

Karta przedmiotu

Nazwa i kod przedmiotu	Metody programowania, PG_00191646						
Kierunek studiów	Informatyka (O)						
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu			2026/2027		
Poziom kształcenia	I stopnia - licencjackie	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			polski		
Semestr studiów	2	Liczba punktów ECTS			3.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca							
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr Tomasz Borzyszkowski					
	Prowadzący zajęcia z przedmiotu						
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	30.0	0.0	0.0	45
	W tym liczba godzin zajęć na odległość: 0.0						
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	0.0		30.0		75
Cel przedmiotu	Celem przedmiotu jest zapoznanie słuchaczy z zasadami programowania strukturalnego i obiektowego w połączeniu z dobrymi praktykami programistycznymi na bazie wybranego języka programowania.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu		Sposób weryfikacji i oceny efektu		
	[INFOL3_U06] potrafi dobrać i stosować właściwe metody i narzędzia informatyczne do rozwiązywania złożonych problemów, potrafi wykorzystać posiadaną wiedzę poprzez właściwy dobór źródeł i informacji z nich pochodzących, dokonywanie oceny, krytycznej analizy i syntezy tych informacji						
	[INFOL3_W05] zna i rozumie zaawansowane zagadnienia w zakresie różnych paradygmatów programowania i języków programowania; a także metody i wzorce projektowania i programowania obiektowego		student wyjaśnia działanie podstawowych konstrukcji programistycznych takich jak instrukcje wyboru, pętle, bloki, struktury itp.		[SW4] test/egzamin - ustny lub pisemny [SW2] prezentacja/projekt/referat/raport		

Treści przedmiotu	<ol style="list-style-type: none"> <li>1. <b>Podstawowe koncepcje:</b> <ul style="list-style-type: none"> <li>• kompilatory i interpretery</li> <li>• proste i strukturalne typy danych oraz operacje na danych.</li> </ul> </li> <li>2. <b>Funkcje:</b> <ul style="list-style-type: none"> <li>• definicja funkcji i jej znaczenie w konstrukcji programu komputerowego</li> <li>• sposoby przekazywania parametrów do funkcji oraz zwracania wyniku</li> <li>• lambda-funkcje, ich podstawy matematyczne oraz wykorzystanie w programie komputerowym</li> <li>• dokumentowanie funkcji.</li> </ul> </li> <li>3. <b>Moduły i pakiety:</b> <ul style="list-style-type: none"> <li>• koncepcja przestrzeni nazw</li> <li>• definicja modułu oraz sposoby jego importowania</li> <li>• tworzenie pakietów/bibliotek rozwiązań.</li> </ul> </li> <li>4. <b>Klasy i obiekty:</b> <ul style="list-style-type: none"> <li>• definicja klasy i tworzenie obiektów</li> <li>• dziedziczenie i atrybuty</li> <li>• wykorzystanie funkcji polimorficznych w programowaniu obiektowym.</li> </ul> </li> <li>5. <b>Testy jednostkowe:</b> <ul style="list-style-type: none"> <li>• koncepcje programowania sterowanego testami</li> <li>• tworzenie testów jednostkowych</li> <li>• automatyzacja testów</li> <li>• tworzenie testów behawioralnych</li> </ul> </li> <li>6. <b>Pliki i wyjątki:</b> <ul style="list-style-type: none"> <li>• podstawowe operacje na plikach</li> <li>• serializacja i deserializacja danych</li> <li>• tworzenie, podnoszenie i obsługa wyjątków.</li> </ul> </li> <li>7. <b>Wyrażenia regularne:</b> <ul style="list-style-type: none"> <li>• przegląd podstawowych konstrukcji</li> <li>• grupy i podgrupy wyrażeń regularnych</li> <li>• wykorzystanie wyrażeń regularnych do wybranych zadań programistycznych.</li> </ul> </li> <li>8. <b>Czysty kod:</b> <ul style="list-style-type: none"> <li>• przegląd podstawowych paradygmatów programistycznych oraz ich praktyczne zastosowanie na wybranych przykładach</li> <li>• wykorzystanie wybranych wzorców projektowych</li> <li>• pojęcie długu technologicznego.</li> </ul> </li> </ol>														
Wymagania wstępne i dodatkowe	Ukończony przedmiot: Wstęp do programowania.														
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Sposób oceniania (składowe)</th> <th style="width: 33%;">Próg zaliczeniowy</th> <th style="width: 33%;">Składowa oceny końcowej</th> </tr> </thead> <tbody> <tr> <td>prezentacja projektu, obserwacja pracy studenta</td> <td>51.0%</td> <td>50.0%</td> </tr> <tr> <td>aktywność na zajęciach</td> <td>0.0%</td> <td>10.0%</td> </tr> <tr> <td>egzamin</td> <td>51.0%</td> <td>40.0%</td> </tr> </tbody> </table>			Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej	prezentacja projektu, obserwacja pracy studenta	51.0%	50.0%	aktywność na zajęciach	0.0%	10.0%	egzamin	51.0%	40.0%
Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej													
prezentacja projektu, obserwacja pracy studenta	51.0%	50.0%													
aktywność na zajęciach	0.0%	10.0%													
egzamin	51.0%	40.0%													
Zalecana lista lektur	<p>Podstawowa lista lektur</p> <p>Uzupełniająca lista lektur</p> <p>Adresy eZasobów</p>	<ol style="list-style-type: none"> <li>1. Metodologia programowania: <ul style="list-style-type: none"> <li>• Robert C. Martin, Czysty kod. Podręcznik dobrego programisty. Helion 2023.</li> <li>• Beck Kent, TDD. Sztuka tworzenia dobrego kodu Wydawnictwo Helion, 2020.</li> <li>• Harry J.W. Percival. TDD w praktyce. Niezawodny kod w języku Python. Wydawnictwo Helion, 2020.</li> <li>• Eric Freeman, Elisabeth Freeman, Kathy Sierra, Bert Bates, Head First Design Patterns. Edycja polska (Rusz głową!). Helion 2020.</li> </ul> </li> <li>2. Wybrany język programowania: <ul style="list-style-type: none"> <li>• Guido van Rossum, Python Tutorial, <a href="http://docs.python.org/tut/">http://docs.python.org/tut/</a>.</li> <li>• Mark Pilgrim, Dive into Python. <a href="http://diveintopython.org/">http://diveintopython.org/</a>.</li> <li>• Bruce Eckel, Thinking in Python, <a href="http://www.mindview.net/Books/TIPython">http://www.mindview.net/Books/TIPython</a>.</li> <li>• Python's official documentation, <a href="http://docs.python.org/">http://docs.python.org/</a>.</li> </ul> </li> </ol>													
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania															
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy														

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.