

Karta przedmiotu

Nazwa i kod przedmiotu	Języki programowania (OA), PG_00143564						
Kierunek studiów	Informatyka (O)						
Data rozpoczęcia studiów	październik 2024 r.	Rok akademicki realizacji przedmiotu			2024/2025		
Poziom kształcenia	I stopnia - licencjackie	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			polski		
Semestr studiów	2	Liczba punktów ECTS			7.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			egzamin		
Jednostka prowadząca							
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr Tomasz Borzyszkowski				
	Prowadzący zajęcia z przedmiotu		dr Tomasz Borzyszkowski mgr Laura Grzonka				
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	30.0	0.0	0.0	60
	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	60		0.0		115.0	175
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		
	[INFL3_W05] ma ogólną wiedzę na temat różnych paradygmatów programowania i języków programowania; szczególnie zna 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		
	[INFL3_U06] potrafi projektować, tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz adekwatnych wzorców		student potrafi zastosować poznane konstrukcje programistyczne do pisania działających i poprawnych programów		[SU2] prezentacja/projekt/referat/raport [SU8] obserwacja samodzielnej lub zespołowej pracy studenta		
	[INFL3_U09] potrafi oceniać przydatność paradygmatów i narzędzi programistycznych do rozwiązywania problemów różnego typu		student potrafi zastosować poznane konstrukcje programistyczne do pisania działających i poprawnych programów		[SU2] prezentacja/projekt/referat/raport [SU8] obserwacja samodzielnej lub zespołowej pracy studenta		

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

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.