

KRAVCHUK OLHA

Khmelnytsky National University

<https://orcid.org/0000-0001-6937-5001>e-mail: kravchukoa2@gmail.com**SYNYUK NATALIA**

Khmelnytsky National University

<https://orcid.org/0000-0002-6889-1351>e-mail: nataliya.synjuk@khmnu.edu.ua**KRAVCHUK DENYS**

Khmelnytsky National University

<https://orcid.org/0009-0005-0806-8426>e-mail: lklokiiitstep@gmail.com

REVIEW AND ANALYSIS OF DATA PROCESSING INFORMATION TECHNOLOGIES IN THE COURSE OF MODERN COMPUTER SCIENCE

The article focuses on the review and analysis of data processing information technologies in the course of modern computer science. The article also presents theoretical aspects of such concepts as algorithms, data structures, computer architecture, logic, software, programming languages and concepts of their development, which have a significant impact on technological progress and social development, which, in turn, contribute to innovations in various fields, from medicine to finance, making life easier and improving working conditions, creating new business sectors and labor markets that meet the requirements of the digital economy, changing the way people communicate, educate and interact in society, in particular through the development of social networks and Internet platforms. Information technology has become not only the basis of modern computer science, but also a key determinant of technological progress and societal development.

Understanding their role and interconnection helps to unlock the potential of these industries for further development and achievement of new heights in technology and innovation. Informatics, as the science of information processing, and information technologies that provide tools for this processing, form the basis for modern programming. They provide not only the necessary knowledge and tools, but also define strategies and methodologies for software development. Computer science covers many aspects, such as algorithms, data structures, computing theory, and much more. However, at the first stages of study, the basics of computer science usually include concepts such as algorithms, data structure, logic, computer architecture, programming languages, and software. They provide not only the necessary knowledge and tools, but also define strategies and methodologies for software development. Computer science covers many aspects, such as algorithms, data structures, computing theory, and much more. However, at the first stages of study, the basics of computer science usually include concepts such as algorithms, data structure, logic, computer architecture, programming languages, and software, which are discussed in this article.

Keywords: computer science, data processing, information technology, algorithms, data structures, computer architecture, logic, software, programming languages.

КРАВЧУК ОЛЬГА, СИНЮК НАТАЛІЯ, КРАВЧУК ДЕНІС

Хмельницький національний університет

ОГЛЯД ТА АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ОБРОБКИ ДАНИХ В КУРСІ СУЧАСНОЇ ІНФОРМАТИКИ

Стаття зосереджується на огляді та аналізі інформаційних технологій обробки даних в курсі сучасної інформатики, а також в статті наведені теоретичні аспекти таких понять як алгоритми, структури даних, комп'ютерна архітектура, логіка, програмне забезпечення, мови програмування та концепції їх розвитку, що мають значний вплив на технологічний прогрес та суспільний розвиток, які, в свою чергу, сприяють інноваціям у різних галузях, від медицини до фінансів, що полегшують життя та покращують умови праці, створенню нових галузей бізнесу та ринків праці, що відповідають вимогам цифрової економіки, зміні у способі спілкування, освіти та взаємодії у суспільстві, зокрема через розвиток соціальних мереж та інтернет-платформ. Інформатика та інформаційні технології стали невід'ємною основою для сучасного програмування, знання цих основ дуже важливе для майбутніх програмістів. А розуміння їхньої ролі та взаємозв'язку допомагає розкрити потенціал цих галузей для подальшого розвитку та досягнення нових вершин у технологіях та інноваціях.

Ключові слова: інформатика, обробка даних, інформаційні технології, алгоритми, структури даних, комп'ютерна архітектура, логіка, програмне забезпечення, мови програмування.

Statement of the problem

Computer science and information technology have become an integral part of our modern life. They permeate almost all spheres of human activity, including business, education, medicine, and, of course, programming. Computer science is considered not only the science of information processing and transmission, but also the skill that underlies modern programming. Thanks to computer science, we understand how computers work, how they process data, and how we can use this information to create programs and systems. In this article, we'll look at how computer science and information technology have become an integral part of modern programming and how this interaction affects technological progress and the development of society, as

well as how computer science is the key to learning programming and why knowing its basics is very important for future programmers.

The purpose of this article is to review and analyze computer science and information technology as the foundations of modern programming.

Presentation of main material

Computer science, as the science of information processing, and information technology, which provides tools for this processing, form the basis for modern programming [1]. They provide not only the necessary knowledge and tools, but also define strategies and methodologies for software development. Computer science covers many aspects, such as algorithms, data structures, computing theory, and much more. However, at the first stages of study, the basics of computer science usually include such concepts as algorithms, data structure, logic, computer architecture, programming languages, and software. Let's consider and characterize them [3,5,7,8,9].

Algorithms are a sequence of steps that are performed to solve a particular problem. Developing efficient algorithms is a key aspect of programming. Knowledge of algorithms and data structures is the foundation of any programmer. They determine the efficiency and productivity of the programs being developed. Algorithms are a well-defined set of instructions or procedures that are executed step by step to solve a particular problem or achieve a certain goal. They are the basic element for solving various tasks in computer science and programming. Algorithms can be written in the form of a sequence of steps or displayed in the form of flowcharts, diagrams, or pseudocode. The main characteristics of algorithms include: clarity (the algorithm should be understandable and easily readable for people, each step should be clearly described and leave no room for ambiguity or misunderstanding); correctness (the algorithm should solve the task correctly, which means it should return the correct result for any input data and in all possible execution conditions); efficiency (an algorithm should be efficient in terms of time and resources required to execute it, which means it should work quickly and efficiently even for large amounts of data or complex tasks); versatility: a good algorithm can be applied to different types of input data and different situations, it should be general and meet the needs of different tasks); modularity (an algorithm can be divided into separate parts (modules) that can be executed separately and can be reused in different contexts); stability (the algorithm should produce a stable result for the same input data at each execution, without unfair or unpredictable changes in the result); scalability (a good algorithm should be able to work efficiently even for very large amounts of data or a large number of operations); usability (the algorithm should be implemented in practical program code and be available for use in real-world applications) and the ability to execute in a finite time. These characteristics help determine the quality of an algorithm and its suitability for solving specific tasks. Algorithms are used for a variety of purposes, including sorting data, finding a path in a graph, optimizing tasks, and much more [2,3,8].

Data structures are ways of organizing and storing data in a computer so that it can be efficiently processed and accessed. For example, arrays, lists, stacks, queues, etc. They are a key element in programming because they determine how data will be represented and processed in programs. Let's look at some of the main aspects of data structures. Data organization: Data structures define how data will be organized and stored in computer memory. This can include lists, arrays, trees, graphs, and other structures. Access efficiency: One of the main characteristics of data structures is their speed and efficiency of data access. Some data structures can be optimized for quick access to items or for quick searches. Data operations: Many data structures support various operations such as insert, delete, search, and sort. It is important that these operations are efficient and take a reasonable amount of time. Access methods: Data structures can have different methods of accessing their elements. For example, some data structures support access to elements by index, while others support access by key or path. Memory usage: It is important to consider memory usage when using data structures. Some structures can be more efficient in terms of memory usage than others. Suitability for use: The choice of a particular data structure depends on the specific task or use case. Some data structures may be more suitable for certain operations or data types than others. In summary, data structures provide a convenient and efficient way to organize, store, and process data in programs. They are a key element in programming and allow developers to create fast and efficient programs to solve a variety of tasks.

Logic. Logic is an important aspect of computer science, especially in programming. Computer science studies the principles of logic and the conditional constructs that allow programs to make decisions. Logic defines the rules and principles of thinking that are used to solve problems and make decisions in programs. Let's take a closer look at this aspect.

Logic principles are logical operations:

- logical and (and): returns true only if both expressions are true;
- boolean or: returns true if at least one of the expressions is true;
- logical not (not): changes the truth to false and vice versa.

Conditional statements:

- if-else statement: used to execute code if a certain condition is true;
- switch-case: allows you to execute different code depending on the value of the expression.

Loops and repetition:

- while loop: executes while the condition is true;
- for loop: used to repeat a block of code a specified number of times.

Logical thinking:

- program developers must have the ability to think logically and analyze problems using formal logic.

The importance of logic in programming: decision-making (logic allows programs to make decisions based on conditions and circumstances), error avoidance (logical programming helps to avoid errors and wrong decisions in programs); code efficiency (using conditional statements and logical operations helps to make code more efficient and readable); algorithms (developing effective algorithms requires logical thinking and the ability to make the right decisions at every step).

Thus, logic in programming plays a critical role, as it determines the ways of making decisions and the conditions of code execution, which affects the efficiency, reliability, and correctness of programs.

Computer architecture. Computer architecture is the structure and organization of computer system components that defines its functionality and how they interact. It is a field of computer science that studies how computers are built and operate in terms of hardware and software. Let's look at some key aspects of computer architecture.

Firstly, it is the hardware architecture: the processor (CPU) - the central processor executes instructions and controls the computer; memory - RAM for storing data and programs, and permanent memory (e.g., hard disk or solid-state drive) for storing information even after the computer is turned off; data bus - provides communication between different components, data and instructions.

Secondly, it is system software: the operating system (manages computer resources and provides an environment for executing programs), drivers (software that provides an interface between hardware and the operating system), utilities (auxiliary programs for diagnosing, debugging, and maintaining a computer system).

Third, network architecture: network connectivity (allows computers and other devices to share data and resources); protocols and standards (define the rules for exchanging data between devices on a network).

Fourthly, it is system development: integration and testing (the process of combining different components and testing their interaction); scalability (the ability of the system to work efficiently when the amount of data or load increases).

Fifth, security and privacy: encryption and authentication (security measures to protect data from unauthorized access); protection against malicious programs (anti-virus software and other security measures to prevent vulnerabilities and attacks).

Computer architecture defines the basic structure and organization of computer systems. It includes hardware, software, networking, and system design. Understanding these principles helps developers create reliable, efficient, and secure computer systems [6].

Software is a collection of programs, applications, tools, and instructions that control and manage the operation of a computer or other devices. This includes operating systems, applications, development programs, web browsers, games, and much more. Here are some key aspects of software. Operating systems (OS): manage the computer's resources and provide an interface for user and program interaction. Examples: Windows, macOS, Linux, Android, iOS. Application programs and applications: programs for performing specific tasks or operations, such as text editors, spreadsheets, graphic editors, etc. Examples: Microsoft Office, Adobe Photoshop, Google Chrome, VLC Media Player. Development tools and environments: tools for creating, developing, and debugging software. Examples: Visual Studio, IntelliJ IDEA, Eclipse, Git. Internet tools and Web applications: Internet software, web browsers, email clients, social networks, online services, etc. Examples: Google Chrome, Mozilla Firefox, Gmail, Facebook, Twitter. Gaming software: video games and entertainment applications for various platforms. Examples: Fortnite, Minecraft, Call of Duty, Candy Crush. Mobile applications: software for mobile devices such as smartphones and tablets. Examples: WhatsApp, Instagram, Uber, Spotify. System software: software that keeps a computer system running, such as device drivers, antivirus software, security systems, etc. Examples: NVIDIA Graphics Driver, Norton Antivirus, Windows Firewall.

Software plays an important role in modern life, helping to solve various tasks and facilitating the daily activities of users in various spheres of life. Therefore, understanding its role and diversity is key to the effective use of technology in the modern world.

Programming languages. A programming language is a formal language construct designed to write computer programs. It defines the syntax and semantics that programmers use to create programs and interact with a computer. Today, there are many different programming languages, each of which has its own features and purpose. Here are some of the most popular programming languages [3,4].

Python: easy to use and understandable for beginners, has a wide range of applications including web development, data science, artificial intelligence, etc., and is known for its clean code and readability.

Java: a cross-platform programming language, which means that programs can run on different operating systems, is used to create web applications, mobile applications, embedded systems, etc., and is known for its reliability and high performance.

JavaScript: used to develop web applications and dynamic websites, can be used as a client and server-side programming language (using Node.js), provides interactivity and dynamism of web pages.

C++: general-purpose programming language, used to create fast and efficient programs, widely used for the development of system software, embedded systems, games, etc., provides greater control over the

memory and resources of the computer. C#: developed by Microsoft, is used to create programs for the .NET platform, is used to develop web applications, mobile applications and desktop programs for Windows, supports object-oriented programming and has a large ecosystem of libraries and tools.

Ruby: a simple and elegant programming language used for web development, in particular for the Ruby on Rails framework, provides a high level of performance and flexibility.

PHP: a programming language specialized in web application development and web server operations, used to create dynamic websites and work with databases, is widely supported and has a large community of users.

These are just a few of the most popular programming languages, and there are many other languages with different features and applications. The choice of programming language depends on the specific needs of the project, the characteristics of the target platform, etc.

Conclusions

This article provides an overview and analysis of information technologies for data processing in the course of modern computer science, as well as theoretical aspects of such concepts as algorithms, data structures, computer architecture, logic, software, programming languages and concepts of their development.

Thus, the development of information technology has a significant impact on technological progress and social development. It facilitates innovations in various fields, from medicine to finance, which make life easier and improve working conditions, the creation of new business sectors and labor markets that meet the requirements of the digital economy, and changes in the way society communicates, educates and interacts, in particular through the development of social networks and Internet platforms.

Informatics and information technology have become not only the basis of modern programming, but also key determinants of technological progress and societal development. Understanding their role and interconnection helps to unlock the potential of these industries for further development and achievement of new heights in technology and innovation.

References

1. Gurzhiy A.M., Povoroznyuk N.I., Samsonov V.V. Informatics and Information Technologies - Kharkiv, 2017-352 p.
2. Kravchuk OA Review and analysis of the architecture of progressive web applications / Olga Kravchuk // Bulletin of Khmelnytsky National University - Khmelnytsky: KHNU, 2017, №5. - C.132-136
2. Kravchuk OA The use of the Go language in software development / O.M. Yashyna, O.A. Kravchuk // Bulletin of Khmelnytsky National University: KHNU, 2018, №3. - C.155-157
3. Kravchuk OA Analysis of existing software and hardware for the development of a software product for automating the processing of orders of online stores / OA Kravchuk // Bulletin of Khmelnytsky National University: KHNU, 2019, №1. - C.178-181
4. Kravchuk OA Features and advantages of the PHP 7 programming language / OA Kravchuk, OM Sinyuk, AY Kravchuk // Abstracts of the international scientific and technical conference "VIII Ukrainian-Polish Scientific Dialogues", October 16-19, 2019, Khmelnytsky-KNU - P.150-151
5. Kravchuk O.A. The essence and content of information and communication technologies / O.A. Kravchuk, O.M. Sinyuk // Abstracts of the International Scientific Conference "News of Advanced Science - 2020", 15-22.06.2020, Bulgaria, Sofia. - C.63-66
6. Kravchuk O.A. Research of microservice architecture, architectural style rest and their modern implementation in Java / O.M. Yashyna, O.A. Kravchuk // Bulletin of Khmelnytsky National University: KHNU, 2020, №5. - C.106-114
8. Kravchuk OA Modern information and communication technologies in the educational process / OA Kravchuk // Proceedings of the XVIII International Scientific Conference "Modern Advances in Science and Education", Netanya, Israel, 13-20. 09.2023 - P.46-50 <https://iftomm.ho.ua/docs/MASE-2023.pdf>
9. Kravchuk OA Analysis and method of applying best practices for 10. successful risk management in IT projects / Olga Kravchuk, Denys Kravchuk / International scientific and technical journal "Measuring and computing devices in technological processes". - Khmelnytskyi, KhNU, №4 (2023). - PP. 160-164 DOI: <https://doi.org/10.31891/2219-9365-2023-76-21>.