**Nazwa przedmiotu:**

Specyfikacje formalne i programy funkcyjne

**Koordynator przedmiotu:**

dr inż. Marcin Szlenk

**Status przedmiotu:**

Fakultatywny ograniczonego wyboru

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne - zaawansowane

**Kod przedmiotu:**

SPOP

**Semestr nominalny:**

3 / rok ak. 2015/2016

**Liczba punktów ECTS:**

4

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

- udział w wykładach: 15 x 2h = 30h,
- przygotowanie do wykładów (przejrzenie slajdów i dodatkowej literatury): 12h,
- przygotowanie do kolokwiów (rozwiązanie przykładowych zadań, udział w konsultacjach): 2 x 7h + 2h = 16h,
- udział w zajęciach laboratoryjnych: 4 x 2h = 8h,
- przygotowanie do zajęć laboratoryjnych (przejrzenie slajdów, rozwiązanie przykładowych zadań): 4 x 2h = 8h,
- realizacja zadania projektowego (praca indywidualna, udział w konsultacjach): 25h + 1h = 26h
Suma: 30 + 12 + 16 + 8 + 8 + 26 = 100h

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

30 + 2 + 8 + 1 = 41h, co odpowiada 1.5 punktom ECTS

**Język prowadzenia zajęć:**

polski

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

8 + 8 + 26 = 42h, co odpowiada 1.5 punktom ECTS

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

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

**Wymagania wstępne:**

Podstawowa wiedza z zakresu logiki (znajomość rachunku zdań i predykatów) oraz teorii mnogości. Umiejętność programowania strukturalnego i obiektowego.

**Limit liczby studentów:**

58

**Cel przedmiotu:**

Celem przedmiotu jest zapoznanie studentów z programowaniem w językach funkcyjnych (na przykładzie języka Haskell) oraz z możliwościami wykorzystania języków formalnych do zapisywania specyfikacji oprogramowania (na przykładzie języka Alloy oraz sieci Petriego).

**Treści kształcenia:**

Treść wykładu:

1. Programowanie funkcyjne (14h):
programowanie imperatywne a programowanie funkcyjne, programowanie w języku Haskell: proste typy danych, listy, krotki, typy polimorficzne i parametryzowane, definiowanie typów użytkownika, typy rekurencyjne, klasy typów, definiowanie funkcji i wartości, definiowanie operatorów, lambda abstrakcje, funkcje wyższego rzędu, leniwe wyliczanie, nieskończone struktury danych, operacje wejścia/wyjścia i programy interaktywne, modularyzacja programów, przykłady.

2. Modelowanie oprogramowania przy użyciu języka Alloy (8h):
elementy notacji języka Alloy: relacje, operatory, kwantyfikatory, sygnatury i pola sygnatur, predykaty, funkcje, asercje, fakty; automatyczna analiza modeli przy użyciu narzędzia Alloy Analyzer, porównanie języka Alloy oraz UML, przykłady zastosowania.

3. Modelowanie oprogramowania przy użyciu sieci Petriego (6h):
podstawowa definicja sieci Petriego i jej reprezentacja graficzna, właściwości sieci, budowa i analiza drzewa osiągalności sieci, analiza sieci przy użyciu metody niezmienników, przykłady zastosowania, przegląd innych rodzajów sieci Petriego: sieci priorytetowe, sieci czasowe, sieci kolorowane.

4. Kolokwium (2h): zadania.

Zakres laboratorium:

W ramach laboratorium studenci nabierają praktycznych umiejętności programowania w języku Haskell, realizując kilka prostych zadań (pierwsze trzy zajęcia laboratoryjne) oraz pisząc jeden większy program jako zadanie projektowe. Szczegółowa treść zadań jest określana przez prowadzących i w przypadku większego programu może uwzględniać indywidualne zainteresowania studentów. Odrębnym zadaniem laboratoryjnym (czwarte zajęcia laboratoryjne) jest napisanie specyfikacji zadanego systemu w języku Alloy oraz jej analiza z użyciem narzędzia Alloy Analyzer.

**Metody oceny:**

Zaliczenie przedmiotu odbywa się na podstawie ocen z dwóch kolokwiów (pierwsze dotyczy programowania funkcyjnego, drugie języka Alloy i sieci Petriego) oraz oceny z projektu w języku Haskell. Za każde z kolokwiów można maksymalnie otrzymać 15 pkt., natomiast za projekt - 20 pkt. Łącznie można uzyskać 50 pkt. Uzyskana liczba punktów przekłada się na ostateczną ocenę w sposób następujący: ocena 2 (mniej niż 26 pkt.); 3 (26 - 30 pkt.); 3,5 (31 - 35 pkt.); 4 (36 - 40 pkt.); 4,5 (41 - 45 pkt.); 5 (co najmniej 46 pkt.).

**Egzamin:**

nie

**Literatura:**

1. Graham Hutton, Programming in Haskell, Cambridge University Press, 2007
2. Daniel Jackson, Software Abstractions: Logic, Languages, and Analysis, The MIT Press, 2012
3. Marcin Szpyrka, Sieci Petriego w modelowaniu i analizie systemów współbieżnych, WNT, 2008

**Witryna www przedmiotu:**

www.ia.pw.edu.pl/~mszlenk

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt 1:**

Student, który zaliczył przedmiot, rozumie na czym polega programowanie funkcyjne oraz posiada wiedzę na temat programowania w języku Haskell

Weryfikacja:

Ocena wyników pierwszego kolokwium

**Powiązane efekty kierunkowe:** K\_W07, K\_W08

**Powiązane efekty obszarowe:** T2A\_W05, T2A\_W07

**Efekt 2:**

Student, który zaliczył przedmiot, posiada wiedzę na temat możliwości zapisywania i weryfikacji formalnych specyfikacji systemów przy użyciu języka Alloy oraz sieci Petriego

Weryfikacja:

Ocena wyników drugiego kolokwium

**Powiązane efekty kierunkowe:** K\_W02, K\_W05, K\_W07, K\_W08

**Powiązane efekty obszarowe:** T2A\_W03, T2A\_W04, T2A\_W05, T2A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt 3:**

Student, który zaliczył przedmiot, potrafi programować w języku Haskell

Weryfikacja:

Ocena wyników pierwszego kolokwium oraz ocena realizacji projektu

**Powiązane efekty kierunkowe:** K\_U05

**Powiązane efekty obszarowe:** T2A\_U07, T2A\_U09

**Efekt 4:**

Student, który zaliczył przedmiot, potrafi napisać prostą specyfikację w języku Alloy

Weryfikacja:

Ocena wyników drugiego kolokwium

**Powiązane efekty kierunkowe:** K\_U05

**Powiązane efekty obszarowe:** T2A\_U07, T2A\_U09