**Тема 1. Розроблення алгоритмів та блок-схем оброблення економічної інформації**

Прийняття рішень на будь-якому рівні управління об’єктами чи процесами у всіх галузях діяльності людини завжди складається з таких етапів:

— формування цілей;

— визначення основних властивостей очікуваних результатів;

— опис вхідної інформації;

— побудова алгоритму перетворення вхідної інформації у вихідну із заданими властивостями.

І якщо перші три етапи виконує фахівець відповідної галузі, четвертий етап виконує спеціаліст з обробки інформації.

Від того, які характеристики має інформація, як вона організована та зберігається, чи змінюється вона у часі, чи може змінюватись її структура і т.ін., залежить і процес її обробки і алгоритм цього процесу.

На сучасному рівні розвитку організації процесів управління практично відсутні окремі поодинокі задачі обробки інформації. Найчастіше це комплекси задач, пов’язаних між собою спільною інформаційною базою.

Невпинна автоматизація технологічного процесу розробки комплексів прикладних програм підтримки функціонування інформаційних систем, об’єктів управління іде шляхом перерозподілу робіт і функцій між фахівцями відповідної галузі, постановниками задач, проектувальниками систем, програмістами, користувачами та супровідниками програм.

Рівень інформатизації цих робіт і функцій (визначення й формального опису їх складових частин і взаємозв’язку між ними) зумовлює рівень автоматизації. І якщо раніше процес опису алгоритму на мові програмування та налагодження програми вважався найбільш творчим процесом і левова доля витрат часу припадала саме на нього, то тепер цей технологічний етап перетворився на суто технічний, майже рутинний.

Найбільшу увагу приділяють структурному опису даних, їх взаємозв’язку, напрямкам інформаційних потоків, алгоритмам зв’язку функцій обробки.

Соціально-економічна інформація має свої особливості щодо складу та структури, а також потребує певних припустимих операцій обробки. Як правило, ця інформація є деякою сукупністю (записом) елементів (різних типів), кожний з яких містить певну характеристику деякого об’єкта, звідки випливає, що припустимі операції визначатимуться для кожного елемента окремо. Ця інформація має такі особливості:

— більшість операцій обробки логічні;

— простота арифметичних операцій;

— великі обсяги та складна ієрархічна структура інформації;

— циклічність і періодичність обробки;

— обробка порціями (записами).



7

Рисунок 1 - Типи процедур обробки соціально-економічної інформації

**Створення та контроль наборів даних**

Процедура синтаксичного контролю та запису інформації на носії складається з послідовності дій, таких як введення із клавіатури, візуальний та програмний контроль введеної інформації з формуванням відповідних повідомлень щодо результатів контролю. У разі вірної вхідної інформації запис виводиться на носії.

***Види контрольних дій***

— Перевірка відповідності типу, тобто перевірка використання тих символів алфавіту, які припустимі для представлення інформації кожного з типів даних.

— Перевірка відповідності форматів введення даних їхньому синтаксису.

— Перевірка приналежності даних до певних сукупностей, до певних відрізків значень, перевірка відповідності певним логічним умовам і т. ін.

***Приклад 1.*** Підготувати набір даних, створити файл KADR — «Відомості про співробітників» на носії, використовуючи програму синтаксичного контролю. Набір має таку структуру:

|  |  |  |
| --- | --- | --- |
| Елемент | Ідентифікатор | Значність |
| Код підприємства | KPRED | 99999 |
| Код цеху | KZEX | 99 |
| Табельний номер працівників | TABNOM | 9999 |
| Прізвище працівника | FIO | (30)`A` |
| Стать | POL | `A` |
| Рік народження | ROGD | 9999 |
| Стаж роботи загальний | STZAG | 99 |
| Стаж роботи неперервний | STNEP | 99 |
| Код професій | KPROF | 99 |
| Оклад | OKL | 9999.99 |

Елементи даних, що належать запису.

Кодом підприємства є ціле число, як табельний номер та код професії. Отже, їх можна контролювати за приналежністю множині цілих чисел.

Процедура синтаксичного контролю та запису інформації на носії складається з послідовності дій, таких як введення із клавіатури, візуальний та програмний контроль введеної інформації з формуванням відповідних повідомлень щодо результатів контролю. У разі вірної вхідної інформації запис виводиться на носії.

Якими можуть бути контрольні дії?

— Перевірка відповідності типу, тобто перевірка використання тих символів алфавіту, які припустимі для представлення інформації кожного з типів даних.

— Перевірка відповідності форматів введення даних їхньому синтаксису.

— Перевірка приналежності даних до певних сукупностей, до певних відрізків значень, перевірка відповідності певним логічним умовам тощо.

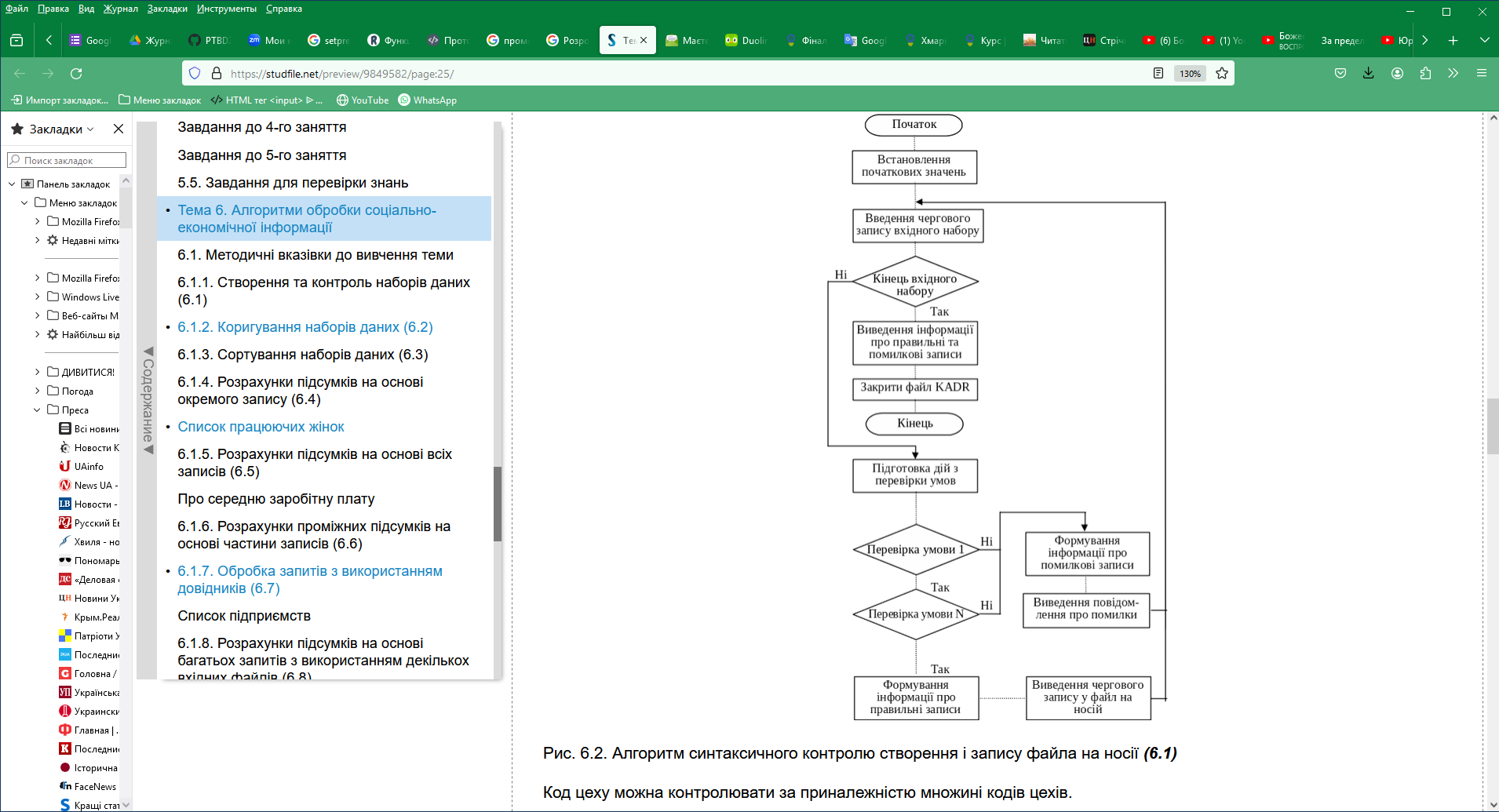


Рисунок 2 - Алгоритм синтаксичного контролю створення і запису файлу на носії

Код цеху можна контролювати за приналежністю множині кодів цехів.

Прізвище контролюється використанням множини літерних символів та символу дефіс.

Елемент «стать» перевіряється приналежністю множині {Ч,Ж}.

Рік народження повинен знаходитись у деяких межах, наприклад 1900 < ROGD < RIK — 14, де RIK — поточний рік.

Стаж роботи загальний не повинен перевищувати вік людини, зменшений на 14 років.

Стаж роботи неперервний не повинен перевищувати стаж загальний.

Оклад має відповідати формату чисел із фіксованою точкою.

***Коригування наборів даних***

Коригування вже створеного файлу може вестися у трьох напрямках: видалення окремих записів, зміна значень деяких полів у всіх або деяких записах файлу, дозапис файлу. В усіх випадках необхідно ввести коригуючу інформацію для пошуку відповідних записів та нові значення полів, якщо вони є. Ця інформація повинна бути заздалегідь підготовленою та вводитись з клавіатури, причому, якщо поле, що є ознакою для пошуку запису, коригується, то можлива подальша обробка файлу, наприклад упорядкування.

Нехай до коригуючої інформації належать: поле ознаки коригування р:

1 — додавання нового запису;

2 — вилучення запису;

3 — коригування полів.

***Сортування наборів даних***

Що стосується сортування та впорядкування записів файлу за значеннями певних окремих полів або груп полів, то тут, як правило, використовуються стандартні процедури, що реалізують різні методи сортування файлів.

***Обробка інформації***

Взагалі кажучи, термін «обробка» передбачає зміну того, на що спрямована дія — оброблення. Але тільки, якщо це не стосується інформації. Тут під «обробкою» розуміють отримання нової інформації з **використанням** тієї, що вже існує (зібрана і зберігається), без зменшення і пошкодження її.

Задачі обробки інформації формулюються у вигляді запиту на отримання інформації, з якої утворюється вихідний документ певної форми.

Перш ніж розробляти алгоритм обробки запиту, необхідно проаналізувати, які дані містить вихідний документ, які з них можна передати з вхідного файлу (файлів), а які треба розрахувати в процесі перегляду записів вхідного файлу.

Крім того, треба визначити, чи достатньо даних вхідних файлів для формування вихідного документа за запитом, тобто чи є коректною постановка задачі обробки запиту.

Вміст та форма представлення результату обробки запиту визначаються у самому запиті у вигляді документа або сформульовані словесно, і тоді треба вміти самостійно розробляти форму вихідного документа чи структуру запису вихідного файлу.

Після такого аналізу можна складати алгоритм обробки запиту з використанням усіх необхідних файлів.

***Обробка запитів з використанням довідників***

Вже давно стало цілком зрозумілим, що існує довідкова інформація, яку використовують користувачі при розв’язанні багатьох задач обробки або комплексів задач. Довідники містять повні атрибути підприємств, характеристики матеріалів, відомості про мешканців міст, про автомобілі, тобто про все, що може знадобитись людині в її професійний діяльності, соціальному та приватному житті. Довідники можуть мати локальний характер використання або використовуватись у державному масштабі, але принцип організації вони мають однаковий. Записи в них завжди впорядковані за певною ознакою, яка використовується для пошуку потрібної інформації. Але треба зазначити, що є можливість впорядкувати той же файл-довідник за іншою ознакою, тобто отримати так званий індексований файл. Залежно від характеру задачі до довідника можна звертатися один чи декілька разів, тому потрібно визначити, чи можна при повторному звертанні продовжувати читати довідник, чи треба повернутися до його початку (тобто закрити його, а потім знову відкрити).

**Постановка завдання**

В Постановці задачі потрібно розкрити такі аспекти:

* Мета розробки ПЗ, призначення ПЗ
* Вхідні дані.
* Вихідні дані, очікувані результати.
* Функції обробки інформації (коротка технологічна схема за принципом введення інформації → контроль інформації → оброблення → формування результатів ), які відображують через які процеси вхідні дані трансформуються у вихідні.

Вхідні дані повинні бути чітко визначені. Доцільно подавати їх у вигляді такої таблиці:

|  |  |  |  |
| --- | --- | --- | --- |
| №№ | Назва атрибута | Формат | Обмеження |
| 1 | 2 | 3 | 4 |
|  |  |  |  |

Порядок визначення формату та обмежень залежить від встановлених в організації – розробника стандартів. Наприклад, можна надавати визначення відповідно обраної мови програмування або виразами "текст" чи "число", або визначенням формату таким як "ХХХ" - ціле, "ХХ.ХХ" –десяткове з 2 знаками після коми, "ДД.ММ.РРРР" – дата.

За аналогічною схемою визначаються і вихідні дані.

**Постановка завдання для практики.**

Головний офіс мережі з 4-х магазинів зібрав інформацію за рік по групі товарів (3 найменування товарів) за форматом форми "3-торг квартальна". Потрібно побудувати аналітичний звіт за цією інформацією (розрахунок абсолютних, середніх та відносних показників).

***Опис функцій обробки інформації***

Функції зазвичай описуються словесно та схематично. Прикладом такої схеми може бути блок-схема.

**Алгоритм рішення**

Алгоритм - сукупність правил, що однозначно визначають процес перетворення вхідних і проміжних даних у результат рішення задачі. Опис алгоритму являє собою загальну схему рішення задачі. Алгоритмічний процес це - процес послідовного перетворення конструктивних об'єктів, що проходить дискретними кроками (тобто зміна відбувається стрибкоподібно), кожний крок полягає в зміні одного конструктивного об'єкта іншим. Алгоритми характеризуються обчислювальною складністю і ємкісною складністю. За видом використовуваної обчислювальної моделі алгоритми діляться на послідовні (або детерміновані), паралельні (або недетерміновані), розподілені та ін.

Алгоритм може бути реалізований в обчислювальних системах, якщо він містить тільки елементарні команди, тобто не потребуючими деталізації, можна вважати такі команди або операції:

1) початок, кінець;

2) список даних;

3) введення, виведення;

4) обчислювальні операції, реалізовані оператором присвоювання

Для алгоритму характерні наступні властивості:

- детермінірованість, чи визначеність, тобто однозначність –його розуміння для будь-якого виконавця, що приводить до точного виконання однієї і тієї ж послідовності дій;

- результативність, чи спрямованість , тобто властивість досягнення за кінцеве число досить простих кроків шуканого результату розглянутої задачі;

- масовість, тобто придатність для рішення будь-якої задачі з деякого класу задач.

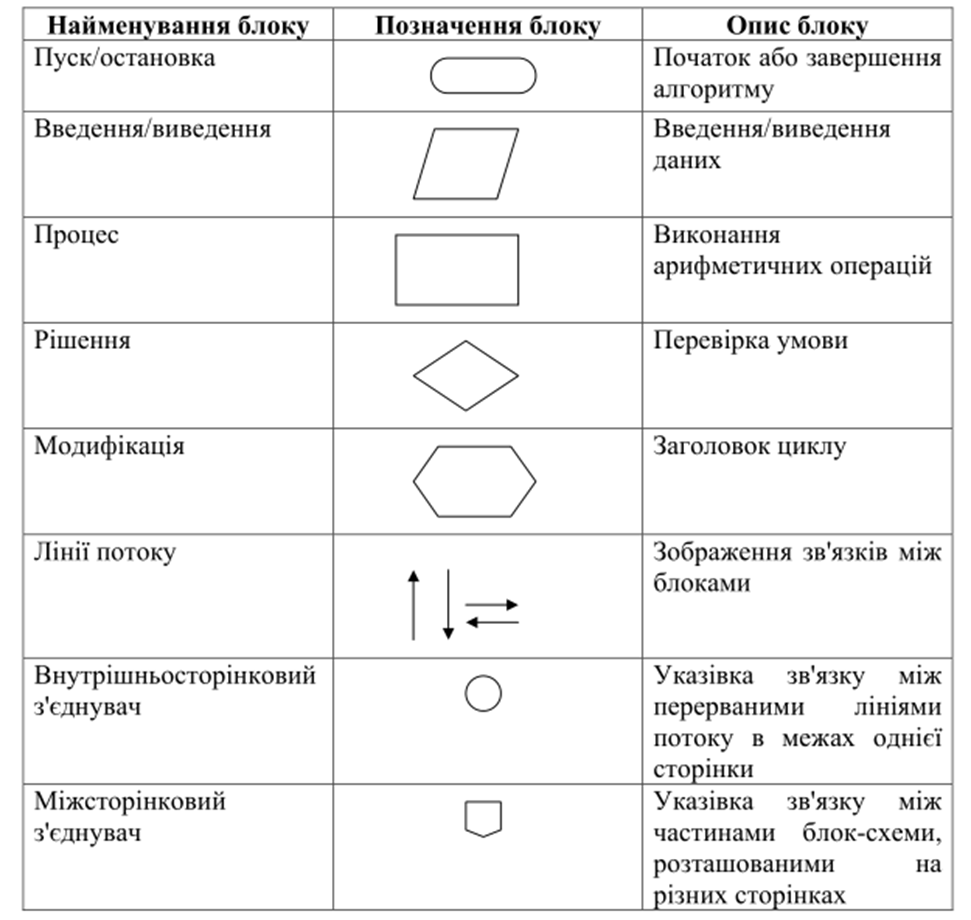
Розрізняють наступні способи представлення алгоритмів: текстуальний, операторний і графічний.

Найбільше поширення в даний час одержав графічний спосіб, при якому обчислювальний процес розчленовується на окремі операції, що відображаються у виді умовних графічних символів (блоків). Серед графічних способів найпростішим є блок-схема.

**Побудова блок- схем**

Для побудови блок-схеми використовуються графічні символи представлені в табл.1. Блоки з'єднуються між собою в визначеної послідовності лініями чи стрілками. Усередині блоків у виді формул чи тексту вказується інформація, що пояснює, характеризує виконувані ними дії. Блоки звичайно мають наскрізну нумерацію. Номер ставиться у верхньому лівому куті блоку в розриві. ліній.

Таблиця 1. Символи для створення блок-схеми



Блоки з'єднуються між собою в визначеної послідовності лініями чи стрілками. Усередині блоків у виді формул чи тексту вказується інформація, що пояснює, характеризує виконувані ними дії. Блоки звичайно мають наскрізну нумерацію. Номер ставиться у верхньому лівому куті блоку в розриві ліній.

Теорія структурного програмування доводить, що алгоритм будь-якого ступеня складності можна побудувати за допомогою основного базового набору структур:

1) послідовна (лінійна) структура (рис.1);

2) структура, що розгалужується (рис. 2);

3) циклічна структура (рис. 3).

Найпростішими для розуміння і використання є лінійні структури (рис. 1). Лінійним називається алгоритм (фрагмент алгоритму), у якому окремі розпорядження виконуються незалежно від значень вихідних даних і проміжних результатів

**Алгоритм лінійної структури**

Лінійним називається алгоритм (фрагмент алгоритму), в якому окремі команди виконуються послідовно друг за другом, не залежно від значень вхідних даних і проміжних результатів. Приклад алгоритму лінійної структури (рис. 1).

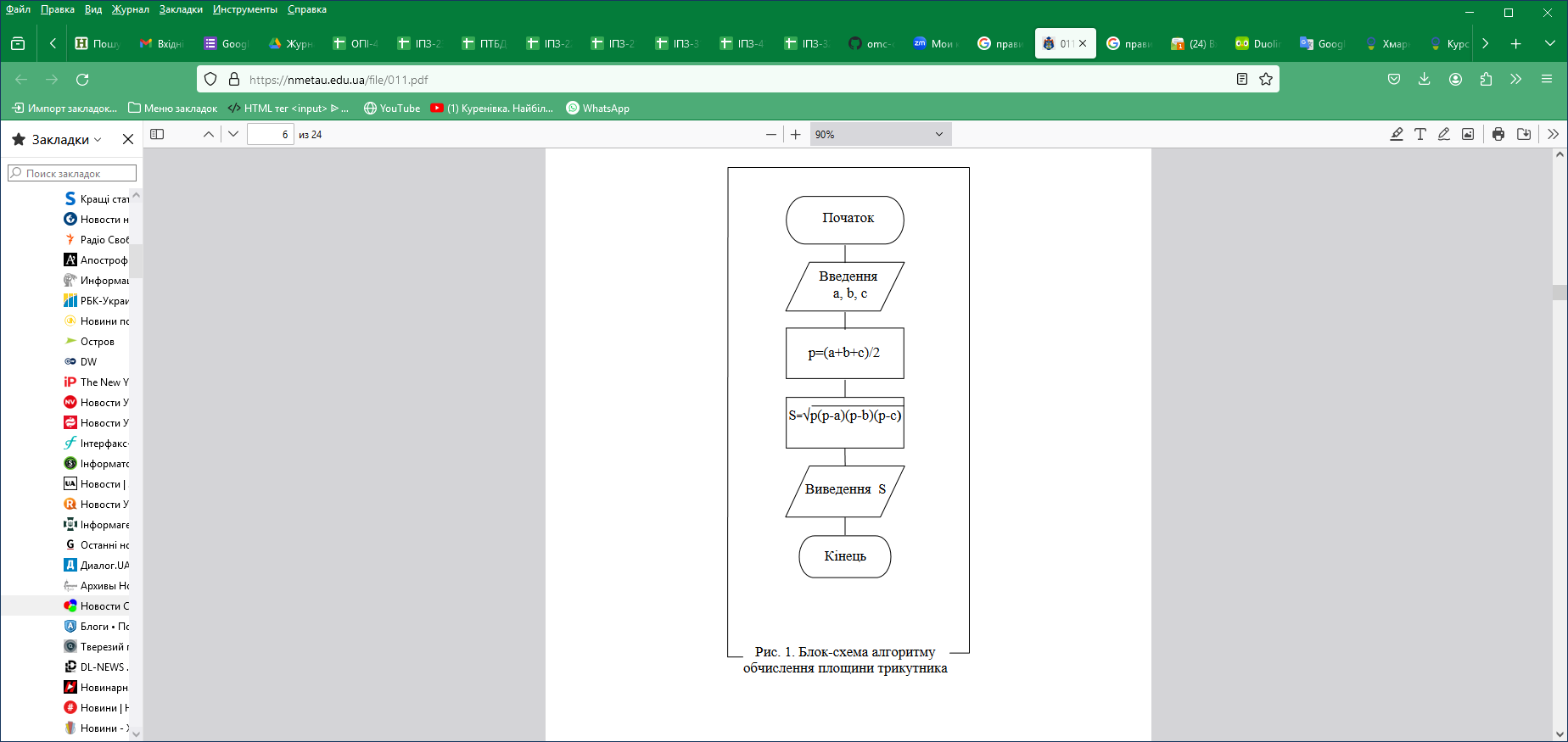


Рисунок 1 - Блок-схема лінійного алгоритму

**Блок-схеми процесів, що розгалужуються**

Часто для подальшої деталізації використовуються структури, що розгалужуються, тобто такі, в яких у залежності від вихідних даних або проміжних результатів алгоритм реалізується в одному з декількох, заздалегідь передбачених (можливих) напрямків. Такі напрямки часто називаються гілками. Кожна гілка може бути будь-якого ступеня складності, а може взагалі не містити команд, тобто бути виродженою. Вибір тієї або іншої гілки здійснюється в залежності від результату перевірки умови з конкретними даними. У кожному випадку алгоритм реалізується тільки по одній гілці, а виконання інших виключається. Приклад алгоритму, що розгалужується (рис. 2).

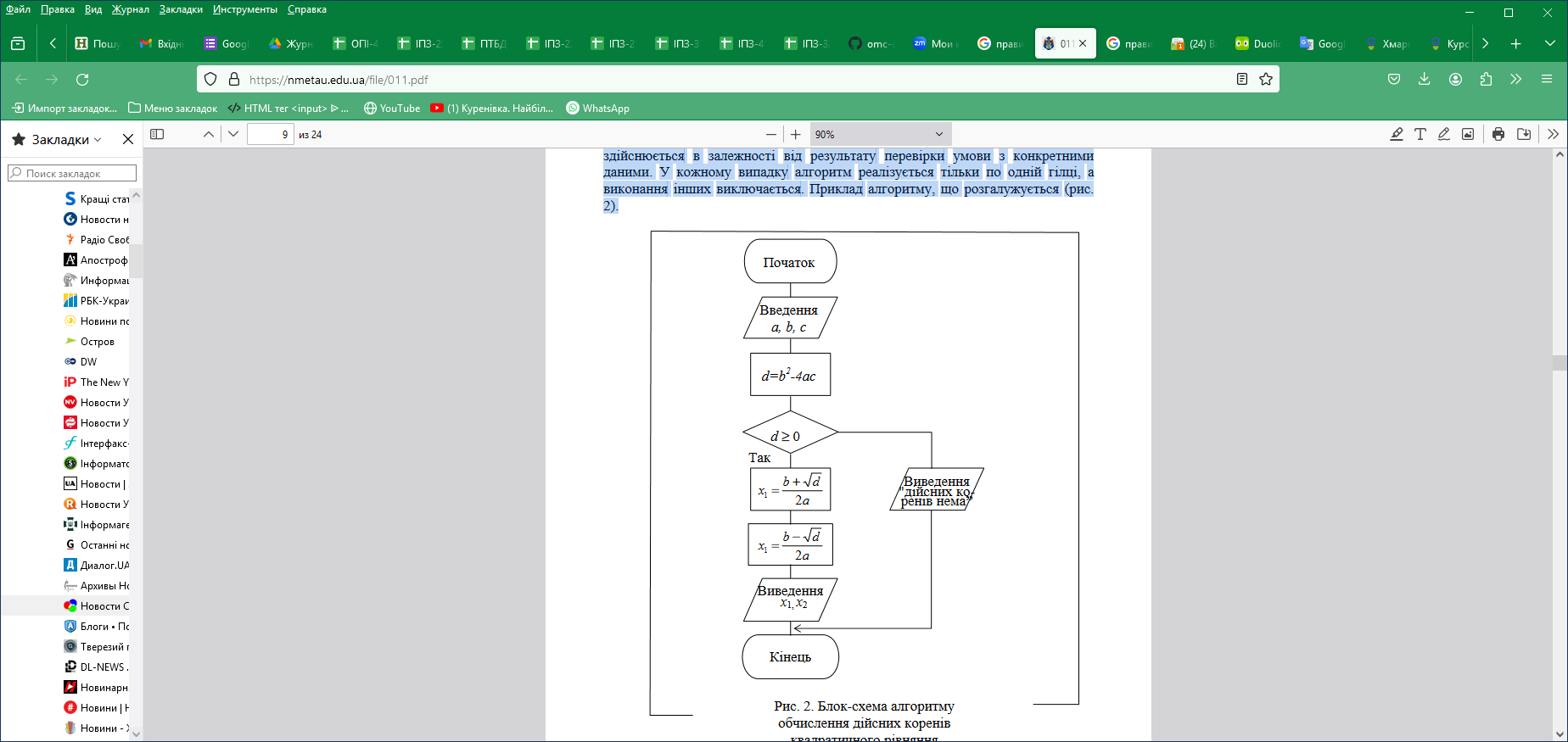


Рисунок 2 - Блок-схема алгоритму обчислення дійсних коренів квадратичного рівняння

**Блок-схеми циклічних обчислювальних процесів**

Виконання деякої частини алгоритму багаторазово при різних значеннях деяких змінних називається циклічним обчислювальним процесом. Циклами називаються повторювані ділянки обчислень. Для організації циклів необхідно: задати початкове значення змінної, що визначає цикл (параметр циклу), зміняти цю змінну перед кожним повторенням циклу, перевіряти умову продовження (закінчення) циклу.

У циклічних алгоритмах виконання деяких операторів (груп операторів) здійснюється багаторазово з тими ж або модифікованими даними У залежності від способу організації кількості повторень циклу розрізняють три типи циклів:

1) цикл із заданою умовою закінчення роботи (ЦИКЛ – ДО / DO - WHILE);

2) цикл із заданою умовою продовження роботи (ЦИКЛ – ПОКИ / WHILE);

3) цикл із заданою умовою повторень роботи (ЦИКЛ 3 ПАРАМЕТРОМ / FOR).

Приклади циклічного алгоритму (рис. 3- 8).

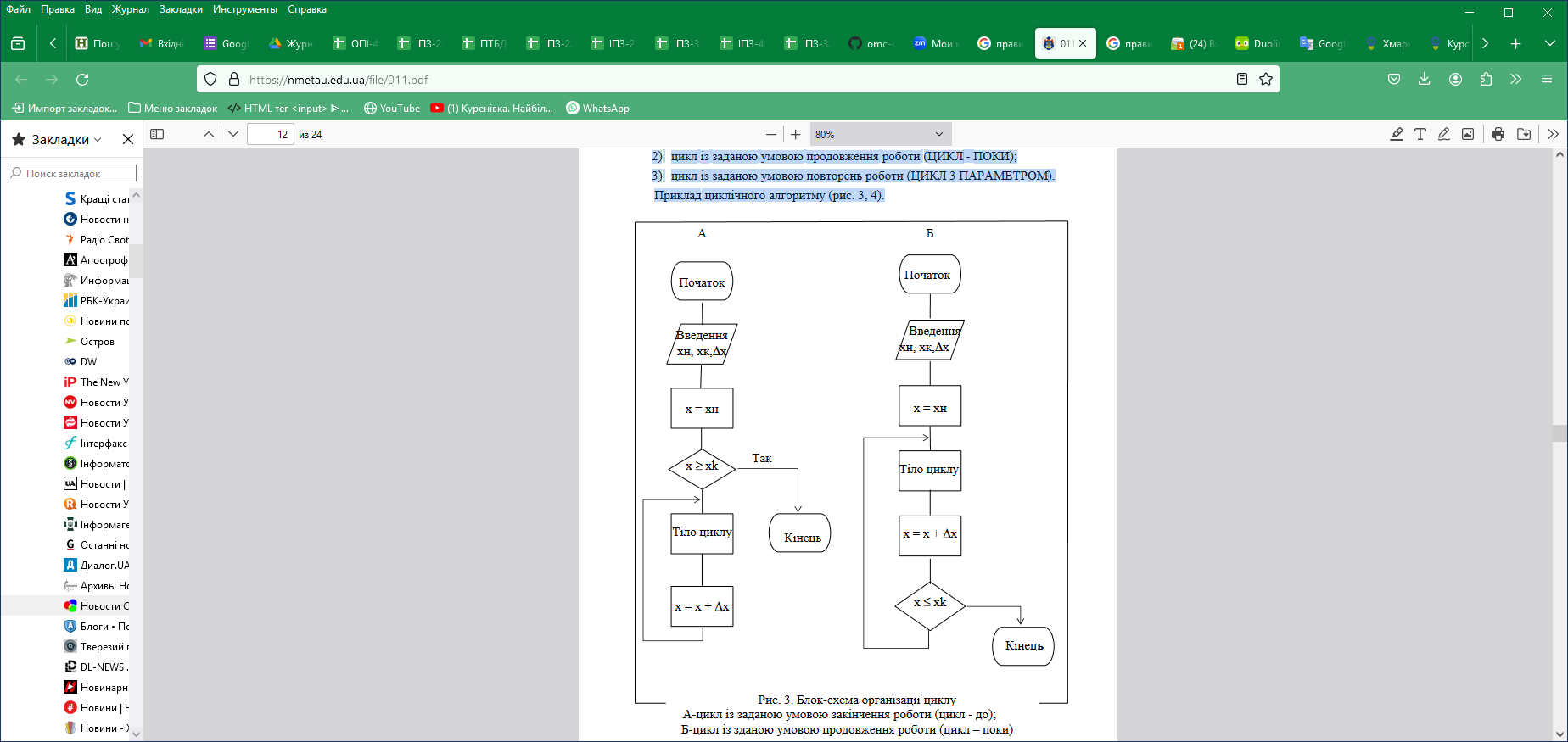


Рисунок 3 - Блок-схема організаціі циклу

А-цикл із заданою умовою закінчення роботи (цикл – до/DO);

Б-цикл із зданою умовою продовження роботи (цикл – поки/WHILE)

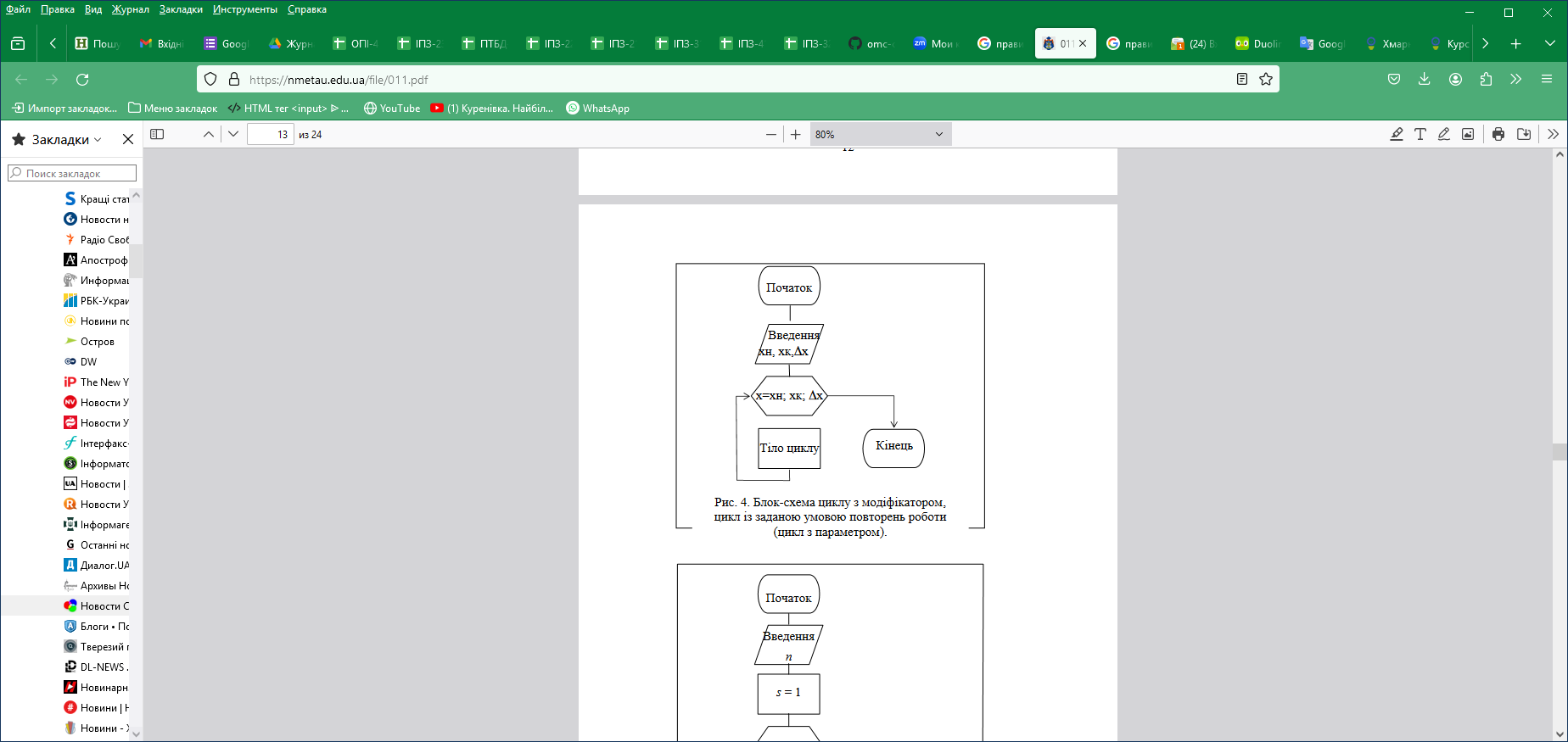


Рисунок 4 - Блок-схема циклу з модіфікатором, цикл із заданою умовою повторень роботи (цикл з параметром / FOR).

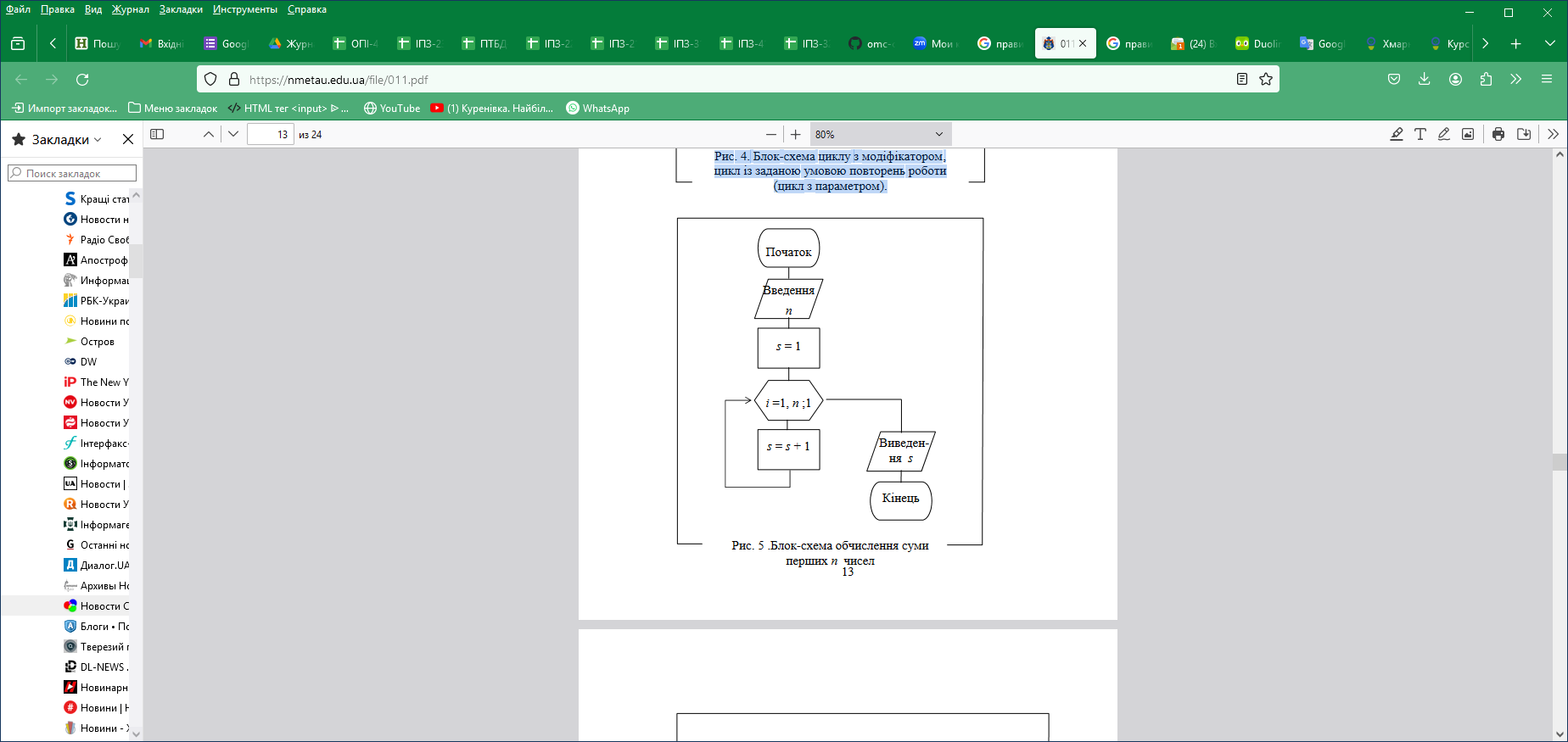


Рисунок 5 - Блок-схема обчислення суми перших n чисел

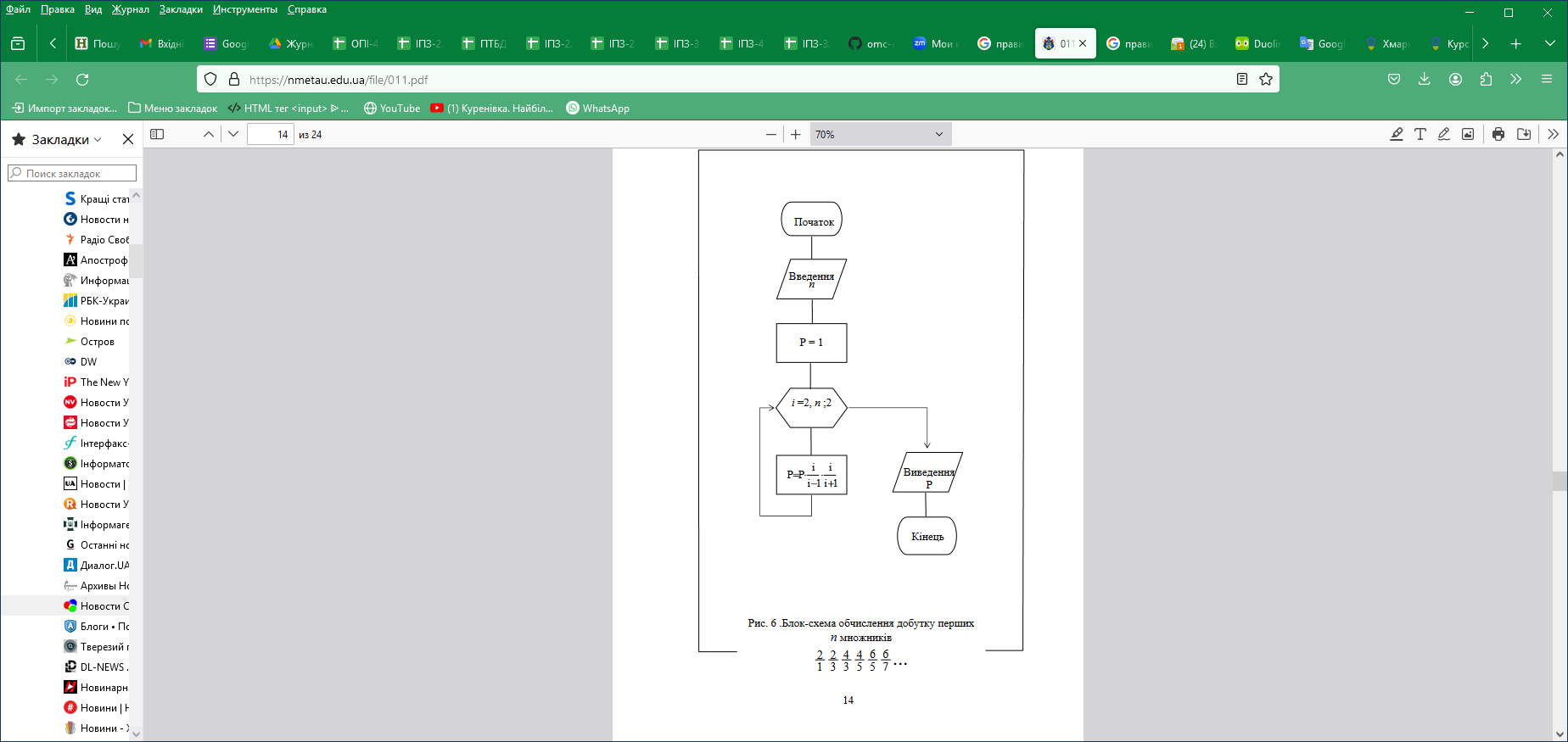


Рисунок 6 - Блок-схема обчислення добутку перших n множників

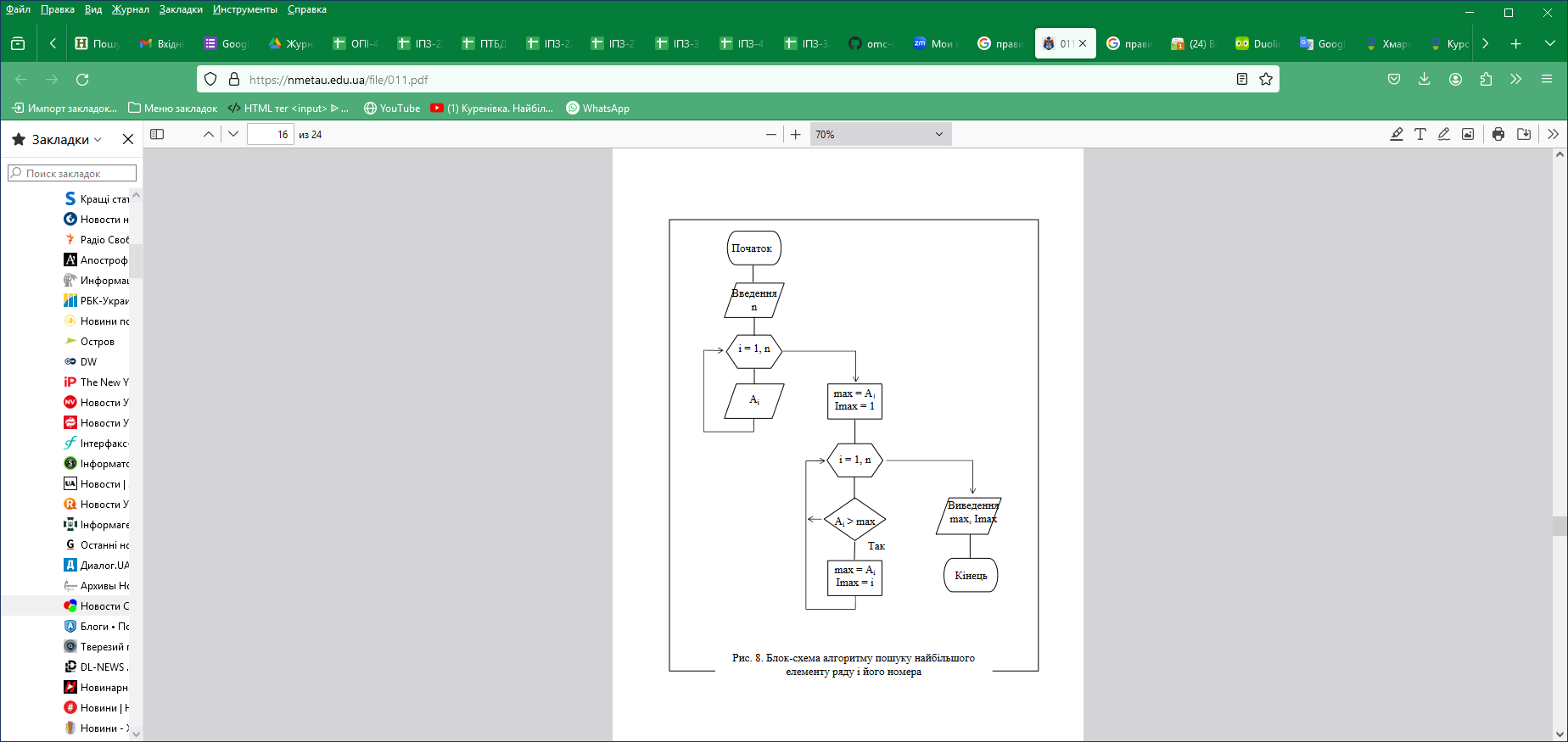


Рисунок 8 - Блок-схема алгоритму пошуку найбільшого елементу ряду і його номера