

КРАВЧУК ОЛЬГА

Хмельницький національний університет

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

КРАВЧУК АНДРІЙ

Хмельницький національний університет

<https://orcid.org/0000-0001-8909-432X>e-mail: iiiiiiiilokiiii@gmail.com

ОГЛЯД ТА АНАЛІЗ ІНФОРМАТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ЯК ОСНОВ СУЧАСНОГО ПРОГРАМУВАННЯ

Стаття зосереджується на огляді та аналізі інформатики та інформаційних технологій як основ сучасного програмування, а також в статті наведені теоретичні аспекти 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 не тільки основою сучасного програмування, але й ключовими визначниками технологічного прогресу та розвитку суспільства. 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, програмування, інформаційні технології, застосування.

KRAVCHUK OLGA, KRAVCHUK ANDRII

Khmelnitskyi National University

IMPROVEMENT OF METHODS FOR ESTIMATING THE LABOR INTENSITY AND DEVELOPMENT TIME OF IT PROJECTS

The article focuses on the review and analysis of computer science and information technology as the foundations of modern programming, and the article presents theoretical aspects and concepts of their development that 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 of communication, education and interaction in society, including h The article analyzes software, which plays an important role in modern life, helping to solve various tasks and facilitating daily activities of users in various spheres of life. Accordingly, understanding its role and diversity is key to the effective use of technology in the modern world. In this article, we use the concept 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 (the use of conditional statements and logical operations helps to make the code more efficient and readable); algorithms (developing effective algorithms requires logical thinking and the ability to make the right decisions at every step).

In this article, we have also provided examples of 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. This article describes some of the most popular programming languages. Informatics and information technology have become not only the basis of modern programming, but also key determinants of technological progress and social 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.

Keywords: computer science, programming, information technology, applications

Постановка проблеми

Інформатика та інформаційні технології стали неодмінною частиною нашого сучасного життя. Вони пронизують майже всі сфери людської діяльності, включаючи бізнес, освіту, медицину, та, безумовно, програмування. Інформатика вважається не лише наукою про обробку та передачу інформації, але й майстерністю, яка лежить в основі сучасного програмування. Завдяки інформатиці ми розуміємо, як комп'ютери працюють, як вони обробляють дані та як можна використовувати цю інформацію для створення програм і систем. У цій статті ми розглянемо, як інформатика та інформаційні технології стали невід'ємною основою для сучасного програмування та як ця взаємодія впливає на технологічний прогрес та розвиток суспільства, а також що інформатика є ключем до вивчення програмування та чому знання її основ дуже важливе для майбутніх програмістів.

Метою цієї статті є огляд та аналіз інформатики та інформаційних технологій як основ сучасного програмування.

Виклад основного матеріалу

Інформатика, як наука про обробку інформації, та інформаційні технології, що забезпечують інструменти для цієї обробки, утворюють основу для сучасного програмування [1]. Вони надають не лише необхідні знання та інструменти, але і визначають стратегії та методології розробки програмного забезпечення. Інформатика охоплює багато аспектів, таких як алгоритми, структури даних, теорія обчислень та багато іншого. Однак на перших етапах навчання основи інформатики зазвичай включають в себе такі

концепції, як алгоритми, структура даних, логіка, комп'ютерна архітектура, мови програмування, програмне забезпечення. Розглянемо і охарактеризуємо їх [3,5,7,8,9].

Алгоритми - це послідовність кроків, які виконуються для вирішення певної задачі. Розробка ефективних алгоритмів - ключовий аспект програмування. Знання про алгоритми та структури даних є основою будь-якого програміста. Вони визначають ефективність та продуктивність програм, що розробляються. Алгоритми - це точно визначений набір інструкцій або процедур, які виконуються крок за кроком для вирішення певної задачі або досягнення певної мети. Вони є базовим елементом для розв'язання різноманітних завдань в комп'ютерних науках та програмуванні. Алгоритми можуть бути записані у формі послідовності кроків або відображені у вигляді блок-схем, діаграм або псевдокоду. Основні характеристики алгоритмів включають: чіткість (алгоритм повинен бути зрозумілим та легко читабельним для людей, кожен крок повинен бути чітко описаним та не залишати місця для двозначності або непорозуміння); коректність (алгоритм повинен вирішувати поставлену задачу правильно, що означає, він має повертати правильний результат для будь-яких вхідних даних та у всіх можливих умовах виконання); ефективність (алгоритм повинен бути ефективним з точки зору часу та ресурсів, необхідних для його виконання, що означає, він має працювати швидко та ефективно навіть для великих обсягів даних або складних завдань); універсальність: хороший алгоритм може бути застосований до різних типів вхідних даних та різних ситуацій, він має бути загальним і відповідати потребам різних задач); модульність (алгоритм може бути розділений на окремі частини (модулі), які можуть бути виконані окремо та можуть бути перевикористані у різних контекстах); стабільність (алгоритм повинен давати стабільний результат для одних і тих самих вхідних даних при кожному виконанні, без несправедливих або непередбачуваних змін у результаті); масштабованість (хороший алгоритм повинен бути здатний працювати ефективно навіть для дуже великих обсягів даних чи великої кількості операцій); придатність для використання (алгоритм повинен бути реалізованим у практичному програмному коді та бути доступним для використання в реальних програмах) та можливість виконання за скінченний час. Ці характеристики допомагають визначити якість алгоритму та його придатність для вирішення конкретних завдань. Алгоритми використовуються для різних цілей, включаючи сортування даних, пошук шляху в графі, оптимізацію завдань та багато іншого [2,3,8].

Структури даних-це способи організації та зберігання даних у комп'ютері таким чином, щоб їх можна було ефективно обробляти та отримувати доступ до них. Наприклад, масиви, списки, стеки, черги тощо. Вони є ключовим елементом в програмуванні, оскільки визначають, як дані будуть представлені та оброблені в програмах. Розглянемо деякі основні аспекти структури даних. Організація даних: структури даних визначають, як дані будуть організовані та зберігатися в пам'яті комп'ютера. Це може включати списки, масиви, дерева, графи та інші структури. Ефективність доступу: одні з основних характеристик структур даних - це їхня швидкість та ефективність доступу до даних. Деякі структури даних можуть бути оптимізовані для швидкого доступу до елементів або для швидкого пошуку. Операції над даними: багато структур даних підтримують різні операції, такі як вставка, видалення, пошук та сортування. Важливо, щоб ці операції були ефективними та виконувалися за прийнятним часом. Методи доступу: структури даних можуть мати різні методи доступу до їхніх елементів. Наприклад, деякі структури даних підтримують доступ до елементів за індексом, а інші - за ключем або шляхом. Використання пам'яті: важливо враховувати використання пам'яті при використанні структур даних. Деякі структури можуть бути більш ефективними з точки зору використання пам'яті, ніж інші. Придатність до застосування: вибір певної структури даних залежить від конкретного завдання або сценарію використання. Деякі структури даних можуть бути більш підходящими для певних операцій або типів даних, ніж інші. Узагальнюючи, структури даних надають зручний та ефективний спосіб організації, зберігання та обробки даних у програмах. Вони є ключовим елементом в програмуванні та дозволяють розробникам створювати швидкі та ефективні програми для вирішення різноманітних завдань.

Логіка. Логіка є важливим аспектом в інформатиці, особливо при програмуванні. Інформатика вивчає принципи логіки та умовні конструкції, які дозволяють програмам приймати рішення. Логіка визначає правила та принципи мислення, які використовуються для вирішення проблем та прийняття рішень у програмах. Розглянемо дещо детальніше цей аспект. Принципи логіки - логічні операції:

- логічне і (and): повертає істину лише тоді, коли обидва вирази істинні;
- логічне або (or): повертає істину, якщо хоча б один з виразів істинний;
- логічне не (not): змінює істину на хибність та навпаки.

умовні конструкції:

- умовний оператор (if-else): використовується для виконання коду, якщо певна умова істинна;
- оператор вибору (switch-case): дозволяє виконати різний код в залежності від значення виразу.

цикли та повторення:

- цикл while: виконується, доки умова істинна.
- цикл for: використовується для повторення блоку коду задане число разів.

логічне мислення:

- розробники програм повинні мати здатність мислити логічно та аналізувати проблеми за допомогою формальної логіки.

Важливість логіки в програмуванні: прийняття рішень (логіка дозволяє програмам приймати рішення на основі умов та обставин), уникнення помилок (логічне програмування допомагає уникати помилок та неправильних рішень в програмах); ефективність коду (використання умовних конструкцій та логічних операцій допомагає зробити код більш ефективним та читабельним); алгоритми (розробка ефективних алгоритмів вимагає логічного мислення та уміння приймати правильні рішення на кожному кроці).

Отже, логіка в програмуванні відіграє критичну роль, оскільки вона визначає способи прийняття рішень та умови виконання коду, що впливає на ефективність, надійність та коректність програм.

Комп'ютерна архітектура. Комп'ютерна архітектура - це структура та організація компонентів комп'ютерної системи, яка визначає її функціональні можливості та спосіб взаємодії між ними. Це область інформатики, яка вивчає, як комп'ютери будуються та працюють відносно апаратного та програмного забезпечення. Розглянемо деякі ключові аспекти комп'ютерної архітектури.

По-перше, це апаратна архітектура: процесор (ЦП) - центральний процесор виконує інструкції та керує роботою комп'ютера; пам'ять - оперативна пам'ять (ОЗП) для зберігання даних та програм, та постійна пам'ять (наприклад, жорсткий диск або твердотільний накопичувач) для зберігання інформації навіть після вимкнення комп'ютера; шина даних - забезпечує зв'язок між різними компонентами, передачу даних та інструкцій.

По-друге, це системний софт: операційна система (керує ресурсами комп'ютера та надає середовище для виконання програм), драйвери (програмне забезпечення, яке забезпечує інтерфейс між апаратурою та операційною системою), утиліти (допоміжні програми для діагностики, налагодження та підтримки комп'ютерної системи).

По-третє, це мережева архітектура: мережеве з'єднання (дозволяє комп'ютерам та іншим пристроям обмінюватися даними та ресурсами); протоколи та стандарти (визначають правила обміну даними між пристроями у мережі).

По-четверте, це розробка систем: інтеграція та тестування (процес поєднання різних компонентів та перевірки їх взаємодії); масштабованість (здатність системи працювати ефективно при збільшенні обсягів даних або навантаження).

По-п'яте, це безпека та приватність: шифрування та аутентифікація (заходи безпеки для захисту даних від несанкціонованого доступу); захист від зловмисних програм (антивірусне програмне забезпечення та інші заходи безпеки для запобігання вразливостям та атакам).

Отже, комп'ютерна архітектура визначає основну структуру та організацію комп'ютерних систем. Вона включає апаратне та програмне забезпечення, мережеві зв'язки та розробку систем. Розуміння цих принципів допомагає розробникам створювати надійні, ефективні та безпечні комп'ютерні системи [6].

Програмне забезпечення (ПЗ) - це сукупність програм, додатків, інструментів та інструкцій, які контролюють та керують роботою комп'ютера або інших пристроїв. Це включає в себе операційні системи, додатки, програми для розробки, веб-браузери, ігри та багато іншого програмного забезпечення. Ось деякі ключові аспекти програмного забезпечення. Операційні системи (ОС): керують ресурсами комп'ютера та забезпечують інтерфейс для взаємодії з користувачем та програмами. Приклади: Windows, macOS, Linux, Android, iOS. Додаткові програми та додатки: програми для виконання конкретних завдань або операцій, такі як текстові редактори, електронні таблиці, графічні редактори тощо. Приклади: Microsoft Office, Adobe Photoshop, Google Chrome, VLC Media Player. Розробницькі інструменти та середовища: інструменти для створення, розробки та налагодження програмного забезпечення. Приклади: Visual Studio, IntelliJ IDEA, Eclipse, Git. Інтернет-інструменти та Веб-додатки: програмне забезпечення для роботи з Інтернетом, веб-браузери, поштові клієнти, соціальні мережі, онлайн-сервіси тощо. Приклади: Google Chrome, Mozilla Firefox, Gmail, Facebook, Twitter. Ігрове програмне забезпечення: відеоігри та розважальні додатки для різних платформ. Приклади: Fortnite, Minecraft, Call of Duty, Candy Crush. Мобільні додатки: програмне забезпечення для мобільних пристроїв, таких як смартфони та планшети. Приклади: WhatsApp, Instagram, Uber, Spotify. Системне програмне забезпечення: програмне забезпечення, яке забезпечує роботу комп'ютерної системи, таке як драйвери пристроїв, антивірусне програмне забезпечення, системи безпеки тощо. Приклади: NVIDIA Graphics Driver, Norton Antivirus, Windows Firewall.

Програмне забезпечення відіграє важливу роль у сучасному житті, допомагаючи вирішувати різноманітні завдання та полегшуючи щоденні дії користувачів у різних сферах життя. Відповідно, розуміння його ролі та різноманітності є ключовим для ефективного використання технологій у сучасному світі.

Мови програмування. Мова програмування - це формальний мовний конструкт, який призначений для написання комп'ютерних програм. Вона визначає синтаксис та семантику, які програмісти використовують для створення програм та взаємодії з комп'ютером. На сьогоднішній день існує багато різних мов програмування, кожна з яких має свої особливості та призначення. Наведемо деякі найпопулярніші мови програмування [3,4].

Python: проста у використанні та зрозуміла для початківців, має широкий спектр застосувань, включаючи веб-розробку, науку даних, штучний інтелект тощо, відома своєю чистотою коду та читабельністю.

Java: кросплатформена мова програмування, що означає, що програми можуть запускатися на різних операційних системах, використовується для створення веб-додатків, мобільних додатків, вбудованих систем тощо, відома своєю надійністю та високою продуктивністю.

JavaScript: використовується для розробки веб-додатків та динамічних веб-сайтів, може використовуватися як клієнтська, так і серверна мова програмування (за допомогою Node.js), забезпечує інтерактивність та динамічність веб-сторінок.

C++: мова програмування загального призначення, використовується для створення швидких та ефективних програм, широко використовується для розробки системного програмного забезпечення, вбудованих систем, ігор тощо, надає велику контроль над пам'яттю та ресурсами комп'ютера.

C#: розроблена компанією Microsoft, використовується для створення програм для платформи .NET, використовується для розробки веб-додатків, мобільних додатків та десктопних програм для Windows, підтримує об'єктно-орієнтоване програмування та має велику екосистему бібліотек та інструментів.

Ruby: проста та елегантна мова програмування, використовується для веб-розробки, зокрема для фреймворку Ruby on Rails, забезпечує високий рівень продуктивності та гнучкість.

PHP: мова програмування, спеціалізована на розробці веб-додатків та роботі з веб-серверами, використовується для створення динамічних веб-сайтів та роботи з базами даних, широко підтримується та має велику спільноту користувачів.

Це лише кілька з найпопулярніших мов програмування, існує багато інших мов з різними особливостями та застосуваннями. Вибір мови програмування залежить від конкретних потреб проекту, характеристик цільової платформи, тощо.

Висновки

В даній статті проведено огляд та аналіз інформатики та інформаційних технологій як основ сучасного програмування, а також наведені теоретичні аспекти та концепції розвитку програмування на основі інформатики та інформаційних технологій.

Таким чином, розвиток програмування на основі інформатики та інформаційних технологій має значний вплив на технологічний прогрес та суспільний розвиток. Він сприяє 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 не тільки основою сучасного програмування, але й ключовими визначниками технологічного прогресу та розвитку суспільства. Розуміння їхньої ролі та взаємозв'язку допомагає розкрити потенціал цих галузей для подальшого розвитку та досягнення нових вершин у технологіях та інноваціях.

Література

1. Гуржій А.М., Поворознюк Н.І., Самсонов В.В. Інформатика та інформаційні технології.-Харків, 2017-352 с.
2. Кравчук О.А. Огляд та аналіз архітектури прогресивних веб-додатків / Ольга Кравчук // Вісник Хмельницького національного університету. – Хмельницький : ХНУ, 2017, №5. - С.132-136
3. Кравчук О.А. Використання мови Go у розробці програмного забезпечення / О.М. Яшина, О.А. Кравчук // Вісник Хмельницького національного університету. – Хмельницький : ХНУ, 2018, №3. - С.155-157
4. Кравчук О.А. Особливості та переваги мови програмування PHP 7 / О.А. Кравчук, О.М. Синюк, А.Ю. Кравчук // Тези наукових праць міжнародної науково-технічної конференції «VIII Українсько-Польські Наукові Діалоги», 16-19 жовтня 2019 р., м. Хмельницький-ХНУ - С.150-151
5. Кравчук О.А. Сутність та зміст інформаційно-комунікаційних технологій / О.А. Кравчук, О.М. Синюк // Тези наукових праць Міжнародної наукової конференції «Новости передовой науки – 2020», 15-22.06.2020 р., Болгарія, м. Софія. - С.63-66
6. Кравчук О.А. Дослідження мікросервісної архітектури, архітектурний стиль rest та їх сучасна реалізація на Java / О.М. Яшина, О.А. Кравчук // Вісник Хмельницького національного університету. – Хмельницький : ХНУ, 2020, №5. - С.106-114
7. Кравчук О.А. Сучасні інформаційно-комунікаційні технології в освітньому процесі / О.А. Кравчук // Збірник праць XVIII Міжнародної наукової конференції «Сучасні досягнення в науці та освіті», м. Нетанія, Ізраїль, 13-20. 09.2023 р.- С.46-50 <https://iftomm.ho.ua/docs/MASE-2023.pdf>
8. Кравчук О.А. Модель впровадження CI/CD для оптимізації управління IT-проектами / Ольга Кравчук / Міжнародний науково - технічний журнал «Measuring and computing devices in technological processes». - Хмельницький, ХНУ, №3 (2023). - С. 73–82 DOI: <https://doi.org/10.31891/2219-9365-2023-75-8>
9. Кравчук О.А. Аналіз та метод застосування найкращих практик для успішного управління ризиками в IT-проектах / Ольга Кравчук, Денис Кравчук / Міжнародний науково - технічний журнал «Measuring and computing devices in technological processes». - Хмельницький, ХНУ, №4 (2023). - С. 160–164 DOI:<https://doi.org/10.31891/2219-9365-2023-76-21>.

References

1. Gurzhiy A.M., Povoroznyuk N.I., Samsonov V.V. Informatics and Information Technologies - Kharkiv, 2017-352 p.
2. Kravchuk O.A. Review and analysis of the architecture of progressive web applications / Olga Kravchuk // Bulletin of Khmelnytsky National University - Khmelnytsky: KHNU, 2017, №5. - С.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>.