**Практична робота №1.** Алгоритми і алгоритмізація. Принципи побудови алгоритму вирішення задач.

Накреслити блок-схему системи ведення звітності головного офісу мережі супермаркетів, яка щоденно збирає звіти про продаж товарів в своїх магазинах, виконує їх перевірку та накопичення, а також формує зведений звіт про продажі.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

***Інформаційна система (ІС)*** - взаємозв’язана сукупність засобів, методів і персоналу, яка використовується для зберігання, обробки та надання інформації з метою розв’язання конкретного завдання

***Обробка інформації́*** — вся сукупність операцій (збирання, введення, записування, перетворення, зчитування, зберігання, знищення, реєстрація), що здійснюються за допомогою технічних і програмних засобів, включаючи обмін по каналах передачі даних.

Інформація — це відомості, представлені в документах і масивах інформації на машинних носіях. Інформація ІС відповідає на запитання «що?», відповідає поняттю предмета і засоби праці. Як предмет праці інформація є об'єктом збору, реєстрації, обробки, збереження, передачі. Як засіб праці керуюча інформація впливає на об'єкт управління. Інформація може розглядатися на синтаксичному, семантичному і прагматичному рівнях представлення. Інформаційні технології обумовлюють способи, методи і засоби збору, реєстрації, передачі, збереження, обробки і видачі (поширення або публікації) інформації в ІС. Інформаційні технології відповідають на запитання «як, за допомогою чого?».

***Звітність*** – набір узагальнюючих показників, які використовуються для надання підсумкової характеристики явища, об’єкту або процесу. Звітність є підсумковим етапом роботи ІС підприємства, вона ґрунтується на всій інформації, що була зібрана ІС підприємства, належним чином опрацьована і представлена у вигляді, придатному для подання внутрішнім та зовнішнім користувачам.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Інформація первинних документів |  | Результати обробки первинної інформації |  | Показники звітності |

***Алгоритмізація процесів оброблення інформації***

Робота по розв’язанню прикладної задачі на комп’ютері проходить через такі етапи:

1. Постановка задачі та її змістовний аналіз.
   1. Визначити умову задачі:

* Що дано?
* Що необхідно?
* Які дані допустимі?
* Які результати і в якому вигляді повинні бути отримані?
  1. Провести змістовний аналіз, направлений на уточнення мети вирішення задачі, її компонентів, вихідних даних.
  2. Визначити, за яких умов можливе отримання необхідних результатів, а за яких - ні.
  3. Визначити, які результати будуть вважатися правильними.

1. Формалізація задачі, вибір методу її розв’язання.

2.1 Розгорнутий змістовний опис задачі замінити її математичною моделлю за допомогою математичних залежностей.

2.2 Обгрунтоване обрання методу вирішення задачі.

1. Складання алгоритму на основі вибраного методу.

При складанні алгоритму треба враховувати всі його властивості.

1. Складання програми.

Програмування (складання програми) – кодування складеного алгоритму на одній з мов програмування.

1. Тестування і налагодження програми.

Перевірка правильності роботи програми за допомогою тестів і виправлення виявлених помилок. *Тест* – це спеціально підібрані вихідні дані та результати, отримані за цих даних.

1. Остаточне виконання програми, аналіз результатів.

6.1 Після налагодження програми прибрати всі налагоджувальні засоби.

6.2 Після остаточного виконання програми виконати аналіз результатів. Можлива зміна самого підходу до вирішення задачі і повернення до першого етапу для повторного виконання всіх етапів.

ПРИГАДАЄМО. ***Алгоритм і його властивості***

Алгоритм – це деяке правило перетворення інформації. Він указує, які операції оброблення даних і в якій послідовності необхідно виконати, щоб одержати розв'язок задачі.

***Алгоритм*** *— точне розпорядження, що визначає обчислювальний процес, який веде від початкових даних, що змінюються, до шуканого результату.*

Ефективним методом побудови алгоритмів є метод покрокової деталізації, при якому завдання розбивається на кілька простих підзадач (модулів), і для кожного модуля створюється власний алгоритм.

Здебільшого модуль реалізує певний процес обробки інформації і застосовується як для окремого використання, так і для включення модуля в інші алгоритми. Застосування модульності при створенні алгоритмів дозволяє розбити великі задачі на незалежні блоки (модулі), усуває повторення стандартних дій і значно прискорює процес відлагодження алгоритму в цілому. Найчастіше алгоритм складається з головного модуля, який містить декілька інших модулів, створених раніше. Використовуючи модулі як складові великої конструкції, можна створювати алгоритми будь-якого ступеня складності, і при цьому не втрачати контролю за функціонуванням алгоритму всієї задачі. Такий метод називається структурним проектуванням алгоритму “зверху – донизу”, є універсальним і може використовуватися як для обчислювальних процесів так і для процесів реального життя.

***Схеми алгоритмів***

Для опису алгоритмів використовують такі способи, як словесний опис послідовності обчислень, аналітичний (у вигляді формул), графічний (у вигляді схем і діаграм), псевдокод, запис алгоритмічною мовою.

Запис алгоритму алгоритмічною мовою потребує точ­ного дотримання правил цієї мови, оскільки він має бути зрозумілим не тільки людині, а й комп'ютеру. Псевдокод займає проміжне місце між словесним описом алгоритму і його записом алгоритмічною мовою. У цьому способі вживаються конструкції, близькі до алгоритмічної мови, але не вимагається повного дотримання всіх її правил, оскільки він призначений для сприйняття людиною.

Велике поширення дістав графічний спосіб опису алгоритму у вигляді схем.

***Схема алгоритму*** *— графічне зображення його структури, в якому кожний етап процесу перероблення даних подається у вигляді різних геометричних фігур (сим­волів).*

Ці фігури з'єднуються між собою *лініями потоку,* які для кожного етапу вказують можливих наступників. Усередині фігури дається опис відповідного етапу, якщо він не є занадто громіздким. У противному разі такий опис наводиться в додатку до схеми, а замість нього у відповідній фігурі записується номер або яке-небудь позначення цього етапу. Біля фігури можуть бути деякі зауваження, наприклад такі, що показують, в якому випадку вибір наступника буде робитися відповідно до лінії потоку.

Символам присвоюють порядкові номери, які проставляються в розриві лінії контуру в лівій частині верхнього боку зображення символу. Лінії потоку проводять паралельно лініям зовнішньої рамки схеми. Напрямок лінії потоку зверху вниз і зліва направо прийнято за основний і, якщо вони не мають зламів, стрілками їх можна не позначати. В інших випадках їхній напрямок обов'язково позначають стрілкою. Лінію потоку, як пра­вило, підводять до середини символу.

Відстань між паралельними лініями потоку має бут не меншою від 3 мм, між іншими символами — не меншою від 5 мм. Лінію потоку можна обривати, використовуючи на місці обриву з'єднувачі, якщо схему виконай на двох і більше аркушах, або якщо символи, які з'єднуються, розташовано на значній відстані один від одного.

Запис у середині символу або поруч із ним потрібно виконувати машинописом з одним інтервалом або креслярським шрифтом.

Перевагою схем є те, що з їх допомогою можна наочно зобразити структуру алгоритму в цілому, відобразивши його логічну суть (показати розгалуження шляхи розв'язання задачі залежно від виконання деякої умови відобразити багаторазове повторення окремих етапів обчислювального процесу).

Графічне зображення алгоритму у вигляді схем полег­шує складання програми для розв'язання задачі не комп'ютері.

У табл. 1 наведено символи, що найчастіше викори­стовуються в схемах алгоритмів.

Розмір **а** має вибиратися з ряду 10, 15, 20 мм. Допускається збільшувати розмір **а** на число кратне 5. Розмір **b=1,5а.** При виконанні умовних графічних позначень автоматизованим методом, розміри геометричних елементів округлюються до значень, що визначаються технічними можливостями пристроїв, що використовуються.

*Таблиця 1.*

| Найменування символу | Графічне зображення | Функція символу |
| --- | --- | --- |
| 1. Процес |  | Виконання операції або групи операцій, завдяки яким змінюються значення, форма подання або розташування даних |
| 2. Розв'язу­вання |  | Вибір напрямку виконання алгоритму або програми залежно від деяких змінних умов |
| 3. Введення - виведення |  | Перетворення даних у форму, придатну для оброблення (введення) або відображення здобутих результатів (виведення) |
| 4. З'єднувач |  | Зазначення зв'язку між перерваними лініями потоку, що зв'язують символи |
| 5. Пуск-зупин |  | Початок, кінець, переривання процесу оброблення даних або виконання програми |
| 6. Коментар |  | Зв'язок між елементом схеми і  Пояснення |
| 7. Лінія потоку |  | Зазначення послідовності зв'язків між символами |
| 8. Міжсторінковий з'єднувач |  | Зазначення зв'язку між роз'єднаними частинами схем алгоритмів і програм, розташованих на різних аркушах |

***Графічне зображення різних видів обчислювальних процесів***

Обчислювальні процеси, що виконуються за заданим алгоритмом, поділяють на три основні види:

• лінійні;

• розгалужені;

• циклічні.

Вони, як правило, є окремими частинами обчислювального процесу, тоді як загальний обчислювальний процесмає складнішу (комбіновану) структуру.

У лінійному обчислювальному процесі всі операції ви­конуються послідовно у порядку їх запису. Типовим прикладом такого процесу є стандартна обчислювальна схема, що складається з трьох етапів:

• введення початкових даних;

• обчислення за формулами;

• виведення результату.

***Графічне зображення розгалужених обчислювальних процесів***

*Обчислювальний процес називається* ***розгалуженим,*** *якщо для здобуття кінцевого результату передбачається вибір одного з кількох можливих напрямів обчислень (гілок) залежно від результату перевірки деякої умови.*

Розгалужений обчислювальний процес, що скла­дається з двох гілок, називається *простим,* а з більшої кількості гілок — *складним.* Напрям обчислень вибирається перевіркою, внаслідок якої можливі два виходи:

«так» — умову виконано;

«ні» — умову не виконано.

Умова вказується всередині символу «розв'язування».

***Графічне зображення циклічних обчислювальних процесів***

Для більшості обчислювальних процесів характерною є повторюваність дій.

***Циклом*** *називається послідовність дій, що багаторазово повторюється, а обчислювальний процес, який містить цикл, має назву* ***циклічного.***

Керування повторенням циклу здійснюється за допомогою змінної, яка називається *параметром циклу.* Спочатку цьому параметру присвоюється деяке початкове значення. Потім цикл виконується зі зміною параметра при кожному повторенні від початкового до кінцевого значень на величину, що називається *кроком циклу.*

Крок циклу може бути додатнім або від’ємним. Залежно від цього параметр циклу зростає або змен­шується. Цикл припиняється, якщо параметр циклу має значення, що лежить поза межами діапазону між початко­вим і кінцевим значеннями.

Розрізняють три види циклів:

• з передумовою;

• з постумовою (післяумовою);

• з параметром.

Перші два види циклів використовуються тоді, колі заздалегідь невідома кількість повторень .

У *циклі з передумовою* спочатку перевіряється умова (звідси і назва — цикл з передумовою) і, якщо умова виконується, то здійснюється дія. Потім знову перевіряється умова і т. д. Виконання циклу припиняється, коли умова перестає виконуватися. Для цього необхідно, щоб дія в циклі впливала на зміну умови. Інакше відбудеться "зациклювання" — нескінченне виконання циклу. Воно є типовою помилкою в разі використання циклів.

*Цикл із постумовою* виконується аналогічно, але умова перевіряється після виконання дії (тому цикл і називається з постумовою). Повторення дії відбувається тоді коли умова не виконується.

Дія в циклі з постумовою виконується завжди хоча один раз, а з передумовою може не виконуватися жодного разу, якщо з самого початку умова не виконується.

*Цикл із параметром* будується на підставі одного з перших двох видів циклів. Здебільшого використовуєте цикл із передумовою.

Можлива інша схема циклу з параметром. У ньому послідовно виконуються такі типові операції:

1. Завдання початкового значення параметра циклу, що використовується під час виконання першого кроку.

2. Виконання циклічної ділянки.

3. Зміна параметра циклу, що забезпечує обчислення результату з новими початковими даними і перехід до на­ступного кроку.

4. Перевірка на закінчення циклу порівнянням поточ­ного значення параметра циклу з кінцевим.

5. Перехід до повторення циклічної ділянки, якщо параметр циклу не перевищив кінцевого значення, інакше — виконання чергової дії або виведення резуль­тату.

***Інформаційні моделі, процеси і алгоритми оброблення інформації***

***Інформаційна модель –*** *це матеріальний або уявлений подумки об’єкт, який використовується замість об’єкта-оригіналу чи явища-процесу) при його дослідженні і зберігає інформацію про деякі важливі для даного дослідження типові риси і властивості оригіналу.*

**Інформаційним процесом** називається взаємодія між повідомленням і відправником і споживачем інформації. Іншими словами, **інформаційні процеси** − це сукупність послідовних операцій (реєстрація, передача, накопичення, зберігання, оброблення, видача інформації), дій і зв'язків з обміну інформацією, що здійснюються в системі комунікацій.

До оброблення інформації входять такі взаємопов'язані інформаційні процеси: ***збір; аналітико-синтетична переробка (перетворення); зберігання; пошук; розповсюдження (поширення).***

***1. Збір інформації***− процес полягає в отриманні інформаційними службами повідомлень всіх видів по різних каналах зв'язку. Цей початковий процес - найважливіший для всіх наступних інформаційних процесів. Інформаційні повідомлення, зафіксовані в документах і на інших носіях інформації збираються у інформаційній базі (**базі даних або у масиві інформації)**.

***2. Процес обробки інформації -*** наступний за процесом збору інформації , який поділяється на обробку: 1) технічну - контроль і 2) аналітичну - перетворення. **Технічна обробка** полягає в обліку та реєстрації надходження повідомлень, перевірку їх коректність, в т.ч. на дублювання з наявними у інформаційній базі. **Аналітична обробка** полягає в інформаційному аналізі і синтезі повідомлень; й інакше називається аналітико-синтетичною обробкою (або переробкою) інформації.

***3. Процес зберігання інформації***- це процес, пов'язаний із забезпеченням збереженості зібраних і оброблених (в інформаційних службах) повідомлень для передачі їх у просторі та часі.

***4. Інформаційний пошук***− це процес знаходження у певній упорядкованій множині тих повідомлень, які відповідають запиту споживача або містять необхідні споживачеві факти, дані. Іншими словами, інформаційний пошук розуміється як сукупність логічних і технічних операцій, що мають кінцевою метою знаходження відомостей, фактів, даних, релевантних (міра відповідності результатів пошуку завданню, поставленому в пошуковому запиті) запиту користувача.

***5. Розповсюдження (поширення) інформації*** − це завершальний інформаційний процес, суть якого полягає у видачі відповіді на запит споживача.

Розрізняють два основні режими розповсюдження інформації (або інформування): довідковий і поточний. **Довідковий режим** передбачає доведення до користувача ретроспективної інформації, у відповідь на разовий запит. **Поточне інформування** полягає в наданні користувачам інформації про нові надходження в систему і здійснюється масовими, груповими та індивідуальними методами.

Схема інформаційного процесу у сучасному суспільстві

**Контрольні запитання**

1. *Поняття інформаційної моделі.*
2. *Назвіть етапи вирішення прикладних задач з використання ПК.*
3. *Що ми розуміємо під поняттям “алгоритм”?*
4. *Які є способи опису алгоритмів?*
5. *Поняття схеми алгоритму та правила її побудови.*
6. *Які властивості повинен мати алгоритм?*
7. *Які є три головні алгоритмічні конструкції?*
8. *З чого складаються прості (лінійні) алгоритми?*
9. *Який процес називається розгалуженим?*
10. *Який процес називається циклічним?*
11. *Види циклічних обчислювальних процесів, їх відмінність.*

# За матеріалами :

Бібліотека економіста. URL: <https://library.if.ua/book/80/5675.html>

Інформаційні системи і технології в статистиці (2003) . URL: https://library.if.ua/books/80.html