-3 razy w semestrze)	
5	Udział w zajęciach projektowych	18
6	Konsultacje projektowe	4
7	Udział w egzaminie	
8		
9	Liczba godzin realizowanych przy bezpośrednim udziale nauczyciela akademickiego	40 (suma)
10	Liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczyciela akademickiego (1 punkt ECTS=25-30 godzin obciążenia studenta)	1,54
11	Samodzielne studiowanie tematyki wykładów	30
12	Samodzielne przygotowanie się do ćwiczeń	
13	Samodzielne przygotowanie się do kolokwium	
14	Samodzielne przygotowanie się do laboratoriów	
15	Wykonanie sprawozdań	
15	Przygotowanie do kolokwium końcowego z laboratorium	
17	Wykonanie projektu	34
18	Przygotowanie do egzaminu	
19		
20	Liczba godzin samodzielnej pracy studenta	64 (suma)
21	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy (1 punkt ECTS=25-30 godzin obciążenia studenta)	2,46
22	Sumaryczne obciążenie pracą studenta	104
23	Punkty ECTS za moduł 1 punkt ECTS=25-30 godzin obciążenia studenta	4
24	Nakład pracy związany z zajęciami o charakterze praktycznym Suma godzin związanych z zajęciami praktycznymi	56
25	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym punkt ECTS=25-30 godzin obciążenia studenta	2,15

E. LITERATURA

Wykaz literatury	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.
Witryna WWW modułu/przedmiotu	