

KARTA PRZEDMIOTU**I. Dane podstawowe**

Nazwa przedmiotu	Laboratorium programowania – aplikacje w różnych środowiskach programistycznych
Nazwa przedmiotu w języku angielskim	Application development with different programming languages
Kierunek studiów	Informatyka
Poziom studiów (I, II, jednolite magisterskie)	I stopnia
Forma studiów (stacjonarne, niestacjonarne)	Stacjonarne
Dyscyplina	Informatyka, informatyka techniczna i telekomunikacja
Język wykładowy	Polski

Koordinator przedmiotu/osoba odpowiedzialna	mgr inż. Kamil Zieliński
---	--------------------------

Forma zajęć (<i>katalog zamknięty ze słownika</i>)	Liczba godzin	Semestr	Punkty ECTS
wykład			3+3
konwersatorium			
ćwiczenia			
laboratorium	30+30	V+VI	
warsztaty			
seminarium			
proseminarium			
lektorat			
praktyki			
zajęcia terenowe			
pracownia dyplomowa			
translatorium			
wizyta studyjna			

Wymagania wstępne	Wstęp do programowania Podstawy informatyki i programowania Programowanie obiektowe
-------------------	---

II. Cele kształcenia dla przedmiotu

Zaprezentowanie różnych środowisk programistycznych
Omówienie możliwości wykorzystania języków programowania do rozwiązywania problemów różnego typu
Przedstawienie zaawansowanych środowisk programistycznych oraz możliwości ich zastosowania
Omówienie dobrych praktyk programistycznych

III. Efekty uczenia się dla przedmiotu wraz z odniesieniem do efektów kierunkowych

Symbol	Opis efektu przedmiotowego	Odniesienie do efektu kierunkowego
WIEDZA		
W_01	Student zna składnię i specyfikę różnych języków programowania	K_W01, K_W03, K_W04, K_W06
W_02	Student posiada wiedzę na temat możliwości zastosowania Zintegrowanych Środowisk Programistycznych	K_W01
W_03	Student zna zasady pisania czytelnego kodu źródłowego	K_W01, K_W03, K_W04, K_W06
UMIEJĘTNOŚCI		
U_01	Student potrafi zaimplementować rozwiązania programistyczne z wykorzystaniem różnych języków programowania	K_U08, K_U12, K_U27
U_02	Student potrafi wykorzystywać zaawansowane funkcjonalności Zintegrowanych Środowisk Programistycznych	K_U02
U_03	Student potrafi przygotować czytelny kod źródłowy	K_U04, K_U12
U_04	Student potrafi wykorzystywać dokumentację techniczną podczas pracy z projektem	K_U02
KOMPETENCJE SPOŁECZNE		
K_01	Student potrafi pracować indywidualnie i zespołowo podczas realizacji zadań programistycznych	K_K02, K_K04

IV. Opis przedmiotu/ treści programowe

<p>Programowanie obiektowe w wybranym języku programowania Wykorzystanie Zintegrowanych Środowisk Programistycznych Rozwiązywanie problemów algorytmicznych z wykorzystaniem języków programowania Przygotowanie czytelnego kodu źródłowego</p>
--

V. Metody realizacji i weryfikacji efektów uczenia się

Symbol efektu	Metody dydaktyczne <i>(lista wyboru)</i>	Metody weryfikacji <i>(lista wyboru)</i>	Sposoby dokumentacji <i>(lista wyboru)</i>
WIEDZA			
W_01	Praca pod kierunkiem	Projekt	Karta oceny projektu
W_02	Praca pod kierunkiem	Projekt	Karta oceny projektu
W_03	Praca pod kierunkiem	Projekt	Karta oceny projektu
UMIEJĘTNOŚCI			
U_01	Ćwiczenia praktyczne	Projekt	Karta oceny projektu
U_02	Ćwiczenia praktyczne	Projekt	Karta oceny projektu
U_03	Ćwiczenia praktyczne	Projekt	Karta oceny projektu
U_04	Ćwiczenia praktyczne	Projekt	Karta oceny projektu
KOMPETENCJE SPOŁECZNE			
K_01	Dyskusja, Metoda problemowa PBL	Projekt	Karta oceny projektu

VI. Kryteria oceny, wagi...

Warunkiem zaliczenia przedmiotu jest obecność na zajęciach oraz przygotowanie projektu zaliczeniowego w wybranej technologii.

Obciążenie pracą studenta

Forma aktywności studenta	Liczba godzin
Liczba godzin kontaktowych z nauczycielem	50+50
Liczba godzin indywidualnej pracy studenta	30+30

VII. Literatura

Literatura podstawowa
Hunt A., Thomas D., Pragmatyczny programista. Od czeladnika do mistrza, wyd. Helion, Gliwice 2011
Martin R.C., Czysty kod. Podręcznik dobrego programisty, wyd. Helion, Gliwice 2010
Literatura uzupełniająca
Bloch J., Java. Efektywne programowanie, wyd. Helion, Gliwice 2018
Fowler M., Beck K., Brant J., Opdyke W., Roberts D., Gamma E., Refaktoryzacja. Ulepszanie struktury istniejącego kodu, wyd. Helion, Gliwice 2011
Martin R.C., Mistrz czystego kodu. Kodeks postępowania profesjonalnych programistów, wyd. Helion, Gliwice 2013