KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Course title

Functional Programming

ECTS code

11.3.1510

Name of unit administrating study

Faculty of Mathematics, Physics and Informatics

Studies

faculty	field of study		
Faculty of Mathematics, Physics and Informatics	Informatics	type	second tier studies (MA)
		form	full-time
		specialty	all
		specialization	all

Teaching staff

prof. UG, dr hab. Christoph Schwarzweller

Forms of classes, the realization and number of hours

Forms of classes

Laboratory classes, Lecture

The realization of activities

classroom instruction

Number of hours

Lecture: 30 hours, Laboratory classes: 30 hours

ECTS credits

7

Course in form of a 30h lecture and 30h laboratory + student's own work.

The academic cycle

2021/2022 winter semester

Type of course

an elective course

Language of instruction

polish

Teaching methods

- conducting experiments
- designing experiments
- multimedia-based lecture

Form and method of assessment and basic criteria for eveluation or examination requirements

Final evaluation

- Graded credit
- Examination

Assessment methods

- written exam with open questions
- graded course credit based on individual grades obtained during the semester
- written exam (long written answer/problem solving)

The basic criteria for evaluation

colloquium after the laboratory
written exam

Method of verifying required learning outcomes

Required courses and introductory requirements

A. Formal requirements

no formal requirements

B. Prerequisites

no starting requirements

Aims of education

Introduction to functional programming based on the progrmming language Haskell

Course contents

1. Introduction
2. Introduction to Haskell

3. Types and type classes
4. Theoretical basics of functional programming
5. More techniques and applications of Haskell

Bibliography of literature

Hutton; Programming in Haskell
Tompson; Haskell: The Craft of Programming
Bird; Introduction to Functional Programming
Abelson, Sussman; Structure and Interpretation of Computer Programs

The learning outcomes (for the field of study and specialization)

K_W02: has in-depth knowledge of formal languages, calculation models and computational complexity issues; knows the formal apparatus allowing for the formulation and testing of the properties of IT objects
K_W03: has in-depth knowledge of programming paradigms and advanced programming constructs; knows the current trends in programming languages
K_W04: knows complex data structures and advanced methods of algorithmic solving of computationally difficult problems (exponential algorithms, approximation, heuristics)

K_W06: knows well the rules of occupational health and safety in the IT profession
K_U03: designs, analyzes in terms of correctness and computational complexity and builds algorithms using advanced programming techniques and data structures
K_U05: can apply known algorithms in specific situations, can effectively select the type of algorithm depending on the problem posed

Knowledge

Student:
knows paradigm functional programming
knows model functional programming
knows programming language Haskell and its type system

Skills

Student:
- develops algorithms using paradigm functional programming
- is able to solve problems using functional programming languages

Social competence

Student:
- is able to work with english literature
- knows copyright rules for writing computer programs

Contact

schwarzw@inf.ug.edu.pl