IVANOV S.A.

**DATA STRUCTURES AND ALGORITHMS.**

**HYBRID IMPLEMENTATION (DRAKON + GOLANG)**

**2023**

INTRODUCTION The basis of the processes of cognition, understanding, and construction of natural, non-natural and virtual realities, one way or another, is DATA, which can be understood as unstructured information. To structure this data in order to obtain new knowledge or new meanings, it is necessary to process them according to a certain plan of action. In a stricter definition, a data structure is a form of representation of the properties and relationships of a subject area, focused on the expression of a description of data by means of formal languages. Technologically, a data structure is a container in which information is arranged in a certain way that establishes a way to place data in computer memory for quick and efficient access to them.

In turn, algorithms are sets of step-by-step instructions for solving computational problems by processing data structures. Data structures and the processes of their processing according to various algorithms are inextricably linked with each other. In some problems, the choice of algorithm is dictated by the data structure, in others, the data structure determines the choice of algorithm. One of the founders of the formation of computer science as an academic science, N. Wirth, formulated this connection in the title of the book: "Algorithms + data structures = program" [Wirth].

The variety of objective, non-objective and virtual worlds and the processes taking place in them predetermine the emergence of many problem situations, the effective solution of which involves the choice of data structures and algorithms for their processing. A solution is considered efficient if it solves the problem within known resource constraints. First of all, this refers to the total space available for storing data, as well as the time allotted for the execution of each algorithm. When choosing a data structure for solving a problem, one should first determine the basic operations that must be supported, secondly, to quantify the resource constraints for each operation of the algorithm (occupied memory and execution time of operations).

A data structure requires a certain amount of memory for each stored data element, a certain amount of time to perform one basic operation, and a certain amount of programming effort. Each task has limitations on available space and time. Each solution to a problem uses the underlying operations in some relative proportion, and the data structure selection process must take this into account. Only after a thorough analysis of the characteristics of the problem being solved, it is possible to determine the best data structure for it[].

Based on the considerations presented, the Handbook provides basic information about the most commonly used data structures and algorithms, since they form the basis of the professional activity of a programmer of any level. Considerable attention is paid to the issues of evaluating the efficiency of data structures and algorithms in terms of occupied memory and the execution time of basic operations. The organization of data structures and the study of basic algorithms is carried out in the Golang programming language and is accompanied by explanatory illustrations. In a number of important cases, the algorithm visualization technology based on the Drakon graphical language is used.

**SECTION 1** reveals the meaning of the main concepts in the discipline Data Structure and Algorithms: data, data type, abstract data types, algorithms. A high level of competence in the development of efficient algorithms is necessary for every programmer, not only when solving practical problems related to the use of IT technologies. Knowledge in the field of data structure and algorithms is the most important component in an interview when applying for a job in a large IT company.

**SECTION 2** describes the basics of the Golang programming language in sufficient amount for acquaintance and practical mastery of knowledge and skills in the field of data structure and algorithms. Of course, basic concepts and a certain level of programming are assumed for reading this chapter.

**SECTION 3** gives a brief description of the visual language DRAKON, which allows to represent any complex algorithms in the form of a DRAKON-diagram using icons, corresponding to the basic constructions of the Golang language. The technology of automated transformation of the DRAKON-diagram into the program code is given in the editor DRAKON WEB Editor, with subsequent implementation in the shell of Visual Studio Code. Actually, this is a hybrid, two-stage approach: in the first step a DRAKON-diagram is made, the icons of which are filled with constructions of the language Golang, on the second step - automatically generated software code, implemented in any software environment

**SECTION 4** presents Golang language constructs that implement the basic abstract data types described in Section 1: array, cut, list etc. Explanations are given regarding the use of appropriate Golang language constructs.

**SECTION 5** provides basic insights into the analysis of algorithm complexity and the evaluation and effectiveness of algorithms from the perspective of computer memory (space) and time costs. Theoretical ideas about estimating the complexity of algorithms are decisive when choosing the most effective one.

**SECTION 6** deals with basic sorting algorithms. Each algorithm is represented by a corresponding DRAKON-diagram, accompanied by necessary explanations about the processing process and estimation of complexity.

**SECTION 7** discusses basic algorithms for searching elements in different data structures (slices, lists). In addition, the section includes a description and DRAKON-diagram of hash algorithms that significantly reduce search time by reducing the number of element value comparisons.

**SECTION** 8 provides conceptual information about binary trees – nonlinear data structures, including binary search trees and self-balancing trees. This section describes the basic DRAKON-diagrams of algorithms for inserting new ones, finding nodes with specified values and deleting nodes.

**SECTION 9** provides basic information about the most complex data structures – graphs. Basic concepts of graph theory and their basic types are given. Algorithms of representation of graphs, their traversal, choice of path between vertices are considered.