Course title: Paradigms of Programming

Neptun code:

GEIAL401-a

Course coordinator: Dr. Péter Mileff, PhD, associate professor

type of lesson and number of lessons: **lecture (2)**

method of evaluation: colloquium

curriculum location of the subject: (autumn/spring semester): autumn and spring

pre-study conditions *(if any)*: -

The task and purpose of the subject:

The objective of the course is to familiarize students with programming paradigms that go beyond university studies. The course will provide an in-depth introduction to various paradigms, the associated environments, and the necessity of transitions. Through practical examples, modern directions will be presented along with their advantages and disadvantages. An additional important part of the curriculum covers design patterns, where the most important patterns will be introduced in addition to an overview of the field.

Course description:

Von Neumann architecture. Structure of programming languages, structure of programs. Meaning of paradigm. Imperative programming paradigm and its most important programming elements. Imperative paradigm Part II. Types, orthogonality of declarations. Procedural programming: data types, elementary programs, subprograms, exception handling. Logic languages. Principles, notation system. Rules, Horn clauses, logical variables. Object-oriented programming. Principles, inheritance, overloading. Object orientation from a software technology perspective. Complex case study in the object-oriented pattern. Functional programming. Principles, functions, types, polymorphism, higher-order functions. Lazy evaluation. Miranda, LISP. Component-Entity design model. Principles, components, entities, relationships. Converting an object-oriented inheritance tree to a component-entity model. Modern directions, techniques, and trends in the world of programming. The quantum computer. Principles, elements. Practical examples.

Required literature:

1. Peter Van Roy, Seif Haridi: Concepts, Techniques, and Models of Computer Programming, 2004
2. Maurizio Gabbrielli: Programming Languages: Principles and Paradigms, 2010
3. Daniel Cook: Programming Paradigms: Understanding Procedural, Object-Oriented, and Functional Techniques, 2023

Recommended literature:

1. Peter Van Roy, Seif Haridi: Concepts, Techniques, and Models of Computer Programming, 2004
2. Maurizio Gabbrielli: Programming Languages: Principles and Paradigms, 2010
3. Daniel Cook: Programming Paradigms: Understanding Procedural, Object-Oriented, and Functional Techniques, 2023