Uniwersytet Gdański

Karta przedmiotu

Nazwa i kod przedmiotu	Programowanie obiektowo-funkcyjne, PG_00204174		
Kierunek studiów	Informatyka (P)		
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu	2027/2028
Poziom kształcenia	I stopnia - licencjackie	Grupa zajęć	Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z praktycznym przygotowaniem zawodowym - profil praktyczny
Forma studiów	stacjonarne	Sposób realizacji	na uczelni
Rok studiów	2	Język wykładowy	polski
Semestr studiów	4	Liczba punktów ECTS	3.0
Profil kształcenia	praktyczny	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

Podstawowym celem jest zapoznanie studentów z najważniejszymi koncepcjami podejścia obiektowego i funkcyjnego do programowania. Pozwala to na poszerzerzenie i pogłębienie wiedzy już posiadanej przez studentów oraz ułatwia im dokonywanie doboru odpowiednich technik programowania w zależności od charakteru zadania. Projekty realizowane podczas zajęć laboratoryjnych służą praktycznemu ugruntowaniu koncepcji poznanych podczas wykładu.

Efekty uczenia się przedmiotu

Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
[INFPL3_U04] potrafi wykorzystywać posiadaną wiedzę tworząc, uruchamiając i testując programy przy wykorzystaniu dedykowanych narzędzi oraz wzorców projektowych	potrafi tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz wzorców projektowych	[SU2] prezentacja/projekt/referat/ raport [SU8] obserwacja samodzielnej lub zespołowej pracy studenta
[INFPL3_K02] jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych oraz zasięgania opinii ekspertów w przypadku trudności z samodzielnym rozwiązywaniem problemu	potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania	[SK1] wypowiedź ustna/rozmowa/ dyskusja [SK8] obserwacja samodzielnej lub zespołowej pracy studenta
[INFPL3_W04] zna i rozumie w zaawansowanym stopniu zagadnienia w zakresie programowania, algorytmów i złożoności, języków i paradygmatów programowania oraz złożone zależności między nimi	ma uporządkowaną, podbudowaną teoretycznie wiedzę w zakresie programowania, algorytmów i złożoności, języków i paradygmatów programowania	[SW1] wypowiedź ustna/rozmowa/ dyskusja [SW2] prezentacja/projekt/referat/ raport
[INFPL3_U09] potrafi zgodnie z zadaną specyfikacją zaprojektować oraz zrealizować system informatyczny	potrafi zgodnie z zadaną specyfikacją zaprojektować oraz zrealizować system informatyczny	[SU2] prezentacja/projekt/referat/ raport [SU8] obserwacja samodzielnej lub zespołowej pracy studenta
[INFPL3_U08] potrafi ocenić przydatność różnych paradygmatów i narzędzi programistycznych do rozwiązywania problemów różnego typu	ocenia przydatność różnych paradygmatów i narzędzi programistycznych do rozwiązywania problemów różnego typu	[SU2] prezentacja/projekt/referat/ raport [SU8] obserwacja samodzielnej lub zespołowej pracy studenta

Treści przedmiotu

1. 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.

2. Pakiety klas: ukrywanie implementacji: struktura pakietu; importowanie pakietów; ustalanie praw dostępu do składowych pakietu; budowa interfejsów i ich implementacja.

3. Dziedziczenie i polimorfizm: dziedziczenie: składnia i zachowanie praw dostępu do dziedziczonych pól i metod; od abstrakcji do konkretu: klasy abstrakcyjne i finalne; porównanie własności pól finalnych i statycznych; porównanie własności metod przeciążonych i polimorficznych; przykłady wywołań funkcji polimorficznych.

4. Przegląd klas implementujących typowe struktury danych: pojęcie statycznych i dynamicznych struktur danych; przegląd własności i udostępnianych operacji; klasy implementujące typy wyliczeniowe i iteratory; polimorficzne metody umożliwiające sortowanie elementów przechowywanych w kolekcjach.

5. Pojęcie lambda-funkcji i ich wykorzystanie do przetwarzania strumieni.

6. Programowanie z wykorzystaniem wyjątków: przegląd predefiniowanych wyjątków; zasady tworzenia nowych wyjątków; zgłaszanie i wyłapywanie sytuacji wyjątkowych.

7. Programowanie z wykorzystaniem wątków: pojęcie wątku, zasobów współdzielonych i sekcji krytycznej; przegląd metod współdzielenia zasobów: blokowanie zasobów i problem zakleszczenia wątków oraz kolejki priorytetowe i problem uczciwości w dostępie do zasobów; przykłady wykorzystania wątków do implementacji klasycznych problemów dostępu do zasobów krytycznych.

8. Przegląd wybranych wzorców projektowych.

Wymagania wstępne i dodatkowe

Brak wymagań wstępnych

Sposoby i kryteria oceniania osiąganych efektów uczenia się

Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
prezentacja projetu, obserwacja pracy studenta	51.0%	90.0%
aktywność na zajeciach	51.0%	10.0%

Zalecana lista lektur

Podstawowa lista lektur	1. Eckel B., Thinking in Java edycja polska. Wydawnictwo HELION, Warszawa, 2010. 2. Horstmann C. S., Cornell G., Core Java 2 - Podstawy. Helion, 2010. 3. Horstmann C. S., Cornell G., Core Java 2 - Techniki zaawansowane. Helion, 2010.
Uzupełniająca lista lektur	Brak
Adresy eZasobów	

Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	1. Zaimplementuj klasę wyliczającą wyrażenia arytmetyczne zapisane w Odwrotnej Notacji Polskiej. 2. Zaimplementuj klasę Stack implementującą ideę stosów napisów z następującymi metodami publicznymi: push, pop oraz peek.
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.