

ПРО ДЕЯКІ ЗАГАЛЬНІ ПИТАННЯ ІНФОРМАТИКИ

Стаття присвячена уточненню предмета інформатики та одного з центральних її понять - програмування.

1. Предмет інформатики. Самовизначення інформатики як наукової дисципліни відбувається досить суперечливо і балансує між двома полюсами - "інженерним" та "математичним". Кожен з них має своє об'єктивне підґрунття, проте ані порізно, ані разом вони не вирішують проблему ідентифікації інформатики в цілому. Інженерний підхід розглядає інформатику як науку про комп'ютерні системи (див., наприклад, [1]). Він домінував у період становлення інформатики, коли проектувалися та створювалися перші комп'ютерні системи. Сьогодні такий підхід поширюється і на комплекс технологічних проблем, пов'язаних з проектуванням, розробкою та експлуатацією комп'ютерних систем. Математичний підхід почав превалювати в останні роки у зв'язку з тотальним проникненням інформаційних технологій в усі сфери життя суспільства, що породило проблему різкого підвищення продуктивності праці в галузі продукування інформаційних технологій, підвищення надійності та суттєвого зниження їх вартості. Фактично йдеться вже про запровадження індустриальних методів не тільки в масовому виробництві комп'ютерів, а й і у виробництві програмного забезпечення для них. Подібне виробництво з його всебічною автоматизацією вимагає ґрунтовного наукового забезпечення, що і започаткувало процес виділення інформатики в окрему наукову галузь зі своїм предметом та методами вивчення. То що ж є предметом інформатики? Як ми вже зауважували, єдиної думки щодо цього серед спеціалістів поки що немає. Стосовно самого терміна "інформатика", то на теренах колишнього СРСР він попервах позначав спеціальну "наукову дисципліну, що вивчає структуру і загальні властивості наукової інформації,

а також закономірності всіх процесів наукової комунікації - від неформальних процесів обміну такою інформацією до формальних процесів обміну за допомогою наукової літератури" [2], а сьогоднішня її проблематика належала кібернетикі як науці про керування в складних системах. А.П.Єршов вбачає в терміні інформатика "назву фундаментальної природничої науки, яка вивчає процеси передачі та обробки інформації" [3]. Б.Майер та К.Бодуен наводять два варіанти визначення інформатики - як "Computer Science" (комп'ютерна наука) та як "теорії обробки інформації" [4]. Ф.Л.Бауер та Г.Гооз посилаються на визначення Французької академії: "Інформатика: Наука про здійснювану головним чином за допомогою автоматичних засобів цілеспрямовану обробку інформації, яка розглядається як донесення знань та повідомлень в технічних, економічних та соціальних галузях" [5]. Ряд вчених [6] основні напрями інформатики пов'язують з "розробкою спеціальних комп'ютерних методів розв'язання складних дослідницьких і практичних задач". В деяких останніх працях [7] предмет інформатики визначається як наукова дисципліна, що вивчає "інформаційні технології і соціально-комунікативні процеси, тобто складні процеси збирання, збереження і перетворення інформації, проблеми їх розвитку та "умонтовування" в соціальне середовище." А в деяких підручниках з інформатики взагалі уникають (!) означення її предмета (як, наприклад, в [8] та ін.) Спробуємо запропонувати для обговорення інший підхід для відповіді на поставлене питання. Будемо виходити з того, що будь-яка точна наука має справу з певними моделями природних чи суспільних явищ. Так, математика вивчає математичні моделі систем об'єк-

тів, фізика – фізичні моделі природних явищ, прикладна соціологія – соціологічні моделі і т.д. Звідси природним виглядає таке означення інформатики.

Def. 1. Інформатика – це наука, що вивчає моделі систем обробки інформації на всіх етапах їх життєвого циклу, починаючи з первинної специфікації та проектування і закінчуючи супроводженням та експлуатацією.

Такі моделі логічно було б назвати інформаційними. Найважливішими видовими властивостями інформаційних моделей є конструктивність їх об'єктів та алгоритмічна обчислювальність функцій. Таке обмеження ми вважаємо принциповим, оскільки воно дозволяє чітко зафіксувати межі інформатики й обернути її предмет від надмірного узагальнення й ототожнення з предметами інших наук. Насамперед з кібернетикою як наукою про загальні закони перетворення інформації в складних системах керування [9-11], та інформологією – “узагальнюючою наукою про інформацію в цілому, про всі її прояви та властивості, про всі види інформаційних процесів” [12,13].

Зупинимось більш детально на понятті інформаційної моделі. У самому загальному випадку під інформаційною системою будемо розуміти довільну сукупність інформаційних об'єктів та функціональних відношень між ними. Подібні функції можуть бути і багатозначними. Самі інформаційні об'єкти або мають структуру упорядкованої пари $\langle v, \varpi \rangle$, де v – ім'я об'єкта з певної сукупності імен V , а ϖ – його значення з певної сукупності значень Ω (атомні інформаційні об'єкти), або будуються з таких об'єктів за допомогою певних операцій-конструкторів (складені інформаційні об'єкти). Типовими складеними інформаційними об'єктами є кортежі фіксованої довжини (однорідні та неоднорідні), скінченні послідовності, різні теоретико-множинні та функціональні структури. Коли говорять, що інформаційна система моделює певну систему об'єктів (вхідну систему), то під цим розуміють, що її об'єкти та функції копіюють, імітують об'єкти та функції вхідної системи. Математично це означає, що інформаційна система є гомоморфним образом вхідної системи. Нехай $\mathbf{S} = \langle S, F \rangle$ – довільна система і $\mathbf{I} = \langle I, G \rangle$ – певна інформаційна система з множинами об'єктів S, I та сукупностями $F = \{f_1, f_2, \dots, f_n\}$ та $G = \{g_1, g_2, \dots, g_m\}$ часткових багатозначних функцій вигляду $f_i : S \rightarrow 2^S$ та $g_j : I \rightarrow 2^I$ відпо-

відно. Будь-яке відношення можна трактувати як подібну функцію і використовувати при цьому відповідну функціональну символіку. Так, якщо $y \in f(x)$ для певних x та y , то будемо писати $y = f(x)$ за умови, що з контексту зрозуміло, що мова йде про одне із значень функції f на x , а не про всю їх сукупність. Ін'єктивні функції $\varphi : S \rightarrow I$ та $\phi : I \rightarrow S$ будемо називати відповідно функціями кодування та декодування множини S .

Def. 2. Говорять, що система \mathbf{I} є наближеною моделлю системи \mathbf{S} відносно функцій кодування та декодування φ і ϕ , якщо для всіх $1 \leq i \leq n$ і для будь-якого s з області визначення функції $f_i : f_i(s) \subseteq \phi(g_i(\varphi(s)))$.

У випадку взаємодозначних функцій кодування та декодування про інформаційні моделі говорять як про точні. А коли φ та ϕ є бієктивними, а система \mathbf{S} – у свою чергу моделлю інформаційної системи \mathbf{I} відносно функцій кодування та декодування ϕ та φ , то говорять про еквівалентні або ізоморфні моделі. Наприклад, десяткова $\mathbf{Z}_{10} = \langle Z_{10}; +, -, \times, /, =, < \rangle$ та двійкова $\mathbf{Z}_2 = \langle Z_2; +, -, \times, /, =, < \rangle$ арифметики цілих чисел є еквівалентними відносно функцій кодування та декодування, що переводять десяткові числа в рівні двійкові і навпаки.

Конструктивність об'єктів інформаційної моделі означає, що вони можуть бути побудовані зі своїх складових за допомогою спеціальних операцій-конструкторів. Сукупність атомних елементів та конструкторів, що дозволяють побудувати всі об'єкти даної інформаційної моделі будемо називати системою її подання, а елементи останньої – конструктивними. Кожен з них є або атомним, або складеним, побудованим з атомних елементів за допомогою певної скінченної кількості конструкторів. Конструктивними є будь-які індуктивно визначені об'єкти (натуральні та раціональні числа, слова в певному алфавіті, скінченні множини та графи, дерева, рекурсивні функції і т.д.). Важливим наслідком конструктивності об'єктів є той факт, що всі вони можуть бути подані у вигляді певних слів-термів, що відповідають процесам їх побудови. Конструктивність об'єктів залежить від вибору системи їх подання. Так, дійсна функція $f(x, y) = \sin(x) + \cos(y/2)$ отримана в результаті підстановки в операцію “+” функцій $\sin(x)$ та $\cos(y/2)$ на місце відповідних аргументів і як об'єкт є конструктивною відносно своїх складових – операцій $+$, $/$, функцій \sin , \cos та конст-

руктора-операції підстановки. Але вона є неконструктивним об'єктом, наприклад, у сенсі Тези Черча.

Уточнимо тепер, що таке процедурний та алгоритмічний засоби подання функцій в певній абстрактній області, оскільки вони є центральними в інформатиці. Змістовно, при такому підході (іноді його називають ще інтенціональним) функція трактується як певне правило, що дозволяє за допомогою певних операцій за скінченну кількість кроків знайти її значення для будь-яких заданих аргументів. Звісно, якщо воно взагалі існує. Нехай S довільна множина, елементи якої будемо називати станами. Виділимо підмножини $S_0 \subseteq S$, $S_{fin} \subseteq S$ відповідно початкових та заключних станів. Нехай $\Delta = \{f_1, f_2, \dots, f_n\}$ довільна множина певних багатозначних функцій на станах $f_i : S \rightarrow 2^S$. Будемо їх називати базовими або елементарними. Функцію вигляду $\delta : N \times (S - S_{fin}) \rightarrow 2^\Delta$ назвемо функцією керування. Процедурою над базисом Δ (Δ -процедурою) будемо називати п'ятірку $P = \langle S, S_0, S_{fin}, \Delta, \delta \rangle$. Послідовність $s_0, s_1, \dots, s_n, \dots$, де $s_i = f_{k_i}(s_{i-1})$, а $f_{k_i} = \delta(i, s_{i-1})$ для всіх $i \geq 1$, називається обчисленням за процедурою P на початковому стані s_0 . Стан $s_n \in S_{fin}$ для певного $n \geq 0$ називається заключним станом (результатом) обчислення s_0, \dots, s_n . Результат обчислення будемо позначати $P'(s_0)$ і говорити, що процедура P повертає його на початковому стані s_0 , а про саме обчислення говорити як про результативне. Всі нескінченні обчислення назвемо безрезультатними. Результативні та безрезультатні обчислення називаються повними, решта – проміжними. Кожна процедура P визначає певну багатозначну функцію на станах $P' : S_0 \rightarrow 2^{S_{fin}}$, а саме таку, що $P'(s_0) = s$, де s – результат певного результативного обчислення на початковому стані s_0 . Області визначення та допустимих значень функції P' позначатимемо відповідно $In(P')$ та $Out(P')$. Процедури з однозначними функціями керування називаються детермінованими.

Багато прикладів процедур можна знайти в математиці, зокрема в алгебрі та геометрії. Так, задачі з планиметрії на побудову за допомогою циркуля і лінійки є фактично задачами на побудову певних процедур. Станами тут виступають сукупності фігур на площині, побудовані за допомогою циркуля та лінійки, операції виділення довільної точки площини та точок перетину фігур, іменування точок і фігур, а також умовних варіантів подібних операцій. Звернемось до

арифметики і розглянемо детерміновану процедуру GCD для обчислення найбільшого спільного дільника $\gcd(a, b)$ двох натуральних чисел. Позначимо ff та tt булеві значення відповідно "лже" та "істина". Визначимо операцію розгалуження $(\rightarrow) : \{ff, tt\} \times A \times A \rightarrow A$, як таку, що для будь-яких $a, b \in A : (tt \rightarrow a, b) = a$ і $(ff \rightarrow a, b) = b$. Тоді $GCD = \langle S, S_0, S_{fin}, \Delta, \delta \rangle$, де $S = N \times N$, N – сукупність всіх натуральних чисел, $S_0 = S$, $S_{fin} = \{ \langle d, d \rangle : d \in N \}$, $\Delta = \{f_1, f_2\}$ і для $\forall k \geq 0$, $\forall s \in S - S_{fin}$ $\delta(k, s) = (p(s) \rightarrow f_1, f_2)$. Операції f_1, f_2 та предикат p для будь-яких $a, b \in N$ визначаються як: $f_1(\langle a, b \rangle) = \langle a - b, b \rangle$, $f_2(\langle a, b \rangle) = \langle a, a - b \rangle$, $p(\langle a, b \rangle) = \langle a > b \rangle$. Не складно перевірити, що будь-яке повне обчислення процедури GCD на початковому стані $s_0 = \langle a, b \rangle$ є результативним і $In(GCD') = N \times N$. Безпосередньо із співвідношень: 1) $\gcd(d, d) = d$, 2) $(a > b \Rightarrow \gcd(a, b) = \gcd(a - b, b))$, 3) $(b > a \Rightarrow \gcd(a, b) = \gcd(a, b - a))$ випливає, що, якщо стан $\langle d, d \rangle$ є заключним в обчисленні процедури GCD на початковому стані $s_0 = \langle a, b \rangle$, то $d = \gcd(a, b)$, оскільки на кожному з проміжних станів s_i найбільший спільний дільник компонент залишається незмінним. Отже, $Out(GCD) = S_{fin}$. Насправді функція GCD не співпадає прямо з функцією \gcd , а тільки точно моделює її відносно тотожної функції кодування φ та функції декодування $\phi : N \times N \rightarrow N$ такої, що $\phi(a, a) = a$ для всіх $a \in N$.

Центральним елементом процедур є їх функція керування. Наприклад, щоб реально знайти за скінченну кількість кроків результат обчислення процедури на початковому стані, необхідно забезпечити можливість віднайдення за скінченну кількість результатів функції керування на кожному з проміжних етапів обчислення. З цієї точки зору являють інтерес конструктивні функції керування. Адже особливістю конструктивних функцій (як спеціальних алгебраїчних об'єктів) є можливість обчислення за скінченну кількість кроків їх значень для заданих аргументів з області визначення за умови, що відомі правила таких обчислень для найпростіших складових функцій та конструкторів. Сама процедура обчислення подібна процедурі обчислення значень арифметичних чи алгебраїчних виразів. Звідси важливим є наступне визначення.

Def. 3. Говорять, що Δ -процедура є Δ -алгоритмом відносно певної системи подання, якщо її стани та функція керування є конструктивними в цій системі.

Функція $P' : In(P') \rightarrow Out(P')$, що обчислюється Δ -алгоритмом P , називається Δ -алгоритмічно обчислювальною або Δ -обчислювальною. У разі, коли Δ та система подання зафіксовані, будемо говорити про такі функції як про алгоритмічно обчислювальні функції, або просто як про обчислювальні функції. Повернувшись до процедури GCD , бачимо, що вона є Δ -алгоритмом над базисом $\Delta = \{f_1, f_2\}$ (відповідно функція GCD є Δ -алгоритмічно обчислювальною) відносно складових – сукупності натуральних чисел N , базових функцій f_1 і f_2 , предикату p та операції розгалуження і конструкторів – теоретико-множинних операцій та операції підстановки. Як і процедури, Δ -алгоритми теж можуть бути детермінованими і недетермінованими.

Наведемо основні властивості Δ -алгоритмів, що впливають безпосередньо з їх означення.

Масовість. Δ -алгоритм може бути застосовано до початкових станів з певної їх сукупності.

Дискретність. Обчислення за Δ -алгоритмом складається з дискретної послідовності станів.

Елементарність. На кожному кроці обчислення за Δ -алгоритмом виконується певна елементарна операція з фіксованої сукупності таких операцій Δ .

Визначеність. Порядок застосування операцій в обчисленні за Δ -алгоритмом не є довільним, а обирається відповідно до функції керування.

Направленість. В кожному Δ -алгоритмі є механізм завершення обчислень.

Фінітність. Означає скінченність подання станів та функції керування Δ -алгоритмів як конструктивних об'єктів.

Релятивність. Означає принципову залежність Δ -алгоритму від системи подання його станів та функції керування.

Як бачимо, основні властивості Δ -алгоритмів ті ж, що і у випадку класичних числових чи словарних алгоритмів. Тому поняття Δ -алгоритму може розглядатись як одне з можливих уточнень загального поняття алгоритму в довільній області. Зауважимо, що це уточнення є прямим, а не формулюється в термінах числових або словарних їх моделей, як це трапляється в теорії алгоритмів. Фіксуючи ті чи інші сукупності станів та базових функцій, а також системи подання, можна отримати широкий спектр конкретних алгоритмічних систем. Класи алгоритмів називаються еквівалентними, якщо вони обчислюють один і той же клас функцій. Нескладно показа-

ти, що таким класичним алгоритмічним системам як машини Тюрінга, алгоритми Маркова, операторні алгоритми, формальні граматики та ін. відповідають певні еквівалентні класи Δ -алгоритмів. Цікаво, що для цього досить обмежитися відповідними класами так званих q -періодичних Δ -алгоритмів з умовними базовими функціями, тобто функціями вигляду $(p \rightarrow f, g)$. Функція керування δ називається періодичною з періодом $q \in N$, якщо $\forall k \geq 0, \forall r(0 \leq r < q), \forall s, s' \in S - S_{fin}$ таких, що пари (r, s) та $(kq + r, s')$ належать області визначення $\delta : \delta(kq + r, s') = \delta(r, s)$. Значення такої функції керування фактично не залежить від другого аргументу. Тобто її достатньо визначити, коли перший аргумент пробігає інтервал $[0, q-1]$. Δ -алгоритм $A = \langle S, S_0, S_{fin}, \Delta, \delta \rangle$ називається q -періодичним Δ -алгоритмом, якщо його функція керування є періодичною з періодом q . Функції, що обчислюються за допомогою q -періодичних Δ -алгоритмів, будемо називати періодично-обчислювальними. Наприклад, Δ -алгоритм GCD є 1-періодичним, а функція GCD – періодично-обчислювальною над базисом $\Delta = GCD$. Таким чином, обчислювальність функцій за Тюрінгом, Марковим і т.ін. є синонімом їх періодичної обчислювальності.

Для кожного класу конструктивних функцій важливою є проблема його алгоритмізації (синтезу), яка полягає в побудові того чи іншого класу Δ -алгоритмів, що обчислюють (моделюють) всі його функції. Дуальною є проблема конструктивізації даного класу Δ -обчислювальних функцій, яка полягає у пошуку певної системи його подання (проблема аналізу). Наприклад, алгебра Черча частково-рекурсивних функцій може розглядатись як один із роз'язків проблеми аналізу для числових функцій, обчислюваних машинами Тюрінга чи операторними алгоритмами. Самі ж машини Тюрінга та операторні алгоритми, в свою чергу, алгоритмізують відповідно класи частково-рекурсивних словарних та числових функцій. У загальному випадку, проблеми аналізу та синтезу вимагають уточнення систем подання відповідних класів конструктивних функцій та Δ -алгоритмів і потребують окремого розгляду.

2. Програми та програмування. Для опису (специфікації) інформаційних моделей використовують спеціальні знакові системи – мови програмування. Мова програмування – це спеціальна знакова система, призначена для опису певних фіксованих класів інформаційних об'єктів та алгоритмів.

Def. 4. Даними мов програмування називаються конструкції, які подають в них інформаційні об'єкти, а програмами - конструкції, що подають алгоритми. Функції, що обчислюються програмами, називаються програмними.

Визначимо тепер центральне поняття інформатики - програмування. Зазвичай при цьому апелюють до тих чи інших його окремих аспектів. Найчастіше програмування трактують як процес написання (побудови, конструювання і т.п.) програм та алгоритмів [6,14], або як процес побудови програм для розв'язання певної задачі за допомогою комп'ютера [2,15,16]. Останній варіант хоч і є більш вдалим, але теж не може вважатися цілком задовільним, оскільки потребує в свою чергу з'ясування, що таке задача і т.д. Виходячи з нашого визначення інформатики, більш змістовним виглядає наступне формулювання.

Def. 5. Програмуванням називається процес побудови інформаційних моделей систем.

Отже, програмування - це не просто процес написання якоїсь окремої програми або тієї чи іншої сукупності програм засобами певної мови програмування, а цілеспрямований процес побудови інформаційних систем, що моделюють об'єкти та співвідношення вхідної системи. Зауважимо, що в епіцентрі уваги інформатики перебувають проблеми побудови інформаційних моделей у вигляді так званих обчислювальних систем, тобто симбіозу автоматів з програмним керуванням (наприклад, комп'ютерів) та програмного забезпечення. Тому невід'ємною складовою процесу програмування таких систем є специфікація властивостей апаратури та системних програмних засобів. Розділ інформатики, що вивчає програми та програмування, отримав назву теорії програмування або програмології [17].

3. Про деякі методологічні принципи. Інформаційні моделі систем можуть розглядатись і вивчатись у різних аспектах. Сформулюємо кілька, на наш погляд, важливих принципів, які могли б бути корисними в процесі вивчення та програмуванні таких моделей.

Принцип підпорядкування. Як об'єкти спеціальних знакових систем, всі компоненти інформаційних моделей мають три основні аспекти: синтаксичний, семантичний та прагматичний. Даний принцип фіксує пріоритет прагматичних їх властивостей над семантичними і останні над синтаксичними.

Принцип відокремлення. Цей принцип акцентує увагу на відносній самостійності основних аспектів інформаційних моделей і на їх попередньому відокремленому вивченні з застосуванням своїх спеціальних засобів та методів.

Принцип абстракції. Даний принцип вказує на необхідність використання різних рівнів абстракції в процесі дослідження та програмуванні інформаційних моделей.

Принцип типізації. Згідно з цим принципом кожний об'єкт інформаційної системи належить певному типу і тільки одному. Тип визначає загальну синтаксичну і семантичну структуру даних, а також в певних межах і їх прагматику. Типи мають ієрархічну залежність і утворюють так звану башту типів мов програмування. Засоби утворення цієї башти разом з базовими типами є фундаментальною характеристикою мов програмування.

Принцип аксіоматизації. Принцип аксіоматизації вказує на важливість фіксації певними логічними засобами властивостей систем, щоб зробити процес їх програмування цілеспрямованим та більш контрольованим. З цією метою формують сукупність апріорних властивостей систем (аксіом) та правил, що дозволяють з аксіом отримувати інші необхідні властивості. Даний принцип вимагає ретельного аналізу та аксіоматизації семантики мов програмування й можливостей апаратури.

Принцип функціональності. Принцип функціональності вказує на важливість функціональних структур інформаційних систем в процесі їх вивчення та програмуванні. Такі структури можуть задаватись будь-якими прийнятними засобами, у тому числі і непроедурними. Наприклад, табличними, аналітичними чи алгебраїчними за допомогою рівнянь.

Принцип конструктивізації. Як уже відмічалося, для кожного класу обчислювальних функцій, у тому числі і програмних, важливим є питання його конструктивізації (проблема аналізу), яке полягає у пошуку певної системи його подання за допомогою елементарних базових функцій та відповідних конструкторів. Для програмних функцій мов програмування такі конструктори називаються композиціями. Систему базових програмних функцій та композицій називають імперативною логікою програм [17]. Вона називається повною для даної мови програмування, якщо дозволяє адекватно побудувати

функціональну структуру будь-якої її програми. Побудова та вивчення універсальних повних імперативних логік програм є однією з важливих передумов для порівняльного аналізу та науко-

вої класифікації мов програмування, їх аксіоматизації, для верифікації інформаційних моделей тощо. Більш детальне обговорення цих та інших загальних принципів можна знайти в [18].

1. *Машбиц Е.И. и др.* Основы компьютерной грамотности: Учебное пособие / Под общей редакцией А.А.Стогния и др. - К.: Выща шк., 1988. - 215 с.

2. *Энциклопедія кібернетики.* - К.: УРЕ, 1973. - Т. 1-2.

3. *Ершов А.П.* О предмете информатики // Вестн. АН СССР. - 1984.- № 2.

4. *Мейер Б., Бодуэн К.* Методы программирования. - М.: Мир, 1982.- Т.1-2.

5. *Бауэр Ф.Л., Гооз Г.* Информатика. - М.: Мир, 1990.-Т. 1-2.

6. *Абрамов С.А., Зима Е.В.* Начала информатики. - М.: Наука, 1989. - 256 с.

7. *Хоменко Л.Г.* История отечественной кибернетики и информатики.- К.: Ин-т кибернетики им.В.М.Глушкова Н АН Украины, 1998. - 455 с.

8. *Есаян Р. и др.* Информатика. Учебное пособие для педагогических институтов. - М.: Просвещение, 1991.— 288 с.

9. *Глушков В.М.* Кибернетика. Вопросы теории и практики. - М.: Наука, 1986. - 478 с.

10. *Мороз А.Я.* Кибернетика в системе современного научного знания.-К. .Наукова думка, 1988.- 232с.

11. *Сергієнко І.В.* Становлення і розвиток досліджень з інформатики. - К.: Наукова думка, 1998. - 204 с

12. *Готт В.С., Семенов Э.П., Урсул А.Д.* Методологические проблемы информатики. - Закономерности развития современной математики. - М.: Наука, 1987.— 334 с.

13. *Сифоров В.И.* Информатика и ее взаимодействие с философией и другими науками // Филос. науки. - 1984. - № 2.

14. *Вирт Н.* Систематическое программирование. Введение .- М.: Мир, 1977. - 181 с.

15. *Любимский Э.З. и др.* Программирование,- М.: Наука, - 1980. - 608 с.

16. *Ставровський А.Б., Коваль Ю.В.* Элементы программирования: Учебный посібник,- К.: Фонд "Молоді надії України", 1998. - 210 с

17. *Басараб И.А., Никитченко У.С., Редько В.Н.* Композиционные базы данных. - К.: Либідь,1992. - 192 с.

18. *Редько В.Н.* Экспликативное программирование: ретроспективы и перспективы. // Перша міжнародна науково-практична конференція з програмування УкрПРОГ'98, - К.: Ін-т кібернетики ім.В.М.Глушкова НАНУ, 1998. - С 22-41.

V. V. Zubenko

A NOTE ON THE FOUNDATION OF COMPUTER SCIENCE

The subject of Computer Science and the definition of programming as scientific discipline are discussed.