



## IV. Opis programu studiów

### 3. KARTA PRZEDMIOTU

Kod przedmiotu	Z-ZIPN1-U-621
Nazwa przedmiotu	Programowanie obiektowe w RAD
Nazwa przedmiotu w języku angielskim	RAD Object Oriented Programming
Obowiązuje od roku akademickiego	2019/2020

#### USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	ZARZĄDZANIE I INŻYNIERIA PRODUKCJI
Poziom kształcenia	I stopień
Profil studiów	Ogólnoakademicki
Forma i tryb prowadzenia studiów	Studia niestacjonarne
Zakres	Informatyka w zarządzaniu i modelowaniu
Jednostka prowadząca przedmiot	Katedra Informatyki i Matematyki Stosowanej
Koordinator przedmiotu	dr inż. Sławomir Koczubiej
Zatwierdził	dr hab. inż. Artur Bartosik, prof. PŚk

#### OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	Przedmiot specjalnościowy
Status przedmiotu	Obowiązkowy
Język prowadzenia zajęć	Polski
Usytuowanie modułu w planie studiów - semestr	Semestr VI
Wymagania wstępne	Technologie informacyjne Podstawy informatyki Algorytmy i struktury danych
Egzamin (TAK/NIE)	TAK
Liczba punktów ECTS	3

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

## EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Student ma wiedzę w zakresie syntaktyki i semantyki instrukcji sterujących języka orientowanego obiektowo. Student ma wiedzę na temat typów danych i doboru tych typów do rozwiązania określonego zadania programistycznego. Rozumie pojęcie dziedziczenia, agregacji i polimorfizmu obiektów.	ZIP1_W05
	W02	Student ma poszerzoną wiedzę nt. programowania w zakresie posługiwania się plikami. Zna zasady obsługi plików zgodnej z architekturą systemu operacyjnego.	ZIP1_W04 ZIP1_W05
	W03	Student ma wiedzę nt. zasad wykorzystywania komponentów graficznych do w budowie interfejsu GUI aplikacji. Ma wiedzę nt. zasad posługiwania się narzędziem typu RAD do projektowania aplikacji wielookienkowych.	ZIP1_W05
Umiejętności	U01	Student potrafi zapisać w języku programowania obiektowy model opisujący proste zagadnienie inżynierskie.	ZIP1_U07
	U02	Student potrafi wykorzystać środowisko programistyczne do zaprojektowania i zbudowania programu komputerowego, kompilacji, konsolidacji i testowania programu. Student potrafi zaprojektować i zbudować program komputerowy, korzystając z gotowych obiektów systemu programistycznego z wykorzystaniem plików i ilustracji graficznych.	ZIP1_U07
	U03	Student potrafi dokonać analizy prostego kodu źródłowego, zdiagnozować błędy oraz wprowadzić modyfikacje w treść istniejącego programu.	ZIP1_U07
Kompetencje społeczne	K01	Student rozumie potrzebę stałego uzupełniania wiedzy. Posiada kompetencje w zakresie wykorzystania zasobów sieci Internet dla samokształcenia.	ZIP1_K01
	K02	Student potrafi pracować samodzielnie i w grupie, przyjmując w niej różne role.	ZIP1_K04

## TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	1. Wprowadzenie do języka orientowanego obiektowo. Zmienne i typy. Instrukcje sterujące, tablice i listy. Klasy, obiekty, metody.
	2. Wyjątki i ich obsługa. Obiekty i zarządzanie pamięcią. Tworzenie i niszczenie obiektów.
	3. Operacje na tekstach, znaki specjalne, przetwarzanie łańcuchów znaków. Aplikacje konsolowe z parametrami. Operatory przeciążone. Strumienie i obsługa plików.
	4. Interfejs graficzny, budowa, zadania.
laboratorium	1. Przetwarzanie warunkowe i iteracyjne. Budowa aplikacji orientowanej obiektowo. Definiowane klas. Składowe klasy, obiekty. Dziedziczenie, polimorfizm i tablice wskaźników.
	2. Tworzenie i niszczenie obiektów. Konstruktor, destruktor i zarządzanie pamięcią.
	3. Aplikacje przetwarzające łańcuchy znaków. Budowa aplikacji konsolowej z obsługą wywołania parametrycznego.
	4. Opracowanie aplikacji z graficznym interfejsem użytkownika. Aplikacje sterowane zdarzeniami. Współpraca z plikami. Aplikacje wielookienkowe.

\*) zostawić tylko realizowane formy zajęć

**METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ**

Symbol efektu	Metody sprawdzania efektów kształcenia <i>(zaznaczyć X)</i>					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01		X	X			
W02		X	X			
U01		X	X		X	
U02		X	X		X	
K01		X	X		X	
K02					X	

## FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład	egzamin	Uzyskanie co najmniej 50% punktów z egzaminu, komentarze na wykładach.
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% z kolokwium, aktywność na zajęciach laboratoryjnych.

\*) zostawić tylko realizowane formy zajęć

## NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
L p.	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)	4		2			h
3.	<b>Razem przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>33</b>					h
4.	<b>Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>1,3</b>					ECTS
5.	<b>Liczba godzin samodzielnej pracy studenta</b>	<b>42</b>					h
6.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy</b>	<b>1,7</b>					ECTS
7.	<b>Nakład pracy związany z zajęciami o charakterze praktycznym</b>	<b>50</b>					h
8.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym</b>	<b>2</b>					ECTS
9.	<b>Sumaryczne obciążenie pracą studenta</b>	<b>75</b>					h
10.	<b>Punkty ECTS za moduł</b> <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	<b>3</b>					ECTS

## LITERATURA

1. Prata S. (2006), *Szkoła programowania. Język C*, Wydawnictwo Helion, Gliwice.
2. Prata S. (2006), *Szkoła programowania. Język C++*, Wydawnictwo Helion, Gliwice.
3. Grębosz J. (2009), *Symfonia C++ Standard*, Wydawnictwo Edition 2000, Kraków.
4. Chappell D. (2007), *Zrozumieć platformę .NET*, Wydawnictwo Helion, Gliwice.
5. Michelsen K. (2007), *Język C#. Szkoła programowania*, Wydawnictwo Helion, Gliwice.
6. Matulewski J. (2008), *C# 3.0 i .NET 3.5. Technologia LINQ*, Wydawnictwo Helion, Gliwice.
7. Lee W.M. (2010), *C# 2008. Warsztat programisty*, Wydawnictwo Helion, Gliwice.