A. Torbunova, student
Yu. Lytvyna, PhD in Pedagogy,
As. Prof., research advisor
Tavria State Agrotechnological University

TURING MACHINE AS AN INTEGRAL PART IN FORMATION AND DEVELOPMENT OF PROGRAMMING LANGUAGES

In the article a Turing machine, the theoretical basis for algorithms and scientific base of programming is presented. Special attention is paid to study of Turing machine algorithms, the essence of the algorithm and its influence on the development of information technology. The importance of knowledge on a Turing machine for future programmers is emphasized.

Modern development of information technology and computers are considered to be completely different from the mechanism of a Turing machine, but this invention became the theoretical basis for algorithms and scientific base of programming and it influenced the formation of high-level languages.

A Turing machine became a great invention that marked the beginning of the era of information technology and predicted the architecture of modern computer systems. It was named after the British mathematician Alan Turing, the great English mathematician, the founder of computer science, who proposed the concept in 1936. The genius invention of Alan Turing was successfully applied by British Analytical Crypto Office during World War II to break German secret codes [3].

A Turing machine is an abstract machine that manipulates symbols on a strip of tape according to a table of rules; to be more exact, it is a mathematical model of computation that defines such a device[3]. In other words, it is the specific sequence of elementary actions to solve a problem. We face such algorithms every day. We have a certain algorithm for cooking, planting trees, using computers etc. The principle of solving complex mathematical problems by designing various abstract mechanisms and building algorithms formed the basis of the origin of information technology.

Thus, we think this knowledge is really essential for a programmer. Before learning a programming language, a student has to study Turing machine algorithms. A programmer will be able to understand the essence of an algorithm and it will help him to understand a set task easily.

It should be emphasized that any programming language consists of the sequence of steps in forming a certain solution algorithm, that's why Turing machine is not very good for programming because of lack of instruction set basic arithmetic operations [2, c.102].

The significance of Alan Turing and his invention for mankind are irreplaceable. In 1982, scientists at the University of Toronto created more powerful programming language than Pascal and they called it Turing. The most famous association ACM (Association for Computing Machinery, established in 1947) introduced the award named after Alan Turing [3].

A Turing machine is very simple, but you can perform virtually any application based on precise algorithms. To perform various computation operations, there is a

special table with certain recorded rules representing a set of universal instructions for a machine. Applying this table, which has a fixed procedure for a particular combination of different classes and characters, a device determines which processing operation should be done in each situation [1, c.56]. In fact, the Universal Turing Machine is the first prototype of modern computers.

To sum up, despite the fact that these programming algorithms of a Turing machine are not suitable for computation in practice as it does not use random-access memory, but they help to understand the essence of algorithm sequence of a computer code. We believe that before starting program writing, students have to comprehend the essence of the algorithm to develop the algorithm's logic.

REFERENCES

1. Хопкрофт Джон «Введення в теорію автоматів, мов і обчислень» / Джон Хопкрофт, Раджив Мотвані, Джеффрі Ульман. — М.: «Вільямс», 2002. — 463с.

2. Сінтес Антоні «Опануй самостійно об'єктно-орієнтоване програмування» / Антоні Сінтес. — М.: В«Вільямс», 2002. — 513 с.

3. https://en.wikipedia.org/wiki/Turing_machine

4. Herken, Rolf (1995), The Universal Turing Machine – A Half-Century Survey, Springer Verlag, ISBN 3-211-82637-8.