



KARTA PRZEDMIOTU

Kod przedmiotu	studia stacjonarne:	Z-ZIP1-U-621
	studia niestacjonarne:	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 PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	ZARZĄDZANIE I INŻYNIERIA PRODUKCJI
Poziom kształcenia	I stopień
Profil studiów	Ogólnoakademicki
Forma i tryb prowadzenia studiów	Studia stacjonarne i niestacjonarne
Zakres	Informatyka w zarządzaniu i modelowaniu
Jednostka prowadząca przedmiot	Katedra Technologii Informatycznych
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 w planie studiów - semestr	studia stacjonarne	Semestr VI
	studia niestacjonarne	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	studia stacjonarne:	15		30		
	studia niestacjonarne:	9		18		

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty uczenia się	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	<ol style="list-style-type: none">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	<ol style="list-style-type: none">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.

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów kształcenia (zaznaczyć X)					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01		X	X			
W02		X	X			
W03		X	X			
U01		X	X		X	
U02		X	X		X	
U03		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 kolokwiów, aktywność na zajęciach laboratoryjnych.

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS												
Lp.	Rodzaj aktywności	Obciążenie studenta										Jednostka
		studia stacjonarne					studia niestacjonarne					
		W	C	L	P	S	W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	15		30			9		18			h
2.	Inne (konsultacje, egzamin)	4		2			4		2			h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	51					33					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	2,0					1,3					ECTS
5.	Liczba godzin samodzielnej pracy studenta	24					42					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	1,0					1,7					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	50					50					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	2,0					2,0					ECTS
9.	Sumaryczne obciążenie pracą studenta	75					75					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	3										ECTS

LITERATURA

1. Prata S., Szkoła programowania. Język C, Helion, Gliwice 2016.
2. Prata S., Szkoła programowania. Język C++, Helion, Gliwice 2013.
3. Sokół R., Microsoft Visual Studio 2012. Programowanie w C i C++, Helion, Gliwice 2013.
4. Griffiths J., C# 10. Programowanie. Tworzenie aplikacji Windows, internetowych i biurowych, Helion, Gliwice 2023.
5. Summerfield M., Zaawansowane programowanie przy użyciu C++, Helion, Gliwice 2014.