

Karta przedmiotu

Nazwa i kod przedmiotu	Programowanie obiektowe, PG_00191644						
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		
Forma studiów	stacjonarne		Sposób realizacji		na uczelni		
Rok studiów	1		Język wykładowy		polski		
Semestr studiów	2		Liczba punktów ECTS		5.0		
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		65.0	125
Cel przedmiotu	Celem przedmiotu jest zapoznanie studentów z zasadami programowania obiektowego i zdarzeniowego na przykładzie języka Java.						
	Nauczenie studentów podstawowych koncepcji związanych z programowaniem obiektowym oraz umiejętności konstruowania programów obiektowych w języku Java.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[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 definiuje i objaśnia zaawansowane mechanizmy programowania obiektowego (w tym hermetyzację, interfejsy, dziedziczenie oraz polimorfizm metod), a także charakteryzuje strukturę i przeznaczenie wybranych wzorców projektowych.	[SW2] prezentacja/projekt/referat/ raport
	[INFOL3_U02] potrafi wykorzystywać posiadaną wiedzę z zakresu matematyki wyższej w modelowaniu i rozwiązywaniu złożonych problemów	Student potrafi zamodelować złożone problemy algorytmiczne za pomocą paradygmatu obiektowego, wykorzystując modele logiczne do rozwiązywania klasycznych problemów współbieżności (np. problem uczciwości dostępu do zasobów) oraz dobierając odpowiednie operacje na zbiorach i grafach przy użyciu struktur danych (np. interfejsów Set, Map).	[SU2] prezentacja/projekt/referat/ raport
	[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	Student potrafi samodzielnie dobrać i wdrożyć optymalne mechanizmy języka Java - w tym dynamiczne struktury danych (kolekcje), wielowątkowość oraz mechanizmy zgłaszania i wyłapywania wyjątków - w celu zaimplementowania bezpiecznego i stabilnego rozwiązania złożonego problemu programistycznego.	[SU2] prezentacja/projekt/referat/ raport
Treści przedmiotu	Wprowadzenie pojęć podstawowych: klasa, obiekt, pole, metoda; pola i metody: statyczne, publiczne oraz prywatne w języku Java; inicjalizacja i usuwanie obiektu oraz mechanizm tzw. zbieracza śmieci; przegląd instrukcji sterujących języka Java i ich porównanie z instrukcjami sterującymi języka C. Pakiety klas języka Java ukrywanie implementacji: struktura pakietu; importowanie pakietów; ustalanie praw dostępu do składowych pakietu; budowa interfejsów i ich implementacja. Dziedziczenie i polimorfizm: dziedziczenie składnia i zachowanie praw dostępu do dziedziczonych pól i metod; od abstrakcji do konkretności: klasy abstrakcyjne i finalne; porównanie własności pól finalnych i statycznych; porównanie własności metod przeciążonych i polimorficznych; przykłady wywołań funkcji polimorficznych. Przegląd klas implementujących typowe struktury danych: pojęcie statycznych i dynamicznych struktur danych; przegląd własności i udostępnianych operacji na typie Collection języka Java i typach pochodnych (List, Set, BitSet, Maps, Vector, Stack i Hashtable); klasy implementujące typy wyliczeniowe i iteratory; polimorficzne metody umożliwiające sortowanie elementów przechowywanych w kolekcjach. Programowanie z wykorzystaniem wyjątków: przegląd predefiniowanych wyjątków w języku Java; zasady tworzenia nowych wyjątków; zgłaszanie i wyłapywanie sytuacji wyjątkowych. Programowanie z wykorzystaniem wątków: pojęcie wątku, zasobów współdzielonych i sekcji krytycznej; przegląd metod współdzielenia zasobów w języku Java: blokowanie zasobów i problem zakleszczenia wątków oraz kolejki priorytetowe i problem uczciwości w dostępie do zasobów; przykłady wykorzystania wątków języka Java do implementacji klasycznych problemów dostępu do zasobów krytycznych. Przegląd wybranych wzorców projektowych.		
Wymagania wstępne i dodatkowe			
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	projekty	51.0%	50.0%
	egzamin	51.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	- Eckel B., Thinking in Java edycja polska. Wydawnictwo HELION, Warszawa, 2010. - Horstmann C. S., Cornell G., Core Java 2 - Podstawy. Helion, 2010. - Horstmann C. S., Cornell G., Core Java 2 - Techniki zaawansowane. Helion, 2010. - Campione M., Walrath K., Java Tutorial. Addison-Wesley, 2000.	
	Uzupełniająca lista lektur	Brak	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		