

--	--	--

<b>1. Nazwa przedmiotu: WIRTUALNE INSTRUMENTY I SYSTEMY</b>		<b>2. Kod przedmiotu:</b>		
<b>3. Karta przedmiotu ważna od roku akademickiego: 2012</b>				
<b>4. Forma kształcenia:</b> studia pierwszego stopnia				
<b>5. Forma studiów:</b> studia stacjonarne				
<b>6. Kierunek studiów:</b> BIOTECHNOLOGIA; WYDZIAŁ AEII				
<b>7. Profil studiów:</b> ogólnoakademicki				
<b>8. Specjalność:</b> BIOINFORMATYKA				
<b>9. Semestr:</b> 5, 6				
<b>10. Jednostka prowadząca przedmiot:</b> Instytut Automatyki, RAu1				
<b>11. Prowadzący przedmiot:</b> dr inż. Witold Nocoń				
<b>12. Przynależność do grupy przedmiotów:</b> przedmioty wspólne				
<b>13. Status przedmiotu:</b> obowiązkowy				
<b>14. Język prowadzenia zajęć:</b> polski				
<b>15. Przedmioty wprowadzające oraz wymagania wstępne:</b> Informatyka. Dobre zrozumienie podstawowych zasad programowania w dowolnym języku tekstowym. Zrozumienie działania oraz umiejętność zastosowania struktur programowych (pętle, instrukcje warunkowe), używanie tablic, złożonych struktur danych, tworzenie i używanie podprogramów.				
<b>16. Cel przedmiotu:</b> Celem przedmiotu jest zdobycie praktycznej wiedzy i umiejętności projektowania, implementacji i testowania zaawansowanych aplikacji do automatyzacji, sterowania, monitorowania, symulacji i modelowania. Szczególny nacisk kładziony jest na umiejętność stosowania wzorców projektowych do tworzenia skalowalnych, rozszerzalnych, łatwych w interpretacji i walidacji aplikacji. Środowiskiem programowym wykorzystywanym na zajęciach jest LabVIEW.				
<b>17. Efekty kształcenia:<sup>1</sup></b>				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
W1	Zna podstawowe wzorce projektowe (maszyna stanów, kolejkowana maszyna stanów) oraz mechanizmy synchronizacji (kolejki, zdarzenia itp.)	SP	WM	K_W20 K_W21
W2	Zna wielowątkowe wzorce projektowe (master/slave, producent/konsument) oraz podstawy programowania obiektowego.	SP	WM	K_W21
U1	Potrafi zaimplementować podstawowe wzorce projektowe.	CL, PS	L	K_U09 K_U11 K_U26
U2	Potrafi zaimplementować hierarchię klas dla zadanego problemu oraz używać obiektów tych klas w aplikacji.	CL, PS	L	K_U09 K_U26
K1	Potrafi wybrać do zadanego problemu najbardziej odpowiadający mu wzorzec projektowy oraz zaimplementować go.	CL, PS	L	K_K03
K2	Ma świadomość ograniczeń stosowanych wzorców projektowych oraz potrafi proponować rozszerzenia tych wzorców.	PS, OS	L	K_K04
K3	Samodzielnie potrafi określać potrzebę zastosowania	PS, OS	L	K_K07

<sup>1</sup> należy wskazać ok. 5 – 8 efektów kształcenia

	obiektywnych rozwiązań programowych dla zadanych problemów.			
<b>18. Formy zajęć dydaktycznych i ich wymiar (liczba godzin)</b>				
W. : 30 L.: 15				
<b>19. Treści kształcenia:</b>				
<b>Wykład</b>				
1. <b>Podstawy programowani w LabVIEW.</b> Panel frontowy i diagram. Kontrolki i wyświetlacze. Terminale kontrolki i wyświetlaczy. Przepływ danych i łączenie elementów diagramu. Wyświetlanie danych an wykresach. Właściwości obiektów.				
2. <b>Sterowanie przebiegiem programu 1.</b> Pętla while. Struktura case. Rejestr przesuwny. Obliczenia iteracyjne.				
3. <b>Sterowanie przebiegiem programu 2.</b> Pętla for. Struktura sekwencji. Sterowanie kolejnością wykonania kodu.				
4. <b>Złożone struktury danych.</b> Tworzenie, użycie i wyświetlanie tablic. Tworzenie, użycie i wyświetlanie klastrów. Tworzenie wykresów.				
5. <b>Zmienne lokalne i globalne.</b> Tworzenie i używanie zmiennych lokalnych/globalnych. Działanie mechaniczne przycisków.				
6. <b>Podprogramy.</b> Tworzenie i wykorzystanie podprogramów SubVI. Własności podprogramów. Sposoby wykonania podprogramów. Dynamiczne wykonywanie podprogramów.				
7. <b>Programowanie zdarzeniowe.</b> Użycie struktury zdarzeń. Zdarzenia wywoływane programowo.				
8. <b>Łańcuchy, pliki, obsługa błędów, projekty.</b> Zmienne łańcuchowe. Pliki tekstowe i binarne. Zaawansowane funkcje obsługi plików. Funkcje arkuszy danych. Pliki z danymi pomiarowymi. Klaster błędu. Propagacja błędów. Tworzenie i używanie projektów. Tworzenie plików wykonywalnych (exe). Środowisko uruchomieniowe LabVIEW. Tworzenie wersji instalacyjnych aplikacji.				
9. <b>Definicja typu, proste wzorce projektowe.</b> Użycie i zastosowanie definicji typu. Podstawowe wzorce projektowe.				
10. <b>Standardowa maszyna stanów, funkcje synchronizacji.</b> Typ enumeratywny. Użycie wzorca standardowej maszyny stanów. Ograniczenia standardowej maszyny stanów. Funkcje synchronizacji.				
11. <b>Kolejkowana maszyna stanów, Wzorzec dynamiczny.</b> Użycie wzorca kolejkowanej maszyny stanów. Ograniczenia kolejkowanej maszyny stanów. Wzorzec dynamiczny. Typ variant.				
12. <b>Funkcje komunikacyjne.</b> Zmienne dzielone, <b>protokoły</b> TCP, UDP, POP, SMTP itp.				
13. <b>Przykłady modułów rozszerzających:</b> Datalogging and Supervisory Control Module, PID Module, Image Acquisition and Image Processing.				
14. <b>Przykłady praktycznych zastosowań.</b> Praktyczne tworzenie rzeczywistych aplikacji na przykładzie oprogramowania sterującego pilotażową oczyszczalnią ścieków oraz aplikacji komunikującej się z i monitorującej pracę elektrycznego samochodu podczas wyścigu.				
15. <b>Programowanie obiektowe.</b> Podstawy programowania obiektowego w LabVIEW. Tworzenie hierarchii klas (dziedziczenie). Polimorfizm i wykorzystanie metod wirtualnych.				
<b>Zajęcia laboratoryjne</b>				
1. Podstawy programowania w LabVIEW 1. Użycie podstawowych struktur programistycznych, kontrola wykonania programu, użycie zmiennych liczbowych, logicznych, łańcuchowych, obliczenia iteracyjne.				
2. Podstawy programowania w LabVIEW 2. Użycie złożonych struktur danych (tablice, klastry), tworzenie i wykorzystanie podprogramów. Programowanie zdarzeniowe.				
3. Aplikacja akwizycji danych. Tworzenie aplikacji umożliwiającej akwizycję danych pomiarowych z wykorzystaniem układu I/O, prezentację tych danych oraz archiwizację tych danych.				
4. Aplikacja sieciowa. Tworzenie aplikacji wykorzystującej protokół TCP/IP do komunikacji w sieci.				
5. Aplikacja układu sterowania. Tworzenie aplikacji realizującej układ sterowania z wykorzystaniem sterownika programowanego.				
<b>20. Egzamin:</b> nie.				

**21. Literatura podstawowa:**

1. Jeffrey Travis, Jim Kring, Labview for Everyone: Graphical Programming Made Easy and Fun, Prentice Hall, 2006.
2. Peter Blume A., The Labview Style Book (National Instruments Virtual Instrumentation), Prentice Hall, 2007

**22. Literatura uzupełniająca:**

1. Bitter R., Mohiuddin T., nawrocki M.: LabVIEW – advanced programming techniques. CRC Press, Taylor & Francis Group, 2007.

**23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia**

Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1	Wykład	30/15
2	Ćwiczenia	0/0
3	Laboratorium	15/30
4	Projekt	0/0
5	Seminarium	0/0
6	Inne	15/15 (5/5)
	Suma godzin	60/60 (50/50)

**24. Suma wszystkich godzin: 120 (100)****25. Liczba punktów ECTS:<sup>2</sup> 4****26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego: 2****27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty): 1 (3)****26. Uwagi:**

Zatwierdzono:

.....  
(data i podpis prowadzącego).....  
(data i podpis dyrektora instytutu/kierownika katedry/  
Dyrektora Kolegium Języków Obcych/kierownika lub  
dyrektora jednostki międzywydziałowej)<sup>2</sup> 1 punkt ECTS – 30 godzin.