

KARTA MODUŁU

I. OGÓLNE INFORMACJE O MODULE						
PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA IM. WITELONA W LEGNICY WYDZIAŁ NAUK TECHNICZNYCH I EKONOMICZNYCH						
Kierunek studiów:	INFORMATYKA					
Poziom studiów:	studia pierwszego stopnia					
Profil studiów:	praktyczny					
Forma studiów:	stacjonarne/niestacjonarne					
Nazwa modułu:	Zaawansowane metody programowania					
Rodzaj modułu:	Obowiązkowy					
Język wykładowy:	Język polski*					
Rok studiów:	2	Formy prowadzenia zajęć wraz z liczbą godzin dydaktycznych:				
Semestr:	6	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
Liczba punktów ECTS ogółem:	3	30/24	-	30/16	-	-
Forma zaliczenia:	Egzamin					
Wymagania wstępne:	Ukończone kursy Programowanie i projektowanie obiektowe I oraz Programowanie i projektowanie obiektowe II Umiejętność pracy zespołowej, rozumienie potrzeby posiadania kompetencji społecznych w pracy programisty Znajomość diagramów UML					
II. CELE KSZTAŁCENIA						
Cele kształcenia:						
Cel 1: Przedstawić teorię oraz przykłady wykorzystania dobrych praktyk programistycznych Cel 2: Przedstawić teorię oraz przykłady wykorzystania wzorców projektowych Cel 3: Przedstawić teorię oraz przykłady wykorzystania paradygmatów programowania Cel 4: Przedstawić praktyczne podejście do zespołowej pracy projektowej						
III. EFEKTY UCZENIA SIĘ WRAZ Z ODNIESIENIEM DO EFEKTÓW KIERUNKOWYCH ORAZ METODY WERYFIKACJI EFEKTÓW						
Efekt	Student, który zaliczył moduł w zakresie:	Odniesienie do efektów kierunkowych	Metody weryfikacji			
wiedzy:						
W1	Rozumie potrzebę wytwarzania oprogramowania zgodnie z zasadami czystego kodu	K1I_W04	Egzamin			
W2	Potrafi opisać i wskazać zastosowanie wzorców projektowych	K1I_W04	Egzamin			
umiejętności:						
U1	Umie dobrać odpowiednią technologię do wybranej klasy problemu	K1I_U01	Egzamin + praca projektowa			
U2	Potrafi ocenić kod pod względem wykorzystania dobrych praktyk programistycznych	K1I_U01	Egzamin + praca projektowa			
U3	Potrafi wskazać odpowiedzialności wskazanych klas i metod oraz stwierdzić czy łamana jest zasada pojedynczej odpowiedzialności	K1I_U08	Egzamin + praca projektowa			
U4	Potrafi wykorzystać branżową nomenklaturę do opisanego problemu w kodzie	K1I_U16	Egzamin + praca projektowa			
U5	Potrafi zaproponować elastyczną architekturę kodu opartą o polimorfizm i abstrakcję	K1I_U03	Egzamin + praca projektowa			
U6	Potrafi wykorzystać programowanie ekstremalne	K1I_U08	Egzamin + praca projektowa			
kompetencji społecznych:						
IV. TREŚCI PROGRAMOWE						

Treści programowe (tematyka zajęć, zaprezentowana z podziałem na poszczególne formy zajęć z określeniem liczby godzin potrzebnych na ich realizację)		
**		
Kod	Tematyka zajęć	Liczba godzin S/N
w1	Wprowadzenie do zaawansowanych metod programowania; sposoby doboru technologii projektowej	2/2
w2	Dobre praktyki programistyczne: zasady SOLID	2/2
w3	Dobre praktyki programistyczne: KISS, DRY, YAGNI i inne	2/2
w4	Dobre praktyki programistyczne: Czysty kod	6/3
w5	Praktyczny polimorfizm	2/2
w6	Programowanie asynchroniczne	2/2
w7	Wzorce projektowe: kreatywne, strukturalne, operacyjne	6/3
w8	Antywzorce projektowe	2/2
w9	Programowanie ekstremalne	2/2
w10	Praca w zespole programistycznym	2/2
w11	Podsumowanie; kolokwium zaliczeniowe	2/2
**		
Kod	Tematyka zajęć	Liczba godzin S/N
I1	Zapoznanie ze środowiskiem pracy i przedstawienie warunków zaliczenia	2/1
I2	Projektowanie rozproszonego systemu informatycznego; dobór technologii projektowej; wytworzenie specyfikacji	6/3
I3	Implementacja wybranych wzorców projektowych	8/4
I4	Refaktoryzacja pod względem dobrych praktyk programistycznych	4/2
I5	Refaktoryzacja pod względem czystego kodu	4/2
I6	Praktyczne wykorzystanie programowania ekstremalnego	2/1
I7	Testowanie integracyjne rozproszonego systemu	2/1
I8	Prezentacja pracy projektowej	2/1
V. METODY KSZTAŁCENIA, NARZĘDZIA DYDAKTYCZNE		
<p>1. Metody kształcenia:</p> <ul style="list-style-type: none"> • wykład multimedialny • praca projektowa <p>2. Narzędzia (środki) dydaktyczne:</p> <ul style="list-style-type: none"> • prezentacje multimedialne • internet • dyskusja ze studentami na temat wybranych tematów 		
VI. FORMA I KRYTERIA ZALICZENIA MODUŁU		
<p>Forma zaliczenia modułu. Egzamin</p> <p>Kryteria oceny formującej***: 1. Aktywność na wykładach</p> <p>Kryteria oceny podsumowującej*** 1. Egzamin pisemny lub ustny z wykładu 2. Projekt podczas zajęć laboratoryjnych</p> <p>Ocena podsumowująca***: Ocena z modułu: średnia ważona ocen podsumowujących: ocena z projektu o wadze 7 + ocena z egzaminu o wadze 3.</p>		
VII. BILANS PUNKTÓW ECTS - NAKŁAD PRACY STUDENTA		

Kategoria	Obciążenie studenta
Liczba godzin realizowanych przy bezpośrednim udziale nauczyciela (godziny kontaktowe)	
Udział w wykładach	30/24
Udział w innych formach zajęć: zajęcia laboratoryjne	30/16
Inne (jakie?)	
Samodzielna praca studenta (godziny niekontaktowe)	
Przygotowanie do wykładu	8/8
Przygotowanie do innych form zajęć: zajęcia laboratoryjne	18/34
Przygotowanie do egzaminu	2/4
Przygotowanie do zaliczenia innych zajęć: zajęcia laboratoryjne	2/4
Inne (np. gromadzenie materiałów do projektu, kwerenda internetowa, opracowanie prezentacji multimedialnej itp.)	0/0
Łączna liczba godzin	90
Punkty ECTS za moduł	3
VIII. ZALECANA LITERATURA	
Literatura podstawowa:	
1. Robert C. Martin, "Czysty kod"	
2. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku"	
Literatura uzupełniająca:	
1. Refactoring Guru: https://refactoring.guru/design-patterns	

*należy odpowiednio wypełnić

**należy wpisać formę/formy przypisane do modułu określone w programie studiów (wykład, ćwiczenia, seminarium, konwersatorium, lektorat, laboratorium, warsztat, projekt, zajęcia praktyczne, zajęcia terenowe, zajęcia wychowania fizycznego, praktyka zawodowa, inne)

*** proszę wpisać odpowiednie kryteria oceny formującej i podsumowującej