Uniwersytet Gdański

Karta przedmiotu

Nazwa i kod przedmiotu	Języki programowania (OA), PG_00143564		
Kierunek studiów	Informatyka (O)		
Data rozpoczęcia studiów	październik 2025 r.	Rok akademicki realizacji przedmiotu	2025/2026
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	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		

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; szczegółowo 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_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
[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

<table>
<tr><td>Treści przedmiotu</td><td>1. Podstawowe koncepcje:
• kompilatory i interpretery
• proste i strukturalne typy danych oraz operacje na danych.
2. Funkcje:
• 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:
• koncepcja przestrzeni nazw
• definicja modułu oraz sposoby jego importowania
• tworzenie pakietów/bibliotek rozwiązań.
4. Klasy i obiekty:
• definicja klasy i tworzenie obiektów
• dziedziczenie i atrybuty
• wykorzystanie funkcji polimorficznych w programowaniu obiektowym.
5. Testy jednostkowe:
• koncepje programowania sterowanego testami
• tworzenie testów jednostkowych
• automatyzacja testów
• tworzenie testów behawioralnych
6. Pliki i wyjątki:
• podstawowe operacje na plikach
• serializacja i deserializacja danych
• tworzenie, podnoszenie i obsługa wyjątków.
7. Wyrażenia regularne:
• przegląd podstawowych konstrukcji
• grupy i podgrupy wyrażeń regularnych
• wykorzystanie wyrażeń regularnych do wybranych zadań programistycznych.
8. Czysty kod:
• przegląd podstawowych paradygmatów programistycznych oraz ich praktyczne zastosowanie na wybranych przykładach
• wykorzystanie wybranych wzorców projektowych
• pojęcie długu technologicznego.</td></tr>
<tr><td>Wymagania wstępne i dodatkowe</td><td>Ukończony przedmiot: Wstęp do programowania.</td></tr>
<tr><td>Sposoby i kryteria oceniania osiąganych efektów uczenia się</td><td>
<table>
<tr><th>Sposób oceniania (składowe)</th><th>Próg zaliczeniowy</th><th>Składowa oceny końcowej</th></tr>
<tr><td>egzamin</td><td>51.0%</td><td>40.0%</td></tr>
<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>
</table>
</td></tr>
<tr><td>Zalecana lista lektur</td><td>
<table>
<tr><td>Podstawowa lista lektur</td><td>1. Metodologia programowania:
• 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:
• 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/.</td></tr>
<tr><td>Uzupełniająca lista lektur</td><td>Brak</td></tr>
<tr><td>Adresy eZasobów</td><td></td></tr>
</table>
</td></tr>
<tr><td>Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania</td><td></td></tr>
<tr><td>Praktyki zawodowe w ramach przedmiotu</td><td>Nie dotyczy</td></tr>
</table>

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.