

## KARTA MODUŁU

I. OGÓLNE INFORMACJE O MODULE						
<b>COLLEGIUM WITELONA UCZELNIA PAŃSTWA WYDZIAŁ NAUK TECHNICZNYCH I EKONOMICZNYCH</b>						
<b>Kierunek studiów:</b>	<b>INFORMATYKA</b>					
<b>Poziom studiów:</b>	studia pierwszego stopnia					
<b>Profil studiów:</b>	praktyczny					
<b>Forma studiów:</b>	stacjonarne/niestacjonarne					
<b>Nazwa modułu:</b>	<b>Zaawansowane metody programowania</b>					
<b>Rodzaj modułu:</b>	Obowiązkowy					
<b>Język wykładowy:</b>	Język polski*					
<b>Rok studiów:</b>	3	<b>Formy prowadzenia zajęć wraz z liczbą godzin dydaktycznych:</b>				
<b>Semestr:</b>	6	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
<b>Liczba punktów ECTS ogółem:</b>	3	30/16	-	30/16	-	-
<b>Forma zaliczenia:</b>	Egzamin					
<b>Wymagania wstępne:</b>	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						
<b>Cele kształcenia:</b>						
<b>Cel 1:</b> Przedstawić teorię oraz przykłady wykorzystania dobrych praktyk programistycznych <b>Cel 2:</b> Przedstawić teorię oraz przykłady wykorzystania wzorców projektowych <b>Cel 3:</b> Przedstawić teorię oraz przykłady wykorzystania paradygmatów programowania <b>Cel 4:</b> 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			
<b>wiedzy:</b>						
W1	Rozumie potrzebę wytwarzania oprogramowania zgodnie z zasadami czystego kodu	K1I_W04 K1I_W10	Egzamin			
W2	Potrafi opisać i wskazać zastosowanie wzorców projektowych	K1I_W04 K1I_W10	Egzamin			
<b>umiejętności:</b>						
U1	Potrafi wykorzystać programowanie ekstremalne	K1I_U08	Egzamin + praca projektowa			
U2	Potrafi wskazać odpowiedzialności wskazanych klas i metod oraz stwierdzić czy łamana jest zasada pojedynczej odpowiedzialności	K1I_U08 K1I_U10	Egzamin + praca projektowa			
U3	Potrafi zaproponować elastyczną architekturę kodu opartą o polimorfizm i abstrakcję	K1I_U12	Egzamin + praca projektowa			
U4	Potrafi ocenić kod pod względem wykorzystania dobrych praktyk programistycznych	K1I_U16	Egzamin + praca projektowa			
<b>kompetencji społecznych:</b>						
-	-	-	-			
IV. TREŚCI PROGRAMOWE						
<b>Treści programowe (tematyka zajęć, zaprezentowana z podziałem na poszczególne formy zajęć z określeniem liczby godzin potrzebnych na ich realizację)</b>						
<b>Wykład:</b>						

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
<b>Laboratorium:</b>		
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
<b>V. METODY KSZTAŁCENIA, NARZĘDZIA DYDAKTYCZNE</b>		
<p><b>1. Metody kształcenia:</b></p> <ul style="list-style-type: none"> <li>● wykład multimedialny</li> <li>● praca projektowa</li> </ul> <p><b>2. Narzędzia (środki) dydaktyczne:</b></p> <ul style="list-style-type: none"> <li>● prezentacje multimedialne</li> <li>● internet</li> <li>● dyskusja ze studentami na temat wybranych tematów</li> </ul>		
<b>VI. FORMA I KRYTERIA ZALICZENIA MODUŁU</b>		
<p><b>Sposób zaliczenia:</b> Egzamin</p> <p><b>Formy zaliczenia:</b> Egzamin pisemny: pytania otwarte</p> <p><b>Podstawowe kryteria oceny:</b> Egzamin pisemny z wykładu: 50-59% - ocena dostateczna, 60-69% - ocena dostateczna plus, 70-79% - ocena dobra, 80-89% - ocena dobra plus, powyżej 90% - ocena bardzo dobra.</p>		
<b>VII. BILANS PUNKTÓW ECTS - NAKŁAD PRACY STUDENTA</b>		
<b>Kategoria</b>		<b>Obciążenie studenta</b>
<i>Liczba godzin realizowanych przy bezpośrednim udziale nauczyciela (godziny kontaktowe)</i>		<b>60/32</b>
Udział w wykładach		30/16

Udział w innych formach zajęć: zajęcia laboratoryjne	30/16
<b>Samodzielna praca studenta (godziny niekontaktowe)</b>	<b>30/58</b>
Przygotowanie do wykładu	8/8
Przygotowanie do innych form zajęć: zajęcia laboratoryjne	16/42
Przygotowanie do egzaminu	4/4
Przygotowanie do zaliczenia innych zajęć: zajęcia laboratoryjne	2/4
<b>Łączna liczba godzin</b>	<b>90</b>
<b>Punkty ECTS za moduł</b>	<b>3</b>

#### VIII. ZALECANA LITERATURA

##### Literatura podstawowa:

1. Robert C. Martin, "Czysty kod", Helion, 2015
2. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku", Helion, 2010

##### Literatura uzupełniająca:

1. Refactoring Guru: <https://refactoring.guru/design-patterns>