POLITECHNIKA GDAŃSKA

Karta przedmiotu

Nazwa i kod przedmiotu	Programowanie obiektowe, PG_00058915		
Kierunek studiów	Informatyka		
Data rozpoczęcia studiów	październik 2024 r.	Rok akademicki realizacji przedmiotu	2025/2026
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć	Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki
Forma studiów	niestacjonarne	Sposób realizacji	na uczelni
Rok studiów	2	Język wykładowy	polski
Semestr studiów	3	Liczba punktów ECTS	6.0
Profil kształcenia	ogólnoakademicki	Forma zaliczenia	zaliczenie
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Systemów Geoinformatycznych		
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr hab. inż. Marek Moszyński	
	Prowadzący zajęcia z przedmiotu	dr hab. inż. Marek Moszyński dr inż. Andrzej Chybicki	

Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	0.0	15.0	0.0	45
	W tym liczba godzin zajęć na odległość: 0.0						
	Adresy kursu na platformie eNauczanie: Moodle ID: 20994 Programowanie obiektowe (ns) 2025 https://enauczanie.pg.edu.pl/moodle/course/view.php?id=20994						

Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach	Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	45	15.0	90.0	150

Cel przedmiotu	Nabycie wiedzy z zakresu teorii programowania obiektowego oraz umiejętności programowania z wykorzystaniem tego typu podejścia.

Efekty uczenia się przedmiotu

Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
[K6_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów	Student zdobywa umiejętności praktyczne wykonując zadania laboratoryjne w określonych językach programowania obiektowego	[SU1] Ocena realizacji zadania
[K6_U41] potrafi wytwarzać, testować lub oceniać oprogramowanie, wykorzystując nowoczesne platformy, narzędzia, języki i paradygmaty programowania różnych poziomów, a także posługiwać się pakietami oprogramowania wspierającymi naukowo-badawcze i biznesowe procesy decyzyjne oraz pracę zespołową	Student zdobywa umiejętności praktyczne wykonując przykładowe zadania w kilku językach programowania obiektowego	[SU1] Ocena realizacji zadania [SU4] Ocena umiejętności korzystania z metod i narzędzi
[K6_W04] zna i rozumie w zaawansowanym stopniu zasady, metody i techniki programowania oraz zasady tworzenia oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, specyficznych dla kierunku studiów, a także organizację pracy systemów wykorzystujących komputery lub te urządzenia	Student zapoznaje się z podstawami programowania obiektowego na przykładzie czterech języków programowania obiektowego	[SW1] Ocena wiedzy faktograficznej

Treści przedmiotu

1. Paradygmaty programowania ze szczególnym uwzględnieniem paradygmatu obiektowego

2. Implementacja enkapsulacji, dziedziczenia, abstrakcji i polimorfizmu w języku C++

3. Specyfika implementacji obiektowości w języku C++

4. Język Java i jego porównanie z językiem C++

5. Język C# i jako następca języka C i porównanie z językiem Java

6. Język Python jako przedstawiciel skryptowych języków programowania obiektowego

Wymagania wstępne i dodatkowe

Nie ma wymagań

Sposoby i kryteria oceniania osiąganych efektów uczenia się

Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
Projekt	55.0%	60.0%
Kolokwia w czasie semestru	55.0%	40.0%

Zalecana lista lektur

Podstawowa lista lektur	1. Bjarne Stroustrup, The C++ Programming Language 2. Bruce Eckel, Thinking in Java 3. Mark Lutz, Programming Python
Uzupełniająca lista lektur	1. John Hunt, Smalltalk and Object Orientation 2. Bruce Eckel, Thinking in C++
Adresy eZasobów	

Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania

Przykładowe pytanie: w jakim kierunku rozwija się język C++?

Przykładowe zadanie: Implementacja prostego programu obiektowego z zastosowaniem paradygmatów programowania obiektowego w różnych językach programowania.

Praktyki zawodowe w ramach przedmiotu

Nie dotyczy

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.