

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**ІНЖЕНЕРНА ТА КОМП'ЮТЕРНА ГРАФІКА**  
**ПРАВИЛА ВИКОНАННЯ СХЕМ АЛГОРИТМІВ, ПРОГРАМ,**  
**ДАНИХ І СИСТЕМ**

Методичні вказівки до вивчення теми дисципліни і контрольні завдання  
для студентів факультету інформатики та обчислювальної техніки

Київ  
НТУУ «КПІ»  
2012

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**ІНЖЕНЕРНА ТА КОМП'ЮТЕРНА ГРАФІКА**  
**ПРАВИЛА ВИКОНАННЯ СХЕМ АЛГОРИТМІВ, ПРОГРАМ,  
ДАНИХ І СИСТЕМ**

Методичні вказівки до вивчення теми дисципліни і контрольні завдання  
для студентів факультету інформатики та обчислювальної техніки

Затверджено на засіданні кафедри  
нарисної геометрії, інженерної та  
комп'ютерної графіки

Протокол № \_\_ від \_\_. \_\_. 2012р.

Київ  
НТУУ «КПІ»  
2012

**Інженерна та комп'ютерна графіка. Правила виконання схем алгоритмів, програм, даних і систем.** Методичні вказівки до вивчення теми дисципліни і контрольні завдання для студентів факультету інформатики та обчислювальної техніки/ Уклад.: В.В. Перевертун, Т.М. Надкернична. – К.: НТУУ «КПІ», 2012. – 25 с.

Гриф надано Вченою радою  
фізико-математичного факультету  
НТУУ «КПІ»  
(протокол № 4 від 18. 05. 2012 р.)

## **ІНЖЕНЕРНА ТА КОМП'ЮТЕРНА ГРАФІКА**

### **ПРАВИЛА ВИКОНАННЯ СХЕМ АЛГОРИТМІВ, ПРОГРАМ, ДАНИХ І СИСТЕМ**

Методичні вказівки до вивчення теми дисципліни і контрольні завдання для студентів факультету інформатики та обчислювальної техніки

Укладачі: Перевертун Валентина Вікторівна, канд. техн. наук, доц.,  
Надкернична Тетяна Миколаївна

Відповідальний редактор Ванін Володимир Володимирович, докт. техн. наук, проф.

Рецензент Тарабаров Сергій Борисович, канд. техн. наук, доц.

# Зміст

<b>ЗМІСТ</b> .....	<b>3</b>
<b>1 ЗАГАЛЬНІ ПОЛОЖЕННЯ</b> .....	<b>4</b>
<b>2 ВИДИ ТА ПРАВИЛА ПОЗНАЧАННЯ ПРОГРАМ І ПРОГРАМНИХ ДОКУМЕНТІВ</b> .....	<b>4</b>
<b>3 ПРАВИЛА ВИКОНАННЯ СХЕМ АЛГОРИТМІВ, ПРОГРАМ, ДАНИХ І СИСТЕМ</b> .....	<b>7</b>
3.1 Основні поняття .....	7
3.2 Опис символів, що їх використовують у схемах алгоритмів, ПРОГРАМ, ДАНИХ І СИСТЕМ.....	8
3.2.1 <i>Символи даних</i> .....	8
3.2.2 <i>Символи процесу</i> .....	10
3.2.3 <i>Символи ліній</i> .....	12
3.3 ПРАВИЛА ВИКОНАННЯ СХЕМ.....	13
<b>4 ЗМІСТ ГРАФІЧНОЇ РОБОТИ ТА ПОСЛІДОВНІСТЬ ЇЇ ВИКОНАННЯ</b> .....	<b>16</b>
<b>ДОДАТОК А</b> .....	<b>18</b>
<b>ДОДАТОК Б</b> .....	<b>23</b>
<b>ДОДАТОК В</b> .....	<b>24</b>
<b>ЛІТЕРАТУРА</b> .....	<b>24</b>

Метою даних методичних вказівок є ознайомлення студентів з основними нормативними положеннями, що діють у сфері розроблення та оформлення програмної документації, умовними позначками (символами), які використовують у схемах алгоритмів, програм, даних і систем, а також з правилами виконання цих схем.

Для практичного засвоєння поданого у цих вказівках довідкового матеріалу та здобуття навичок оформлення схем студентам пропонується скласти та графічно оформити схему алгоритму розв'язання обчислювальної задачі.

## **1 Загальні положення**

Основою національної нормативної бази у сфері документування програмного забезпечення є комплекс стандартів Єдиної системи програмної документації (ЄСПД), які встановлюють взаємопов'язані правила розроблення, оформлення та обігу програм і програмної документації. Регламентація зазначених процесів забезпечує можливість:

- уніфікації програмних виробів для взаємного обміну програмами та застосування раніше розроблених програм у нових розробках;
- зниження трудомісткості та підвищення ефективності розроблення, супроводження, виготовлення та експлуатації програмних виробів;
- автоматизації виготовлення та зберігання програмної документації.

Правила та положення, встановлені у стандартах ЄСПД, поширюються на програми та програмну документацію для обчислювальних машин, комплексів і систем незалежно від їх призначеності та сфери застосування.

## **2 Види та правила позначання програм і програмних документів**

Згідно з ГОСТ 19.101-77 „ЄСПД. Види програм і програмних документів” програми поділяють на компоненти і комплекси.

*Компонент* — це програма, яку розглядають як єдине ціле, яка виконує закінчену функцію та застосовується самостійно або у складі комплексу. *Комплекс* — це програма, яка складається з двох і більше компонентів і (або) комплексів, що виконують взаємопов'язані функції, та яка застосовується самостійно або у складі іншого комплексу.

До програмних відносять документи, що містять відомості, необхідні для розроблення, виготовлення, супроводження та експлуатації програми. Види програмних документів, їх коди та зміст відповідно до ГОСТ 19.101-77 наведено в таблиці 2.1.

**Таблиця 2.1** — Види програмних документів

<b>Вид програмного документа</b>	<b>Код виду документа</b>	<b>Зміст програмного документа</b>
Специфікація	—	Склад програми та документації на неї
Розпис утримувачів <b>правдників</b>	05	Перелік підприємств, на яких зберігають <b>правдники</b> програмних документів.
Текст програми	12	Запис програми з необхідними коментарями
Опис програми	13	Відомості про логічну структуру та функціонування програми
Програма та методика випробувань	51	Вимоги, що підлягають перевірці при випробуванні програми, а також порядок та методи їх контролювання.
Технічне завдання	—	Призначеність та область застосування програми, технічні, техніко-економічні та спеціальні вимоги, що їх ставлять до програми, необхідні стадії та терміни розроблення, види випробувань.
Пояснювальна записка	81	Схема алгоритму, загальний опис алгоритму та (або) функціонування програми, а також обґрунтування прийнятих технічних та техніко-економічних рішень.
Експлуатаційні документи	Відповідно до табл. 2.2	Відомості для забезпечення функціонування та експлуатації програми.

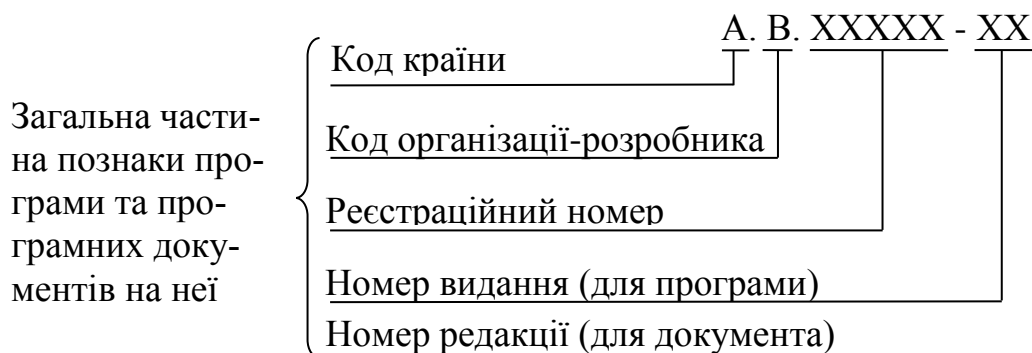
**Таблиця 2.2** — Види експлуатаційних документів

<b>Вид експлуатаційного документа</b>	<b>Код виду документа</b>	<b>Зміст експлуатаційного документа</b>
Розпис експлуатаційних документів	20	Перелік експлуатаційних документів на програму
Формуляр	30	Основні характеристики програми, комплектність та відомості про експлуатацію програми
Опис застосування	31	Відомості про призначеність програми, області застосування, методи, які застосовуються, клас розв'язуваних задач, обмеженнях щодо застосування, мінімальну конфігурацію технічних засобів
Настанова системного програміста	32	Відомості щодо перевірки, забезпечення функціонування та налагодження програми на умови конкретного застосування

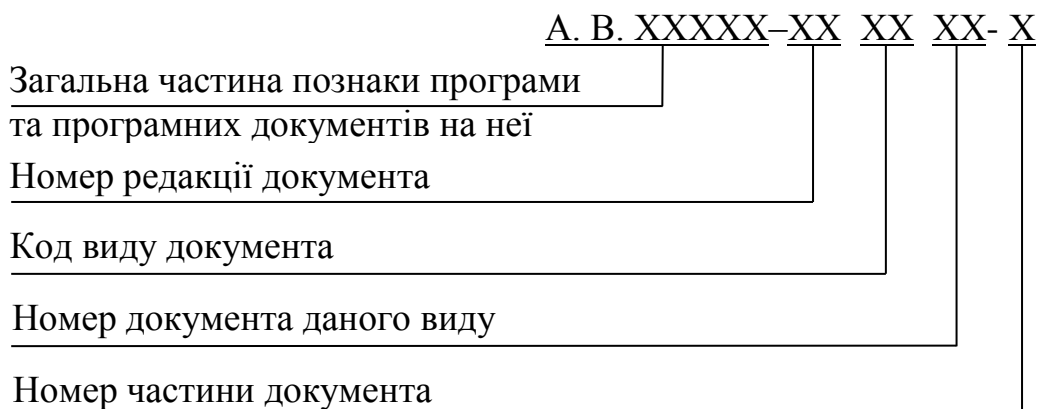
Настанова програ- міста	33	Відомості щодо експлуатації програми
Настанова операто- ра	34	Відомості щодо забезпечення процеду- ри спілкування оператора з обчислюва- льною системою в процесі виконання програми
Опис мови	35	Опис синтаксису та семантики мови
Настанова щодо те- хнічного обслугову- вання	46	Відомості щодо застосування тестових та діагностичних програм при обслуго- вуванні технічних засобів

Правила позначання програм і програмних документів встановлює ГОСТ 19.103-77. Згідно з його положеннями позначка програм і документів має складатися із груп знаків, відокремлених крапками (після коду країни та коду організації-розробника), пробілами (після номера редакції документа і коду виду документа), дефісами (після реєстраційного номера та номера документа даного виду).

Нижче подано структуру позначки програми та її програмного документа — специфікації.



Структура позначки інших документів:



Код країни-розробника та код організації-розробника присвоюють в установленому порядку. Реєстраційний номер присвоюють відповідно до класифікатора програм. Допускається присвоювати реєстраційний номер у порядку зростання, починаючи з 00001 до 99999, для кожної організації (підприємства)-розробника. Номер видання програми або номер редакції документа, а також номер документа даного виду присвоюють у зростаючому порядку від 01 до 99. Номер частини одного і того ж документа присвоюють в порядку зростання від 1 до 9. Якщо документ складається з однієї частини, дефіс і номер частини не проставляють. Зазначимо, що номер редакції специфікації та розпису експлуатаційних документів на програму має збігатися з номером видання тієї самої програми.

## **3 Правила виконання схем алгоритмів, програм, даних і систем**

### **3.1 Основні поняття**

Правила виконання схем алгоритмів, програм, даних і систем встановлює ГОСТ 19.701 90. Згідно з його положеннями *схема* — це графічне подання визначення, аналізу або методу розв'язання задачі, в якому для відображення операцій, даних, потоку, обладнання тощо використовують символи. Зміст інформації, що її відображає схема, визначається видом схеми. Так, *схеми даних* відображають шлях даних при розв'язанні задач і визначають етапи оброблення, а також носії даних. *Схеми програм* відображають послідовність операцій у програмі, а *схеми роботи системи* — керування операціями та потік даних у системі. *Схеми взаємодії програм* відображають шлях активації програм і взаємодій з відповідними даними, *схеми ресурсів системи* — конфігурацію блоків даних та блоків оброблення, яка необхідна для розв'язання задачі чи набору задач.

У загальному випадку схеми алгоритмів, програм, даних і систем складаються із символів, які мають певне значення, короткого пояснювального тексту та з'єднуючих ліній. Схеми можуть використовуватися на різних рівнях деталізації, причому число рівнів залежить від розміру та складності задачі оброблення даних.

Символи, що їх використовують у схемах, поділяють на *основні* та *специфічні*. *Основний символ* — це символ, що його застосовують у випадку, коли точний вид процесу чи носія даних невідомий або відсутня потреба у його описі. *Специфічний символ* — це символ, який застосовують у тих випадках, коли вид процесу чи носія даних точно відомий або ж його необхідно зазначити.

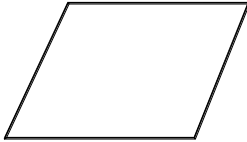


## 3.2 Опис символів, що їх використовують у схемах алгоритмів, програм, даних і систем

### 3.2.1 Символи даних

#### 3.2.1.1 Основні символи даних

##### *Дані*



Символ відображує дані, носій яких не визначений.

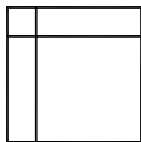
##### *Дані, які запам'ятовуються*



Символ відображує дані, що зберігаються у вигляді, придатному для оброблення. Носій даних не визначений.

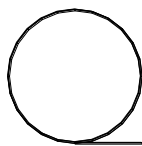
#### 3.2.1.2 Специфічні символи даних

##### *Оперативний запам'ятовуючий пристрій*



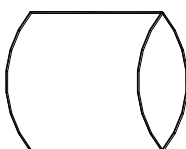
Символ відображує дані, що зберігаються в оперативному запам'ятовуючому пристрої.

##### *Запам'ятовуючий пристрій з послідовним доступом*



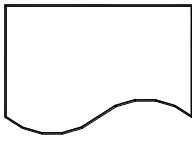
Символ відображує дані, що зберігаються у запам'ятовуючому пристрої з послідовним доступом (магнітна стрічка, касета з магнітною стрічкою, магнітофонна касета).

##### *Запам'ятовуючий пристрій з прямим доступом*



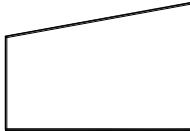
Символ відображує дані, що зберігаються в запам'ятовуючому пристрої з прямим доступом (магнітний диск, магнітний барабан, гнучкий магнітний диск).

### *Документ*



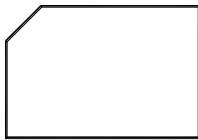
Символ відображує дані, що подані на носії у формі, придатній для читання (машинограма, документ для оптичного чи магнітного зчитування, мікрофільм, рулон стрічки, бланк).

### *Ручне введення*



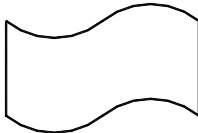
Символ відображує дані, що вводяться вручну під час оброблення з пристрою будь-якого типу (клавіатура, перемикачі, кнопки, світлове перо, смужки зі штриховим кодом).

### *Карта*



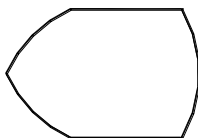
Символ відображує дані, що подані на носії у вигляді карти (перфокарти, магнітні картки, картки з мітками, що скануються або зчитуються, картки з відривним ярликом).

### *Паперова стрічка*



Символ відображує дані, що подані на носії у вигляді паперової стрічки.

### *Дисплей*



Символ відображує дані, подані у формі, придатній для читання людиною, на носії у вигляді пристрою відображення (екран для візуального спостереження, індикатори введення інформації).

## 3.2.2 Символи процесу

### 3.2.2.1 Основні символи процесу

#### *Процес*



Символ відображує функцію оброблення даних будь-якого виду (виконання певної операції чи групи операцій, що призводить до зміни значень, форм чи розміщення інформації або до визначення, яким з кількох напрямків потоку слід рухатись).

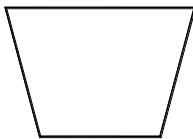
### 3.2.2.2 Специфічні символи процесу

#### *Наперед визначений процес*



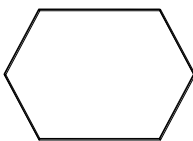
Символ відображує наперед визначений процес, що складається з однієї або кількох операцій чи кроків програми, які визначені в іншому місці (підпрограмі, модулі).

#### *Ручна операція*



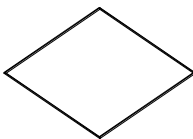
Символ відображує будь-який процес, що виконується людиною.

#### *Підготовка*



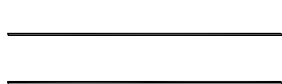
Символ відображує модифікацію команди чи групи команд з метою впливу на деяку наступну функцію (встановлення перемикача, модифікація індексного регістра чи ініціалізація програми).

#### *Рішення*



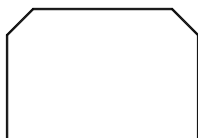
Символ відображує рішення або функцію перемикаючого типу, що має один вхід і ряд альтернативних виходів, один і тільки один з яких може бути активізований після обчислення умов, зазначених всередині цього символу. Відповідні результати обчислення можуть бути записані поряд з лініями, що відображають альтернативні шляхи.

## *Паралельні дії*



Символ відображує синхронізацію двох або більше паралельних операцій.

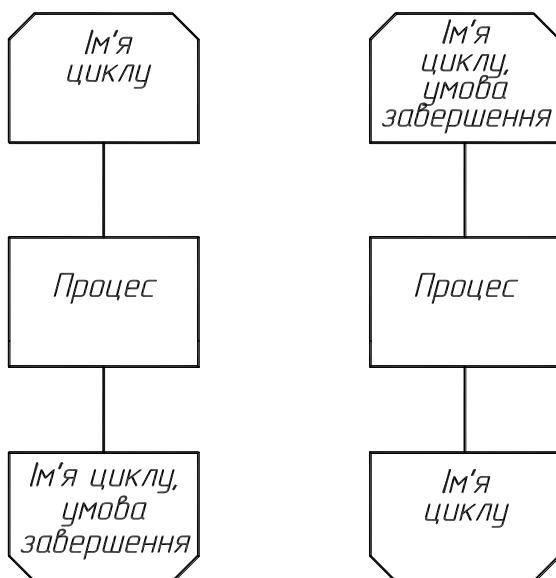
## *Межі циклу*



Символ, що складається з двох частин, відображує початок і кінець циклу. Обидві частини символу мають один і той же ідентифікатор. Умови для ініціалізації, приросту, завершення тощо розташовують всередині символу на початку або в кінці, залежно від розташування операції, що перевіряє умову.



## *Приклад*



### 3.2.3 Символи ліній

#### 3.2.3.1. Основний символ ліній

##### *Лінія*



Символ відображує потік даних або керування. У разі необхідності можуть бути додані стрілки-вказівники.

#### 3.2.3.2. Специфічні символи ліній

##### *Передача керування*



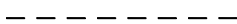
Символ відображує безпосередню передачу керування від одного процесу до іншого. Тип передачі має бути названий всередині символу (наприклад, запит, виклик, подія).

##### *Канал зв'язку*



Символ відображає передачу даних по каналу зв'язку.

##### *Пунктирна лінія*



Символ відображує альтернативний зв'язок між двома або більше символами. Окрім того, символ використовують для обведення ділянки, яку анують.

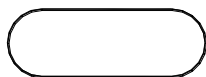
### 3.2.4 Спеціальні символи

##### *З'єднувач*



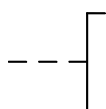
Символ відображує вихід у частину схеми та вхід з іншої частини цієї схеми і застосовується для обриву лінії та продовження її в іншому місці. Відповідні символи-з'єднувачі мають містити одну й ту саму унікальну позначку.

### ***Термінатор***



Символ відображує вихід у зовнішнє середовище і вхід із зовнішнього середовища (початок або кінець програми, зовнішнє використання та джерело або пункт призначення даних).

### ***Коментар***



Символ використовують для додання описових коментарів та пояснювальних записів. Пунктирні лінії в символі коментаря пов'язані з відповідним символом або можуть обводити групу символів. Текст коментаря повинен розташовуватись поряд з обмежуючою фігурою.

### ***Пропуск***



Символ (три крапки) використовують для відображення пропуску символу чи групи символів. Символ використовують тільки в символах ліній або між ними.

## **3.3 Правила виконання схем**

3.3.1 Схеми виконують на аркушах, що мають формати відповідно до ГОСТ 2.301-68.

3.3.2 Символи у схемі мають бути розміщені рівномірно. Слід дотримуватись розумної довжини з'єднань та мінімальної кількості довгих ліній.

3.3.3 Форми символів мають відповідати ГОСТ 19.701-90. Не дозволяється змінювати кути та інші параметри, що впливають на форму символів. Символи повинні бути, за можливістю, одного розміру.

3.3.4 Символи можуть бути накреслені у будь-якій орієнтації, але перевагу слід надавати горизонтальній орієнтації. Дзеркальне зображення символу означає ту саму функцію, проте йому не слід надавати перевагу.

3.3.5 Мінімальну кількість тексту, необхідну для розуміння функції даного символу, слід розміщувати всередині символу. Текст потрібно записувати зліва направо та зверху вниз незалежно від напрямку потоку. Якщо обсяг тексту перевищує розміри символу, слід використовувати символ коментаря.

3.3.6 У схемах дозволяється застосовувати ідентифікатор символу, який визначає символ для використання у довідкових цілях в інших елементах документації. Ідентифікатор має бути розміщений зліва над символом.

3.3.7 Справа над символом можна розміщувати опис символу — будь-яку іншу інформацію, наприклад, для поліпшення розуміння функції як частини схеми.

3.3.8 У схемах можна застосовувати детальне подання, яке позначають за допомогою символу зі смугою. Символ зі смугою являє собою будь-який символ процесу чи даних, усередині якого у верхній частині проведена горизонтальна лінія, над якою розміщується ідентифікатор. Цей символ показує, що в цьому самому комплекті документації в іншому місці є детальне подання. Першим та останнім символами детального подання мають бути термінатори, у яких розміщують ідентифікатори, аналогічні до ідентифікатора у символі зі смугою.

3.3.9 Потоки даних та потоки керування у схемах показують лініями. Стандартним вважається напрямок зліва направо та зверху вниз. Якщо потік має напрямок, відмінний від стандартного, цей напрямок показують стрілками.

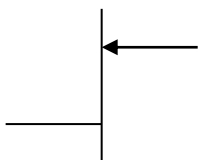
3.3.10 Слід уникати перетину ліній. Лінії, які перетинаються, не мають логічного зв'язку, тому зміна напрямку в точках перетину не допускається.

*Приклад*



3.3.11 Дві чи більше вхідних ліній можуть об'єднуватися в одну вихідну лінію; при цьому місце об'єднання має бути зміщене.

*Приклад*

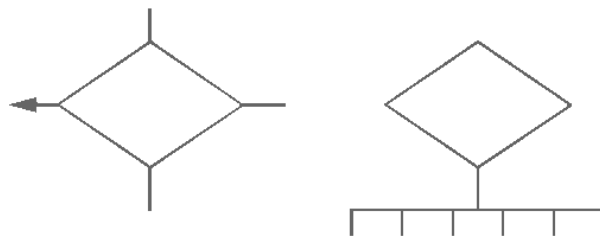


3.3.12 Лінії у схемах повинні підходити до символу або зліва, або зверху, а відходити від нього справа або знизу. Лінії мають бути спрямовані до центру символу.

3.3.13 У разі необхідності лінії у схемах слід розривати (щоб уникнути зайвих перетинів або занадто довгих ліній, а також у випадку, коли схема складається з кількох аркушів). На початку та в кінці розриву ставлять з'єднувачі.

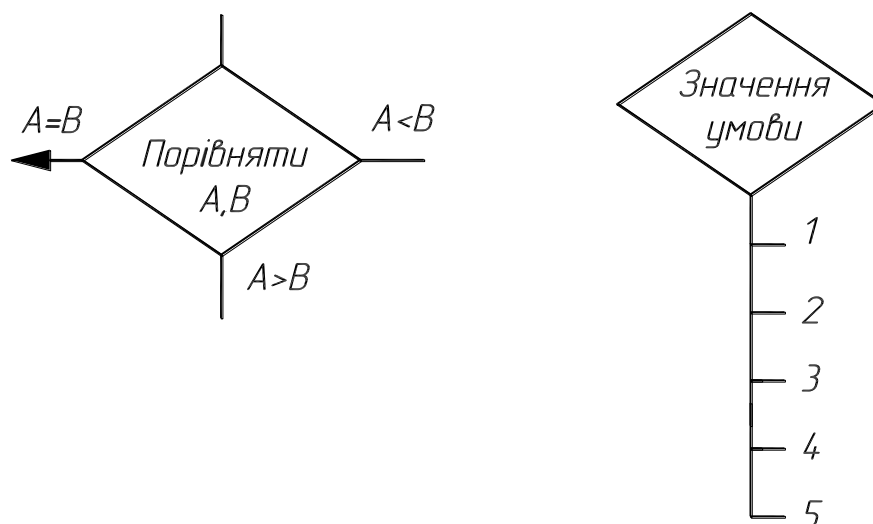
3.3.14 Декілька виходів із символу слід показувати або кількома лініями від даного символу до інших, або однією лінією від даного символу, яка потім розгалужується на відповідну кількість ліній.

#### Приклади



3.3.15 Кожний вихід із символу має супроводжуватися відповідними значеннями умов, щоб вказати логічний шлях, який він відображує.

#### Приклади





## 4 Зміст графічної роботи та послідовність її виконання

Мета графічної роботи — ознайомлення студентів на практиці з умовними позначками (символами), що їх використовують у схемах алгоритмів, програм, даних і систем, та правилами графічного оформлення цих схем відповідно до чинних стандартів. Студент повинен:

- Вивчити теоретичний матеріал, поданий у розділах 1...3 даних методичних вказівок.
- Проаналізувати умову обчислювальної задачі згідно зі своїм варіантом (додаток А) і скласти алгоритм її розв'язання. До окремих задач, які можуть становити для студента певні труднощі, у додатку Б подано рекомендації щодо їх розв'язання.
- Накреслити схему алгоритму на аркуші формату А3 та оформити кресленик відповідно до вимог чинних стандартів.

Як приклад розглянемо наступну задачу.

*Обчислити суму членів нескінченного ряду*

$$z = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots = 1 + \sum_{n=1}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}$$

*з точністю до члена ряду, меншого  $\varepsilon$ , за умови, що вхідні дані вводяться з клавіатури, а результат виводиться на друк.*

Очевидно, що для розв'язання цієї задачі можна застосувати не один алгоритм. Зі всіх можливих алгоритмів, які забезпечують отримання результату, слід вибрати найкращий відповідно до деякого критерію. Найчастіше як критерій використовують або точність розв'язку задачі, або витрати часу на її розв'язання, або деякий інтегральний критерій, що враховує і точність і витрати часу. У даному випадку критерієм ефективності алгоритму буде витрата часу на розв'язання задачі. З метою зменшення витрат часу на обчислення значення поточного члена ряду слід застосувати рекурентну формулу

$$y_n = y_{n-1} \frac{-x^2}{2n(2n-1)}$$

Після того, як вибрано формулу, можна визначити послідовність операцій в обчислювальному процесі, тобто скласти алгоритм. Для даної задачі ця послідовність буде такою:

- Введення даних  $x$ ,  $\varepsilon$ .
- Присвоєння початкових значень величинам  $n$ ,  $y$ ,  $z$ , а саме:  $n = 1$ ,  $y = 1$ ,  $z = 1$ .

- Обчислення поточного значення  $y_n$  члена ряду.
- Обчислення поточного значення  $z_n$  суми.
- Збільшення на одиницю значення  $n$  лічильника членів ряду.
- Перевірка, чи перевищує обчислений поточний член ряду значення  $\varepsilon$ . У випадку, коли перевищує, — перехід до обчислення наступних значень члена ряду та суми, коли ж ні — закінчення обчислень та вивід результату.

Розроблений алгоритм зображуємо у вигляді схеми. Познаку схеми будуємо наступним чином.

Код країни не проставляємо. Код організації-розробника замінюємо шифром групи (наприклад, ОТ). Замість реєстраційного номеру проставляємо номер варіанту, доповнивши його спереду нулями щоб отримати необхідні п'ять цифр (наприклад, для одинадцятого варіанту маємо 00011). Код виду документа визначаємо відповідно до ГОСТ 19.101-77 (див. табл. 2.1). Для пояснювальної записки, до якої належить схема алгоритму, він становить 81.

Приклад виконання графічної роботи наведено у додатку В.

## Додаток А

### Варіанти завдань до графічної роботи «Схема алгоритму»

1. Скласти схему алгоритму обчислення величини

$$R = \sum_{r=2}^n \sum_{i=1}^m (a_i + b_i)^r,$$

за умови, що  $n, m$  вводяться з клавіатури на початку обчислень,  $a_i, b_i$  — з магнітної стрічки на кожному кроці обчислень, результат виводиться на принтер.

2. Скласти схему обчислення величин

$$M = \frac{1}{N} \sum_{i=1}^N X_i, \quad D = \frac{1}{N-1} \sum_{i=1}^N (X_i - M)^2,$$

за умови, що  $N = 40$ , послідовність  $X_i$  вводиться на початку обчислень (пристрій вводу не вказано), результат виводиться на дисплей.

3. Скласти схему обчислення величини

$$S = \sum_{i=1}^N (a_i^2 + a_{i+1}^2),$$

за умови, що  $n = 100$ , послідовність  $a_i$  вводиться з магнітного диска на початку обчислень, результат виводиться на принтер.

4. Скласти схему обчислення величини

$$S = \sum_{k=1}^N (1 + a_k + b_k),$$

за умови, що величина  $N$  вводиться з клавіатури, послідовності  $a_k, b_k$  вводяться у процесі обчислень з магнітного диска, результат виводиться на друк.

5. Скласти схему знаходження мінімального та максимального елементів послідовності  $X_i$  ( $i = 1, 2, 3, \dots, N$ ) і обчислення

$$C = \frac{X_{\min} + X_{\max}}{2},$$

за умови, що значення  $N$  і послідовність  $X_i$  вводяться з клавіатури на початку обчислень, результат виводиться на дисплей.

6. Скласти схему перетворення послідовності чисел  $X_1, X_2, \dots, X_N$  так, щоб у мінімальних за абсолютною величиною елементів послідовності знак змінився на протилежний. Вхідні дані ( $N$  — кількість елементів послідовності,  $X_i$  — елементи послідовності) ввести на початку обчислень (пристрій вводу не вказано), а перетворену послідовність вивести на магнітний диск.

7. Скласти схему алгоритму знаходження суми квадратів окремо додатних і від'ємних чисел заданої послідовності  $X_1, X_2, \dots, X_N$  за умови, що величина  $N$  вводиться з клавіатури, а елементи послідовності  $X_i$  — з магнітного диска на кожному кроці обчислень. Результат виводиться на друк.

8. Скласти схему алгоритму знаходження суми першого та мінімального чисел послідовності  $X_1, X_2, \dots, X_N$  за умови, що величина  $N$  вводиться з клавіатури, а елементи послідовності  $X_i$  — з магнітної стрічки на початку обчислень. Результат виводиться на дисплей.

9. Скласти схему алгоритму знаходження максимального від'ємного числа послідовності  $X_1, X_2, \dots, X_N$  за умови, що величина  $N$  і послідовність  $X_1, X_2, \dots, X_N$  вводяться з магнітного диска на початку обчислень. Результат виводиться на друк.

10. Скласти схему обчислень найбільшого додатного цілого числа, що задовольняє умові  $-n^n + 3n^2 - 1 > 0$ . Результат виводиться на друк.

11. Скласти схему алгоритму формування послідовності  $p_1, p_2, \dots, p_N$ , де

$$p_k = e^{-a} \frac{a^k}{k!}, \quad k=1, 2, \dots, N,$$

за умови, що пристрій вводу для вхідних даних не вказаний, а результат виводиться на магнітний диск наприкінці обчислення.

12. Скласти схему алгоритму формування послідовності чисел  $R_1, R_2, \dots, R_N$ , де

$$R_k = p^k \left( -p \right)^{N-k}, \quad k=1, 2, \dots, N,$$

за умови, що вхідні дані  $N, p$  вводяться з клавіатури, а результат обчислень виводиться на магнітний диск після обчислення чергового елемента послідовності.

13. Скласти схему алгоритму формування послідовності  $B_1, B_2, \dots, B_N$ , де

$$B_k = \frac{k}{k+2}, \quad k=1, 2, \dots, N,$$

за умови, що  $N$  вводиться з клавіатури. Результати обчислень виводяться на друк після закінчення обчислювального процесу.

14. Скласти схему алгоритму формування послідовності  $B_1, B_2, \dots, B_N$ , де

$$B_k = \begin{cases} A_k + \frac{X}{2}, & \text{якщо } A_k < 0 \\ A_k - X, & \text{якщо } A_k > 0 \\ 2X, & \text{якщо } A_k = 0, \end{cases}$$

за умови, що константи  $X, N$  вводяться з клавіатури, елементи послідовності  $A_k$  – на кожному кроці обчислень з магнітної стрічки, послідовність  $B_k$  виводиться на магнітну стрічку в кінці обчислень.

15. Скласти схему алгоритму формування послідовності чисел

$$V_i = \frac{X_i^n}{i},$$

де  $i = 1, 2, \dots, R$ , за умови, що величини  $R, n$  і послідовність  $X_i$  вводяться на початку обчислень (пристрій вводу не вказано), елементи послідовності  $V_i$  виводяться на кожному кроці обчислень на друк.

16. Скласти схему алгоритму формування послідовності  $p_1, p_2, \dots, p_N$ , де

$$p_k = \left(1 - \frac{a^k}{k!}\right), \quad k=1, 2, \dots, N,$$

за умови, що пристрій вводу для величин  $a, N$  не вказаний, а елементи послідовності виводяться на друк відразу після їх обчислення.

17. Скласти схему алгоритму визначення кількості всіх додатних елементів заданої послідовності  $X_1, X_2, \dots, X_N$  за умови, що величина  $N$  вводиться з клавіатури, а елементи послідовності  $X_i$  – з магнітної стрічки на початку обчислень, результат виводиться на друк.

18. Скласти схему алгоритму визначення кількості елементів послідовності  $X_1, X_2, \dots, X_N$ , які не перевищують першого числа цієї послідовності, за умови, що всі вхідні дані вводяться на початку обчислень (пристрій вводу не вказано), а результат виводиться на друк.

19. Скласти схему алгоритму обчислення величини

$$T = \sum_{j=1}^m \sum_{i=1}^n (x_i^j - y_i),$$

за умови, що всі вхідні дані вводяться на початку обчислень з магнітної стрічки, а результат виводиться на дисплей.

20. Скласти схему алгоритму обчислення величини

$$M = \prod_{i=1}^{h-1} (C_i - C_{i+1}),$$

за умови, що  $h$  вводиться з клавіатури, а послідовність  $C_i$  — з магнітної стрічки на початку обчислень. Результат виводиться на друк.

21. Скласти схему алгоритму обчислення величини

$$V = \sum_{k=1}^N k! a_k b_k,$$

за умови, що величина  $N$  та послідовності  $a_k, b_k$  вводяться на початку обчислень, а результат виводиться наприкінці обчислень (пристрої вводу та виводу не вказані).

22. Скласти схему алгоритму формування послідовності чисел

$$R_k = \frac{a^k}{k!}, \quad k=1, 2, \dots, N,$$

за умови, що вхідні дані  $a, N$  вводяться з клавіатури, а елементи послідовності  $R_k$  виводяться на друк на кожному кроці обчислень.

23. Скласти схему алгоритму формування послідовності

$$X_k = \begin{cases} y_k + a^2, & \text{якщо } y_k < 0, \\ y_k + a^3 + a^2, & \text{якщо } y_k > 0, \\ a^4 + a^3 + a^2, & \text{якщо } y_k = 0, \end{cases}$$

за умови, що вхідні дані вводяться на початку обчислень (пристрій вводу не вказаний), а елементи послідовності  $X_k$  виводяться на друк на кожному кроці обчислень.

24. Скласти схему алгоритму формування послідовності

$$C_i = -B_i + A_i^n,$$

де  $i = 1, 2, \dots, R$ , за умови, що величини  $R, n$  і послідовність  $B_i$  вводяться перед початком обчислень з клавіатури, а елементи послідовності  $A_i$  вводяться з магнітного диска на кожному кроці обчислень, послідовність  $C_i$  виводиться на друк після закінчення обчислень.

25. Скласти схему визначення суми квадратів усіх від'ємних чисел послідовності  $A_1, A_2, \dots, A_N$  за умов, що величина  $N$  і послідовність  $A_k$

вводяться на початку обчислень з магнітного диска, а результат виводиться на друк.

26. Скласти схему алгоритму визначення найменшого числа послідовності  $X_1, X_2, \dots, X_N$ . Вхідні дані (число  $N$  і послідовність  $X_1, X_2, \dots, X_N$ ) вводяться на початку обчислень з магнітної стрічки, а результат виводиться на друк.

27. Скласти схему алгоритму знаходження елементів послідовності  $X_1, X_2, \dots, X_N$ , значення яких задовольняють умові  $0 < x_i < 1$ . Вхідні дані вводяться з магнітної стрічки, а значення елементів послідовності та їх номери виводяться на друк в процесі обчислень.

28. Скласти схему алгоритму обчислення функції

$$z = \sum_{i=1}^N \frac{X_i^2}{i}, \quad \text{де } X_i - \text{елемент масиву } (X_1, X_2, \dots, X_N).$$

Вхідні дані вводяться з магнітного диску, а результат виводиться на дисплей.

29. Скласти схему алгоритму обчислення добутку додатних елементів масиву  $(Y_1, Y_2, \dots, Y_N)$ , за умови, що носій вхідних даних не визначений, а результат виводиться на друк.

30. Скласти схему алгоритму знаходження найменшого з додатних елементів масиву  $(X_1, X_2, \dots, X_N)$ , за умови, що вхідні дані вводяться з магнітної стрічки, а результат виводиться на дисплей.

## Додаток Б

Вказівки до складання алгоритмів розв'язання окремих задач.

1. Слід змінити порядок обчислення сум, щоб під час введення чергових значень  $a_i$  та  $b_i$  обчислювати ступені їх сум:

$$R = \sum_{i=1}^m \sum_{r=2}^n (a_i + b_i)^r.$$

2. Звернути увагу на те, що величини  $M$ ,  $D$  треба обчислювати послідовно.
3. Перетворити формулу наступним чином:

$$S = a_1 + a_n + 2 \sum_{i=2}^{n-1} a_i^2.$$

11. Для формування послідовності застосувати рекурентну формулу:

$$P_k = P_{k-1} \frac{a}{k}, \quad k = 2, 3, \dots, N,$$
$$P_1 = e^{-a}.$$

12. Для формування послідовності застосувати рекурентну формулу:

$$R_k = R_{k-1} \frac{p}{1-p}, \quad k = 2, 3, \dots, N,$$
$$R_1 = p \left( -p \right)^{N-1}.$$

19. Змінити порядок обчислення сум і, запам'ятавши поточне значення  $X_i^j$ , використати його для обчислення  $X_i^{j+1}$ :

$$a_{i0} = 1, \quad a_{ij} = a_{ij-1} x_i, \quad T = \sum_{i=1}^n \sum_{j=1}^m (a_{ij} - y_i).$$

21. Ввести змінну для запам'ятовування поточного значення  $k!$  і використати її для обчислення наступного значення:

$$p = 1,$$
$$p = pk, \quad k = 1, 2, 3, \dots, N.$$

22. Для формування послідовності використати рекурентну формулу:

$$R_1 = a, \quad R_k = R_{k-1} \frac{a}{k}, \quad k = 2, 3, \dots, N.$$

23. Спочатку обчислити значення констант:

$$A = a^2, \quad B = A \left( + a \right), \quad C = B + a \left( a \right).$$





## Література

1. Інженерна графіка: [підруч. для студентів вищих навч. закл.] / В. В. Ванін, В. В. Перевертун, Т. М. Надкернична, Г. Г. Власюк. – К. : Видавнича група ВНУ, 2006. – 400с.
2. ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных и систем.
3. ГОСТ 19.103-77. ЕСПД. Обозначение программ и программных документов.