**Nazwa przedmiotu:**

Paradygmaty programowania

**Koordynator przedmiotu:**

Rajmund Kożuszek

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

PARP

**Semestr nominalny:**

6 / rok ak. 2021/2022

**Liczba punktów ECTS:**

3

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. liczba godzin kontaktowych – 46 godz., w tym
 obecność na wykładach: 15 godz.,
 obecność na zajęciach laboratoryjnych: 30 godz.,
 udział w konsultacjach związanych z realizacją przedmiotu: 1 godz.
2. praca własna studenta – 42 godz., w tym
 analiza literatury i materiałów wykładowych związana z przygotowaniem do kolejnych wykładów: 8 godz.
 wykonanie ćwiczeń przygotowawczych do laboratorium: 10 godz.
 wykonanie zadania projektowego: 16 godz.
 przygotowanie do kolokwium: 8 godz.

Łączny nakład pracy studenta wynosi 88 godz., co odpowiada 3 pkt. ECTS.

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1.75 pkt. ECTS, co odpowiada 46 godz. kontaktowym

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

2 pkt. ECTS, co odpowiada 40 + 16 = 56 godz. realizacji ćwiczeń

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 15h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 30h |
| Projekt:  | 0h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Programowanie aplikacyjne (PAP), Wprowadzenie do sztucznej inteligencji (WSI), Sztuka wytwarzania oprogramowania (SWO)

**Limit liczby studentów:**

120

**Cel przedmiotu:**

Celem przedmiotu jest przedstawienie podstawowych paradygmatów programowania oraz pokazanie w jaki sposób mogą być one wykorzystane przy tworzeniu oprogramowania. Wskazane zostaną kluczowe różnice pomiędzy paradygmatami, obszary ich zastosowań oraz wspierające je języki programowania. Szczególna uwaga poświęcona zostanie programowaniu funkcyjnemu jako tematowi nie poruszanemu na wcześniejszym etapie studiów. Wiedza uzyskana podczas przedmiotu pozwoli studentom na dobór paradygmatu do rozwiązywanego problemu programistycznego oraz ułatwi efektywne posługiwanie się współczesnymi językami programowania. Wybrane zagadnienia będą na wykładzie prezentowane w sposób interaktywny.

**Treści kształcenia:**

WYKŁADY:

1. Wprowadzenie (2 godz.)
Informacje o przedmiocie. Regulamin przedmiotu. Ogólna klasyfikacja paradygmatów i języków programowania. Programowanie imperatywne a programowanie deklaratywne.

2. Programowanie obiektowe (2 godz.)
Programowanie obiektowe na przykładzie języka Smalltalk: klasy i obiekty, klasy jako obiekty, przekazywanie komunikatów, dziedziczenie i polimorfizm, bloki kodu, refleksja, wzorzec MVC, obrazy i maszyna wirtualna.

3. Programowanie funkcyjne (6 godz.)
Programowanie funkcyjne na przykładzie języka Haskell: funkcje czyste, funkcje rekurencyjne, funkcje wyższego rzędu, dopasowanie wzorca, funkcje anonimowe, klasy typów, funktory i monady, proste programy interaktywne, leniwe wartościowanie, wnioskowanie na temat programów.

4. Programowanie w logice (2 godz.)
Programowanie w logice na przykładzie języka Prolog: fakty i reguły, zmienne i ich ukonkretnianie, wykonywanie zapytań, listy, proste programy interaktywne.

5. Pozostałe paradygmaty (2 godz.)
Programowanie zdarzeniowe, programowanie reaktywne, programowanie aspektowe, programowanie agentowe, model aktorów.

LABORATORIA:

W trakcie zajęć laboratoryjnych studenci nabywają praktycznych umiejętności rozwiązywania problemów przy użyciu omawianych na wykładzie paradygmatów programowania. Realizują kilka mniejszych zadań w językach Smalltalk, Haskell i Prolog oraz jedno większe zadanie projektowe, w ramach którego implementują rozwiązanie korzystając z kilku paradygmatów i poznając w praktyce ich zalety i wady.

**Metody oceny:**

Realizacja przedmiotu obejmuje następujące formy zajęć:
- wykład prowadzony w wymiarze 2 godz. tygodniowo przez pierwszą część semestru; na wykładzie, oprócz użycia statycznych slajdów, przewidziane jest prezentowanie na żywo efektów działania omawianego kodu oraz aktywizacja studentów poprzez wspólne rozwiązywanie krótkich zadań,
- zajęcia laboratoryjne w wymiarze 2 godz. tygodniowo; w ramach tych zajęć student, korzystając z oprogramowania i sprzętu komputerowego, będzie realizował pod opieką prowadzącego wskazane zadania.

Sprawdzanie założonych efektów kształcenia realizowane jest przez:
 ocenę wiedzy i umiejętności związanych z realizacją zadań laboratoryjnych – ocena z wybranych zadań laboratoryjnych oraz zadania projektowego;
 ocenę wiedzy wykazanej na kolokwium pisemnym.

**Egzamin:**

nie

**Literatura:**

• M. Gabrielli, S. Martini, Programming Languages: Principles and Paradigms, Springer-Verlag (2010)
• M. L. Scott, Programming Language Pragmatics (fourth edition), Morgan Kaufmann (2015)
• G. Hutton, Programming in Haskell (second edition), Cambridge University Press (2016)

**Witryna www przedmiotu:**

https://usosweb.usos.pw.edu.pl/kontroler.php?\_action=katalog2/przedmioty/pokazPrzedmiot&prz\_kod=103A-INxxx-ISP-PARP

**Uwagi:**

(-)

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W01:**

zna koncepcję programowania obiektowego oraz posiada wiedzę na temat programowania w języku Smalltalk

Weryfikacja:

kolokwium, laboratorium

**Powiązane charakterystyki kierunkowe:** W08

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W02:**

zna koncepcję programowania funkcyjnego oraz posiada wiedzę na temat programowania w języku Haskell

Weryfikacja:

kolokwium, laboratorium

**Powiązane charakterystyki kierunkowe:** W08

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W03:**

zna koncepcję programowania w logice oraz posiada wiedzę na temat programowania w języku Prolog

Weryfikacja:

kolokwium, laboratorium

**Powiązane charakterystyki kierunkowe:** W08

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W04:**

zna podstawowe różnice pomiędzy różnymi paradygmatami programowania

Weryfikacja:

kolokwium, laboratorium

**Powiązane charakterystyki kierunkowe:** W08

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U01:**

potrafi tworzyć proste programy w języku Smalltalk

Weryfikacja:

kolokwium, laboratorium

**Powiązane charakterystyki kierunkowe:** U04, U01

**Powiązane charakterystyki obszarowe:** III.P6S\_UW.o, P6U\_U, I.P6S\_UW.o

**Charakterystyka U02:**

potrafi tworzyć proste programy w języku Haskell

Weryfikacja:

kolokwium, laboratorium

**Powiązane charakterystyki kierunkowe:** U01, U04

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o

**Charakterystyka U03:**

potrafi tworzyć proste programy w języku Prolog

Weryfikacja:

kolokwium, laboratorium

**Powiązane charakterystyki kierunkowe:** U01, U04

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K01:**

jest gotów do stałego aktualizowania i poszerzania posiadanej wiedzy

Weryfikacja:

kolokwium, laboratorium

**Powiązane charakterystyki kierunkowe:** K03

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KK, I.P6S\_KR