

JĘZYKI OBIEKTOWE

Kod przedmiotu: JPO

Rodzaj przedmiotu: kierunkowy; obowiązkowy

Wydział: Informatyki

Kierunek: Informatyka

Poziom studiów: pierwszego stopnia – VI poziom PRK

Profil studiów: praktyczny

Forma studiów: stacjonarna/niestacjonarna

Rok: 2

Semestr: 4

Formy zajęć i liczba godzin:

Forma stacjonarna

wyklady – 30

laboratorium – 25

Forma niestacjonarna

wyklady – 15

laboratorium – 20

Zajęcia prowadzone są w języku polskim.

Liczba punktów ECTS: 4

Osoby prowadzące:

wykład:

laboratorium:

1. Założenia i cele przedmiotu:

Celem zajęć jest uzupełnienie wiedzy studentów dotyczącej zasad projektowania i implementowania programów komputerowych oraz nauczanie pisania czytelnych i sprawnych programów w języku Java. Założeniem jest pokazanie studentom zasad programowania obiektowego w tym języku, uwzględniając też nieobiektywne jego elementy.

2. Określenie przedmiotów wprowadzających wraz z wymaganiami wstępnymi:

Przedmioty wprowadzające to: Podstawy programowania, Języki programowania, Podstawy programowania obiektowego.

Student powinien znać zarówno zasady programowania proceduralnego, jak i podstawy programowania obiektowego.

3. Opis form zajęć

a) Wykłady

- **Treści programowe:**

- Historia i charakterystyka języka, przykład aplikacji
- Typy pierwotne.
- Instrukcje.
- Podstawy programowania obiektowego w Javie.
- Klasy opakowujące.
- Tablice, łańcuchy.
- Klasy – rozwinięcie tematyki: hermetyzacja, dziedziczenie, polimorfizm, interfejsy.
- Obsługa wyjątków.
- Pliki (strumienie).
- Wątki.
- Elementy programowania funkcyjnego.
- Wsparcie zastosowań kryptografii w Javie – JCA i JCE
- Uzupełnienia.

- **Metody dydaktyczne:**

- Wykład prowadzony metodą tradycyjną z wykorzystaniem rzutnika multimedialnego i prezentacją algorytmów oraz ich implementacji w języku Java

- **Forma i warunki zaliczenia:**

Warunkiem zaliczenia całości przedmiotu jest zdanie egzaminu w formie zadaniowej i (ewentualnie) testowej. Część testowa powinna uwzględniać przede wszystkim stronę teoretyczną, zadania są implementacją podanych zagadnień w języku Java.

- **Wykaz literatury podstawowej:**

1. C.S. Horstmann, G. Cornell: Java 2. Techniki zaawansowane, Helion
2. J. Bielecki: Java 3. Programowanie współbieżne, obiektowe i zdarzeniowe, Helion
3. P.-Y. Saumont: Java. Programowanie funkcyjne, Helion
4. D. Hook: Kryptografia w Javie. Od podstaw, Helion

- **Wykaz literatury uzupełniającej:**

1. M. Lis: Java. Ćwiczenia praktyczne, Helion
2. M. Lis: Java. Ćwiczenia zaawansowane, Helion
3. B. Eckel: Thinking in Java. Wydanie 3. Edycja polska, Helion
4. E.Koffman, P.Wolfgang: Struktury danych i techniki obiektowe na przykładzie Javy 5.0. Helion

b) Ćwiczenia

- **Treści programowe:**

- jak dla wykładu

- **Metody dydaktyczne:**

- Prezentacje przypadków,
- Dyskusja,
- Zespołowe rozwiązywanie problemów, realizacja projektów,
- Indywidualne rozwiązywanie zadań.

- **Forma i warunki zaliczenia:**

- Test zaliczeniowy, implementacja podanych zagadnień w języku Java
- Ocena aktywności studentów podczas zajęć

- **Wykaz literatury podstawowej:**

- Jak dla wykładu
 - **Wykaz literatury uzupełniającej:**
 - Jak dla wykładu
- 4. Opis sposobu wyznaczania punktów ECTS**

a. forma stacjonarna

Forma zajęć	Formy aktywności studenta	Średnia ilość godzin na zrealizowanie aktywności
Wykład	Kontakt z nauczycielem	30
	Czytanie wskazanej literatury	5
	Przygotowanie do egzaminu	4
Laboratorium	Kontakt z nauczycielem	25
	Przygotowanie do pracy kontrolnej	15
	Samodzielne rozwiązywanie zadań	10
	Czytanie wskazanej literatury	5
Konsultacje	Kontakt z nauczycielem	3
Zal./Egzamin	Kontakt z nauczycielem	3
Całkowita ilość godzin aktywności studenta		100
Liczba punktów ECTS dla modułu		4

b. forma niestacjonarna

Forma zajęć	Formy aktywności studenta	Średnia ilość godzin na zrealizowanie aktywności
Wykład	Kontakt z nauczycielem	15
	Czytanie wskazanej literatury	15
	Przygotowanie do egzaminu	9
Laboratorium	Kontakt z nauczycielem	20
	Przygotowanie do pracy kontrolnej	15
	Samodzielne rozwiązywanie zadań	10
	Czytanie wskazanej literatury	10
Konsultacje	Kontakt z nauczycielem	3
Zal./Egzamin	Kontakt z nauczycielem	3
Całkowita ilość godzin aktywności studenta		100
Liczba punktów ECTS dla modułu		4

5. Wskaźniki sumaryczne

a. forma stacjonarna

- a) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich
- Liczba godzin kontaktowych – 61
 - Liczba punktów ECTS – 2,4

b) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach o charakterze praktycznym.

- Liczba godzin kontaktowych – 25
- Liczba punktów ECTS – 2,2

b. forma niestacjonarna

a) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich

- Liczba godzin kontaktowych – 41
- Liczba punktów ECTS – 1,6

b) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach o charakterze praktycznym.

- Liczba godzin kontaktowych – 20
- Liczba punktów ECTS – 2,2

6. Zakładane efekty uczenia się

Efekt przedmiotowy (Symbol)	Efekty uczenia się dla przedmiotu	Odniesienie do kierunkowych efektów uczenia się
JPO_W1	Zna pojęcie algorytmu i programu komputerowego, główne metody i techniki programowania: programowanie proceduralne, programowanie obiektowe, programowanie strukturalne. Rozróżnia pojęcia aplikacji, apletu, serwletu	K_W04 K_W07 K_W12 K_W13
JPO_W2	Rozumie podstawowe konstrukcje programistyczne języka Java, zasady ich translacji oraz zna typy pierwotne i obiektowe Javy oraz ich wewnętrzną reprezentację	K_W04, K_W07 K_W12 K_W13
JPO_W3	Ma podstawową wiedzę dotyczącą obsługi wyjątków, strumieni oraz własnych klas, jakie można tworzyć w języka Java	K_W04, K_W07 K_W12 K_W13
JPO_W4	Ma podstawową wiedzę dotyczącą zaawansowanych elementów języka Java, w tym programowania funkcyjnego i kryptografii	K_W04, K_W07, K_W08
JPO_U1	Potrafi skonstruować algorytm rozwiązujący podany problem algorytmiczny i zapisać go w języku Java	K_U02, K_U03 K_U04, K_U11 K_U12, K_U17 K_U23, K_U24
JPO_U2	Potrafi zastosować podstawowe i obiektowe konstrukcje programistyczne Javy	K_U02, K_U03 K_U04, K_U11 K_U12, K_U17 K_U23, K_U24
JPO_U3	Potrafi sprawdzić niezawodność programu komputerowego za pomocą testowania w wybranym środowisku programistycznym i udokumentować program	K_U02, K_U03 K_U04, K_U11 K_U12, K_U17 K_U23, K_U24
JPO_K1	Potrafi pracować w zespole projektowo-programistycznym	K_K02

7. Odniesienie efektów uczenia się do form zajęć i sposób oceny osiągnięcia przez studenta efektów uczenia się.

Efekt przedmiotowy (Symbol)	Forma zajęć		Sposób sprawdzenia osiągnięcia efektu
	Wykład	Ćwiczenia	
JPO_W1	x		praca kontrolna egzamin
JPO_W2	x	x	praca kontrolna egzamin
JPO_W3	x	x	praca kontrolna egzamin
JPO_W4	x	x	praca kontrolna egzamin
JPO_U1		x	praca kontrolna egzamin
JPO_U2		x	praca kontrolna egzamin
JPO_U3		x	praca kontrolna wykonanie implementacji aplikacji
JPO_K1	x	x	prezentacja

8. Kryteria uznania osiągnięcia przez studenta efektów uczenia się.

Efekt przedmiotowy (Symbol)	Efekt jest uznawany za osiągnięty, gdy:
JPO_W1	Student poprawnie rozwiąże zadanie egzaminacyjne sprawdzające wiedzę dotyczącą podejścia obiektowego w programowaniu w języku Java.
JPO_W2	Student poprawnie rozwiąże zadanie egzaminacyjne sprawdzające wiedzę na temat wykorzystania typów pierwotnych i klas udostępnianych w języku Java.
JPO_W3	Student poprawnie rozwiąże zadanie egzaminacyjne sprawdzające wiedzę na temat wykorzystania w języku Java dziedziczenia, hermetyzacji i polimorfizmu.
JPO_W4	Student poprawnie rozwiąże zadanie egzaminacyjne sprawdzające wiedzę na temat wykorzystania w języku Java elementów programowania funkcyjnego i kryptografii.
JPO_U1	Praca kontrolna zawiera poprawną konstrukcję aplikacji języka Java, w pełni zgodnej ze specyfikacją określoną przez prowadzącego.
JPO_U2	Praca kontrolna zawiera poprawną konstrukcję hierarchii klas języka Java na podstawie przygotowanego przez prowadzącego opisu wymagań
JPO_U3	Przygotowane opracowanie zawiera poprawny opis budowy aplikacji, dokumentację kodu programu oraz opis testów.
JPO_K1	Wykonana prezentacja zawiera opis kolejnych etapów realizacji projektu: specyfikację wymagań, założenia projektowe, architekturę systemu, kod programu, opis testów z podziałem na zadania dla poszczególnych członków zespołu projektowego