**Лабораторна робота № 2**

**Тема: Базові поняття теорії алгоритмів**

**Мета роботи:** Отримати практичні навички складання простих алгоритмів рішення задач з обчислення числових функцій.

**Форма звіту:** виконання аудиторного і домашнього завдань.

**Теоретична частина**

Слово "Алгоритм" походить від імені середньовічного вченого Мухаммеда ібн Муси аль-Хорезмі (787-850 гг), що жив в Середній Азії. У XVIII столітті, коли праці аль-Хорезмі були перекладені з арабської мови на латинь, його ім'я записали так: "Algorithmus", але люди винаходили алгоритми задовго до аль-Хорезмі.

**Алгоритм -** це зрозумілий і точний опис виконавцеві зробити певну послідовність дій, спрямованих на досягнення вказаної мети або рішення поставленої задачі, за кінцеве число кроків.

Для запису алгоритмів використовуються:

**Блок-схема -** поширений тип схем, що описує алгоритми або процеси, зображуючи кроки у вигляді блоків різної форми, сполучених між собою стрілками.

Складання алгоритмів графічним способом підкоряється двом ГОСТам:

1. ГОСТ 19.002-80, відповідає міжнародному стандарту ИСО 2636-73. Регламентує правила складання блок-схем.
2. ГОСТ 19.003-80, відповідає міжнародному стандарту ИСО 1028-73.

**Форми представлення алгоритму**

Алгоритм фіксується різними способами: на природній мові, за допомогою спеціальних схем, графічно, на алгоритмічній мові.

**Приклади**

*Алгоритми на природній мові*

1. Є два глеки місткістю 3 і 8 л. Напишіть алгоритм або просто відповідь, як можна набрати з річки 7 л води.

Рішення. Алгоритм: Наповнюєш восьмилітровий глек за допомогою трилітрового: 3+3+2. Після цього в глеку трилітровому залишається 1 літр.Виливаєш усе з восьмилітрового, переливаєш 1 літр з трилітрового, потім 3+3, у результаті - сім літрів.

1. Селянин стоїть на лівому березі річки з вовком, козою і капустою. Йому треба перевезти усе це на правий берег. Але його човен занадто малий: він може узяти тільки одного пасажира - або вовка, або капусту, або козу. Як тут поступити?

Рішення. Алгоритм:

* 1. Перевези козу
  2. Повернутися
  3. Перевези вовка, забрати козу
  4. Повернутися
  5. Висадити козу, перевези капусту
  6. Повернутися
  7. Перевези козу.

*ГОСТ на опис блок-схем*

Для графічного представлення алгоритму використовують певні геометричні фігури. Таке представлення називається блок-схемою. Розміри і співвідношення розмірів фігур приводяться в ГОСТ 19-002-80 і ГОСТ 19-003-80. Згідно з ними усі розміри пов'язані з двома величинами: а і в, де а - величина, кратна 5, а в обчислюється за формулою в = 1,5а, допускається в = 2а.

У січні 1992 року введений новий ГОСТ 19-701-90. Він описує, як і де слід використовувати фігури. Згідно з ним допускаються наступні символи для зображення схем (табл.2.1).

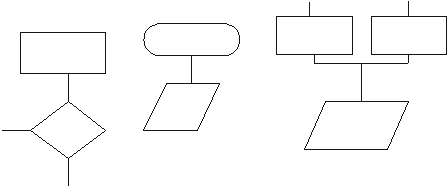
Таблиця 2.1

|  |  |
| --- | --- |
| flowcharts_terminatorТерминатор начала и конца работы функции | Терминатором начинается и заканчивается любая функция. Тип возвращаемого значения и аргументов функции обычно указывается в комментариях к блоку терминатора. |
| flowcharts_dataОперации ввода и вывода данных | В ГОСТ определено множество символов ввода/вывода, например вывод на магнитные ленты, дисплеи и т.п. Если источник данных не принципиален, обычно используется символ параллелограмма. Подробности ввода/вывода могут быть указаны в комментариях. |
| flowcharts_processВыполнение операций над данными | В блоке операций обычно размещают одно или несколько (ГОСТ не запрещает) операций присваивания, не требующих вызова внешних функций. |
| flowcharts_solutionБлок, иллюстрирующий ветвление алгоритма | Блок в виде ромба имеет один вход и несколько подписанных выходов. В случае, если блок имеет 2 выхода (соответствует оператору ветвления), на них подписывается результат сравнения – “да/нет”. Если из блока выходит большее число линий (оператор выбора), внутри него записывается имя переменной, а на выходящих дугах – значения этой переменной.  **Не обозначать циклы!** **Только if или case.** |
| flowcharts_procedureВызов внешней процедуры | Вызов внешних функций помещается в прямоугольник с дополнительными вертикальными линиями. |
| flowcharts_loopНачало и конец цикла | Символы начала и конца цикла содержат имя и условие. Условие может отсутствовать в одном из символов пары. Расположение условия, определяет тип оператора, соответствующего символам на языке высокого уровня – оператор с предусловием (while) или постусловием (do … while). |
| flowcharts_preprocessПодготовка данных | Символ “подготовка данных” в произвольной форме (в ГОСТ нет ни пояснений, ни примеров), задает входные значения. Используется обычно для задания циклов со счетчиком. |
| flowcharts_connectorСоединитель | В случае, если блок-схема не умещается на лист, используется символ соединителя, отражающий переход потока управления между листами. Символ может использоваться и на одном листе, если по каким-либо причинам тянуть линию не удобно. |
| flowcharts_comment | Комментарий может быть соединен как с одним блоком, так и группой. Группа блоков выделяется на схеме пунктирной линией. |

**Правила з'єднання блоків в програмі**

Усі фігури з'єднуються лініями (вертикальними і горизонтальними) до середини блоку. Напрям вниз і управо є основним, і стрілки напряму не вказуються, в інших випадках вказуються обов'язково. Усередині фігури вказується операція. Кожен блок має тільки одну точку входу і тільки одну точку виходу (окрім блоку "умова", де може бути два і більше виходів, на кожному позначається причина/умова). Декілька ліній можуть з'єднуватися над блоком, низхідна лінія не може розбиватися.

Приклади правильної і неправильної блок-схеми:



Неправильні Правильні

**Види алгоритмів**

*Існують три види алгоритмів :*

*1. Лінійні.*

*2. Що розгалуджуються.*

*3. Циклічні.*

*3.1. По лічильнику.*

*3.2. Ітераційні.*

*(Знайти самостійно і записати визначення)*

**Приклади**

У лінійних алгоритмах кожен етап обчислень зводиться до виконання арифметичних операцій, які в процесі обчислень виконуються одноразово. У схемах таких алгоритмів блоки операцій виконуються послідовно один за одним.

Алгоритми, що розгалужуються, залежно від виконання або невиконання в них деяких умов здійснюють ту або іншу послідовність обчислень. При розгалуженні відбувається одноразовий прохід по одній з гілок рішення задачі. Класичний приклад алгоритму, що розгалужується – це алгоритм рішення квадратного рівняння ах2+bх+c=0 при будь-кому а, b, с. На додаток до попередніх блоків алгоритми, що розгалужуються, містять блок "рішення" (умова). В даному прикладі A, B, C, D, x, x1, x2 - змінні.

Спочатку перевіряють можливість А=0, тоді рівняння ах2+bх+с=0 зводиться до виду bх+ с =0, звідки . Потім обчислюють дискримінант D, якщо він негативний, рівняння коренів не має, якщо позитивний, будуть два корені, якщо рівний 0, обидва корені будуть однакові. Алгоритм представлений на рис.1.2.

Початок

D=B2-4AC

Введення А,В,С

А=0

X1=(-B+√D)/2A

X2=(-B-√D)/2A

Х=С/В

Виведення Х Корінь один

Коренів немає

Виведення Х1,X2 Корней два один

D<0

Кінець

так

так

ні

ні

Рис.1.2

Умова

Опертор1

Опертор2

Умова

Оператор

а) повні

а) неповні

Розгалуження можуть бути повні і неповні:

Циклічні алгоритми мають частину обчислень, які виконуються неодноразово. У загальному вигляді цикли містять три види дій:

1) підготовчі до циклу;

2) дії, які повторюватимуться (їх називають тіло циклу);

3) умови виходу з циклу.

Згідно з ГОСТом цикл зображувати так:



Тіло циклу

Умова

цикл 1

Підготовка

цикл 1

Циклічні алгоритми бувають декількох видів:

* цикли з лічильником застосовуються, коли заздалегідь відомо, скільки разів повторюватиметься послідовність дій;
* цикли ітераційні застосовуються, коли це невідомо, повтор блоків виконується, поки якась умова вірна. Умова може бути простою і складеною.

*Ітераційний цикл* - *процес повторення однієї і тієї ж дії, де результат попередньої дії приймається як початкові дані для наступного рішення.*

Розглянемо алгоритм виділення розрядів з числа 1579, тобто вимагається послідовно отримати 9, 7, 5, 1.

Очевидно, 9 вийде, якщо 1579 розділити на 10 і виділити залишок; 7 вийде, якщо цілу частку попереднього кроку (157) розділити на 10 і виділити залишок; 5 вийде, якщо цілу частку попереднього кроку (15) розділити на 10 і виділити залишок; і, нарешті, 1 вийде, якщо цілу частку попереднього кроку (1) розділити на 10 і виділити залишок.

Коли ж вимагається закінчити обчислення? При останньому обчисленні в частку запишеться 0. Це і можна вважати ознакою закінчення рахунку.

Словесний опис рішення вимагається формалізувати. Позначимо у - черговий розряд, N - проміжна частка. Фрази "залишок", "ціле" замінюватимуть функції отримання відповідно залишку і цілої частини.

У = залишок (N/10).

N = ціле (N/10).

тут = не знак рівності, а знак привласнення.

**Індивідуальнe завдання**

Дано натуральне число *n* (*n* ≤ 100). Скільки цифр у числі *n*