**DESCRIPTION OF A STUDY COURSE – SYLLABUS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title of a course** | **Algorithms and Data Structures** | | | | |
| **Study programme** | **Professional undergraduate study Telematics** | | | | |
| **Status of a course** | Obligatory | | | | |
| **Year of study** | 1 | **Semester** | S | **ECTS credits** | 5 |
| **Goals of a course** | | | | | |
| Acquire competencies to work with list data structures, row stack, tree, dictionary and set, and become familiar with basic sorting and search algorithms. | | | | | |
| **Conditions for enrolling course** | | | | | |
| No conditions | | | | | |
| **Expected learning outcomes on a level of a course** | | | | | |
| 1. Apply data structures: list, row, stack, tree, dictionary, set 2. Describe sorting algorithms: bubble sort, selection sort, insertion sort, shell sort, merge sort and quick sort 3. Describe search algorithms: sequential search, block read, binary search, search tables 4. Describe algorithms using pseudocodes and program flow diagrams 5. Use a debugger in a computer program | | | | | |
| **Content of a course** | | | | | |
| Fields and vectors (syntax, semantics, multidimensional fields, development of container class); FIFO and LIFO (sleep awaiting, data arranging); Data structures for chain of characters (String, String Buffer/String tokenizer); Algorithms verification (correctness, static and dynamic finite, guarantee, verification rules, termination); Expenses and complexity (expenses account, effectiveness, comparison of algorithms, complexity classification); Recursion (divide and conquer – strategies, implementation and dynamic complexity of recursive algorithms); Chain lists (simple and double chain lists with and without empty initial and final elements, cyclic chain lists; entering, adapting and removing the elements on the list, complexity of operation on the list); Trees (structure and notions, searching, entering and removing nods in a binary tree, formation of search trees, traversation, balancing, multiple tree); Elementary search engine (sequential search, binary search, interpolative search); Browsing of data given in an unpredictable order (deduction transformations, transition linking, chart). | | | | | |