| | |
|---|---|
| **Level**: bachelor | |
| **Course title:** Object-oriented programming 1 (I032) | |
| **Status:** obligatory | |
| **ECTS:** 7 | |
| **Requirements:** none | |

**Learning objectives**

Introducing students to the basic principles and techniques of object-oriented programming, and enabling them to apply the proven solutions and contemporary tools in the development of a wide spectrum of applications.

**Learning outcomes**

*Minimal:* At the end of the course, successful students should be able to demonstrate the ability to understand and analyze problems, as well as the design and realization of the solution in a concrete object-oriented programming language.

*Optimal:* At the end of the course, successful students should be able to understand, analyze and define problems based on logical foundations, as well as creative design and realization of solutions using the object-oriented style of programming.

**Syllabus**

*Theoretical instruction*

Object-oriented methodology: design and programming. Basic elements of object-oriented programming: classes, inheritance, dynamic bounding. A review of some hybrid object-oriented languages. Object-oriented programming language. Structure and parts of the program. Simple data types, complex data types, classes and objects. Association, expressions. Control and loop commands. Referential data types. Packages. Exceptions.

*Practical instruction*

Structure of a program in a particular object-oriented language. Testing of finished solutions, tools, discussion on the possibilities of application and the like. Individual practical tasks: simple data types, branching commands, iterative statements, termination statements, working with strings, input and output streams, classes, objects, inheritance, abstract classes, interfaces, arrays, modifiers, initializers, nested classes, packages, exceptions.

| **Weekly teaching load** | | | | Other: 0 |
|---|---|---|---|---|
| Lectures: 2 | Exercises: 4 | Other forms of teaching: 0 | Student research: 0 | |