



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	M#1-N2-AiR-AP-109
Nazwa przedmiotu	Programowanie obiektowe
Nazwa przedmiotu w języku angielskim	Object-Oriented Programming
Obowiązuje od roku akademickiego	2019/2020

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	AUTOMATYKA i ROBOTYKA
Poziom kształcenia	II stopień
Profil studiów	ogólnoakademicki
Forma i tryb prowadzenia studiów	studia niestacjonarne
Zakres	automatyka przemysłowa
Jednostka prowadząca przedmiot	Katedra Automatyki i Robotyki
Koordinator przedmiotu	Prof. dr hab. inż. Dariusz Janecki
Zatwierdził	

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	przedmiot specjalnościowy
Status przedmiotu	obowiązkowy
Język prowadzenia zajęć	polski
Usytuowanie modułu w planie studiów - semestr	semestr 1
Wymagania wstępne	podstawy informatyki, programowanie w C
Egzamin (TAK/NIE)	NIE
Liczba punktów ECTS	2

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	seminarium
Liczba godzin w semestrze	9		18		

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Student zna metody i techniki programowania obiektowego w tym specyficzne dla podejścia obiektowego takie jak hermetyzacja, dziedziczenie, polimorfizm	AiR2_W11
	W02	Student zna środowisko graficzne stosowane w aplikacjach obiektowych do tworzenia wykresów i interfejsów użytkownika	AiR2_W11
Umiejętności	U01	Student potrafi napisać program zorientowany obiektowo np. w języku C++/C# symulujący działanie prostego systemu i urządzenia technicznego	AiR2_U05 AiR2_U07
	U02	Student potrafi przy formułowaniu i rozwiązywaniu zadań projektowych dostrzegać ich aspekty ekonomiczne systemowe i inne pozatechniczne.	AiR2_U07
Kompetencje społeczne	K01	Student ma świadomość: odpowiedzialności za pracę projektowe w obszarze automatyzacji procesów, podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.	AiR2_K03 AiR2_K04

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	1. Podstawowe konstrukcje i mechanizmy programowania obiektowego: klasa (atrybuty, metody, komunikaty), obiekt (pola obiektu, metody obiektu), hermetyzacja i ukrywanie danych, dziedziczenie, polimorfizm.
	2. Grafika w technologii zorientowanej obiektowo: środowiska graficzne, rysowanie wykresów, budowanie interfejsu użytkownika
	3. Elementy zaawansowanego programowania obiektowego - interfejsy i klasy abstrakcyjne, projektowanie konstruktorów i destruktorów, projektowanie mechanizmów obsługi błędów, serializacja obiektów.
laboratorium	1. Grafika w technologii zorientowanej obiektowo: budowanie interfejsu użytkownika.
	2. Metody manipulacji obiektami graficznymi.
	3. Wizualizacja danych, tworzenie wykresów i tabel
	4. Przetwarzanie i przechowywanie danych – serializacja
	5. Komunikacja z urządzeniami za pomocą portów szeregowych, sieci LAN
	5. Modele symulacyjne prostych systemów i urządzeń technicznych w języku C++/C#

*) zostawić tylko realizowane formy zajęć

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów kształcenia (zaznaczyć X)					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			x			
W02			x			
U01				x		
U02				x		
K01						x

FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów z kolokwium
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów z zadań programistycznych

*) zostawić tylko realizowane formy zajęć

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
		W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	9		18			h
2.	Inne (konsultacje, egzamin)	2		2			h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	31					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	1,2					ECTS
5.	Liczba godzin samodzielnej pracy studenta	19					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	0,8					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	33					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,3					ECTS
9.	Sumaryczne obciążenie pracą studenta	50					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	2					ECTS

LITERATURA

1. A.Kempa. Wprowadzenie do WPF. Tworzenie aplikacji w WPF przy użyciu XAML i C#. – Gliwice: Wydawnictwo "Helion", 2017.
2. J.Albachani, B. Albachani. C# 6.0. Leksykon kieszonkowy. – Gliwice: Wydawnictwo "Helion", 2017.
3. J.Matulewski. NVVM i XAML w Visual Studio 2015. – Gliwice: Wydawnictwo "Helion", 2016.
4. A.Nathan. WPF 4.5. Księga eksperta. – Gliwice: Wydawnictwo "Helion", 2015.