

Тема 4. ЕЛЕМЕНТИ ПРОГРАМУВАННЯ В MAPLE. ТИПИ ДАНИХ. ВИРАЗИ, ОПЕРАНДИ ТА ОПЕРАЦІЇ

Елементи мови *Maple*

Вихідний файл - текстовий файл, який включає частину або ж увесь програмний код.

Програма може складатися як з одного файла, так із декількох сотень файлів вихідного тексту.

Функції, змінні та константи є головними об'єктами, які складають вихідну програму (вихідні файли).

Елементи мови Maple - базові конструкції, які використовуються при написанні програм. До них відносяться: алфавіт, ідентифікатори, ключові слова, коментар і т.п.

Алфавіт - множина символів, які можна набрати на клавіатурі для створення конструкцій мови *Maple*.

Алфавіт використовується для формування **лексем** - мінімальних лексичних одиниць, які розпізнаються інтерпретатором. До них відносяться **ключові** слова (зарезервовані слова), символи операцій, імена змінних, функцій і команд, рядки, числа, роздільники.

Інтерпретатор *Maple* зчитує кожен рядок та розбиває його на лексеми, які відразу ж виконує.

Синтаксис – сукупність правил запису *Maple* об'єктів.

Український алфавіт може бути використаний **тільки** в символьних рядках, константах чи в **коментарях** (в поясненнях до програми).

Символи проміжку - клавіша проміжку, перехід на новий рядок, горизонтальна і вертикальна табуляції, нова сторінка. Ці символи розділяють лексеми одну від іншої.

Інтерпретатор ігнорує декілька підряд символи проміжку, що дозволяє їх використовувати для покращання наочності програми.

Роздільники використовуються інтерпретатором для організації тексту програми, вказівок компілятору виконувати конкретні дії:

- , кома (*comma*) - при перерахуванні;
- ; крапка з комою (*semicolon*) - закінчення оператора;
- [] квадратні дужки (*square brackets*) - означення масиву;
- { } фігурні дужки (*curly brackets*) - означення множини;
- () круглі дужки (*parenthesis*) - означення функції та інші.

Константи

Константи - число, символ або ж рядок символів, які не змінюється в процесі виконання програми.

Розрізняють константи: цілі; дійсні (з плаваючою крапкою); символні; строкові.

Maple містить цілий ряд **іменованих** констант - таких, до значень яких можна звертатися по імені.

Частина цих констант не може бути змінена. До них відносяться:

- *true, false* – логічні значення "істинно" і "неправдиве";
- *infinity* - нескінченність;

- I - уявна одиниця;
- Pi - константа 3.14159265.

Деякі константи можуть бути **перевизначені**, наприклад, *Digits* - задає число значущих цифр для чисел з **плаваючою** точкою (за замовчуванням 10).

Константи в *Maple*: *?Ininame*.

Типи чисел

Maple працює з наступними **типами чисел**:

- (*integer*) цілими десятковими (0, 1, 123, -456 і т.д.),
- (*fraction*) раціональними у вигляді відношення цілих чисел (7/9, -123/127 і т.д.),
- (*float*) дійсними з мантиси і порядком (1.23E5, 123.456E-10)
- (*complex*) комплексними (2+3I).

Натуральні числа - будь-яка послідовність цифр. *Maple* ігнорує зліва нулі.

Цілі числа - натуральні числа зі знаком + або –.

Цілі числа задаються у вигляді послідовності цифр від 0 до 9. Приклад цілих констант: 10, 32767, -32767.

Maple може працювати з цілими числами довільної величини, кількість цифр обмежено числом 2^{28} . Великі числа, які не поміщаються на одному рядку, *Maple* переносить на наступний рядок, використовуючи символ зворотного слеша (\).

При перевірці типу натурального числа команда *whattype()* повертає цілий тип *integer*.

Для уточнення підтипу цілого числа необхідно використовувати команду *type()* з параметром, що описує підтипи цілого типу: *natural* (натуральний), *negint* (негативне ціле), *posint* (позитивне ціле), *even* (парне), *odd* (непарне) і *prime* (просте).

Чотири арифметичні дії над цілими є додавання +, віднімання –, множення *, ділення / і обчислення факторіала (!).

Отримати список всіх команд для роботи з цілими числами можна, набравши команду: *?Integer*. Деякі з цих команд:

- *irem* - обчислення залишку операції цілого ділення $irem(53,7)=4$
- *igcd* - знаходження найбільшого загального дільника цілих чисел $igcd(192,90)=6$
- *isprime* - перевірка, чи є ціле число простим.

Звичайні дроби - операції ділення двох цілих чисел.

Maple автоматично виконує скорочення дробів, наприклад $32/12$;

$$\frac{8}{3}$$

Над дробами можна виконувати всі основні арифметичні операції.

Для перетворення звичайного дробу в десяткову величину служить команда *evalf()*. Другий параметр цієї команди задає **число значущих цифр**, наприклад: $evalf(3/8, 4) = 0.3750$.

Дріб і її десяткове подання не є ідентичними об'єктами *Maple*.

Числа з плаваючою крапкою - дійсне число, яке має цілу та дробову її частини.

Ціла частина від дробової відділяється крапкою, а не комою.

Для запису дійсної константи в експоненціальній формі використовують символ 'e' або ж 'E', наприклад: 15.78, 0.5, 1.0, 1.2e7, .00075, -2.5e-2. Обчислення дійсних чисел проводиться за замовчуванням з 10 значущими числами.

Якщо всі числа в виразі є цілими, дробами або радикалами, то результат представляється також з використанням цих типів даних.

Якщо ж у виразі присутнє число з плаваючою точкою, то результатом виконання такого "змішаного" виразу буде число з плаваючою точкою.

Приклад. $3e2+3/4+sqrt(5)+3/4*0.1+surd(6,3)*43/10;$

$$300.8250000 + \sqrt{5} + \frac{43}{10} 6^{(1/3)}$$

Радикали задають як результат введення в дробовий степінь цілих або дробових чисел, або обчислення квадратного кореня $sqrt()$.

В *Maple* піднесення до степеню задається символом $^$ або послідовністю двох зірочок (**).

На відміну від чисел з **плаваючою** крапкою, обчислення з цілими, дробами і радикалами є абсолютно **точними**, оскільки *Maple* не виконує ніяких **заокруглень**.

Комплексні числа. Для уявної одиниці в *Maple* використовується константа *I*.

Спеціальні символи (або ж *ESC* послідовності) використовують для виконання характерних дій, наприклад, новий рядок, табуляції.

Рядок (*string*) - будь-який набір символів, взяті в подвійні лапки, наприклад “*This is one string*”.

Довжина рядка в *Maple* може мати 268435439 символів. З’єднання рядків можна виконати за допомогою операції конкатенації ||, або зверненням до функції *cat(рядок 1, рядок 2)*.

Коментар - пояснювальний текст до програми, який можна записати в будь-якій її частині.

Коментар, як послідовність символів, компілятором ігноруються та інтерпретуються як проміжки.

Рекомендується вживати коментар як найчастіше, що дозволить легше зрозуміти програмний код при його модифікації.

Ідентифікатори

Ідентифікатори - назви змінних і функцій.

В ідентифікаторі використовують тільки дві множини символів, які можна набрати на клавіатурі:

- літери латинського алфавіту - A...Z, a...z, знаку підкреслювання «_»;
- арабських цифри 0, 1, 2..9.

Вимоги до написання ідентифікаторів:

- повинен починатися з букви або ж знаку підкреслювання;
- повинен нести зрозумілий смисл назви змінної чи функції;
- не повинен співпадати з ключовим словом.

Приклади: *temp1*, *Skip_1*, *Alfa*, *Fi*, *Sum*, *Ad*, *aD*.

Maple чутлива до регістру - малі та великі літери **відрізняються**.

Ключові (службові, зарезервовані) слова - наперед визначені ідентифікатори, які мають спеціальне значення.

*and break by catch description do done elif else end error export fi finally
for from global if in intersect local minus mod module next not od option
options or proc quit read return save stop then to try union use while*

Службові слова **не можна** вживати **як власні** ідентифікатори.

Типи даних

Повідомлення A разом з відповідною інформацією B називають **величиною**. Інформацію B називають **значенням** величини, а повідомлення A - **позначенням** величини.

Тип об'єкта визначає множину значень, які цей об'єкт може приймати і операції, які над ними можуть виконуватися.

Цілі числа, дробові числа, числа з плаваючою точкою відносяться до **основних типів** даних. До більш складних типів даних відносять масиви, множини, списки.

Одно- та багатовимірні масиви створюються шляхом оголошення масивів із елементів простих типів даних. Елементи багатовимірного масиву запам'ятовуються по рядках. Для створення масиву використовується команда *array(i1..j1, i2..j2,..., M)*, яка повертає масив з елементами із списку *M*.

Множина – перелік елементів, обмежених фігурними дужками:

$$A:=\{a,b,c\};$$

Списки – перелік елементів, обмежених квадратними дужками:

$$B:=[a,b,c];$$

Вектори – перелік елементів, обмежених трикутними дужками:

$$C:=\langle a,b,c \rangle;$$

Звернення до елементів множини, списку, масиву здійснюється вказівкою індексу елемента в квадратних дужках.

Індексні імена - символічні імена з нижнім індексом.

Комп'ютерна змінна - поійменоване місцезнаходження частини пам'яті, де зберігаються дані певного типу.

По замовчуванню змінна є скалярною величиною.

Якщо у виразі використовується змінна, якої не присвоєно ніякого числового значення, то вона розглядається як невідома величина, а вираз, що містить невідомі, називається **символьним** виразом.

Змінні дозволяють зберігати й обробляти різноманітні типи даних. При цьому за замовчуванням змінна *Maple* має тип *symbol*, представляє символьну змінну, і її значенням є її власне ім'я.

При присвоєнні змінній якого-небудь значення, її тип змінюється на тип присвоєного їй значення.

Вирази, операції, операнди

В *Maple* розрізняють поняття **функції, команди і оператори**.

Команда *restart* очищає внутрішню пам'ять *Maple* від усіх визначень - відкриває нову *Maple*-сесію.

Вираз є комбінацією імен змінних, чисел і інших об'єктів, з'єднаних знаками допустимих операцій (додавання + , віднімання – , множення * , ділення / , присвоєння := , рівності = тощо).

Введення виразів задається по правилам строкового редактора.

Знак фіксації кінця виразу є точка з комою ; , що вказує на виведення результатів на монітор.

Знак дві крапки : відміняє виведення і може використовуватися для запису декількох виразів в одному рядку.

Операції - комбінації символів, які задають дії по перетворенню значень. **Операції** визначають дії над **операндами** - константами, змінними чи символьними рядками.

Операнди за допомогою операцій можуть комбінуватися в первинні **вирази**, які в свою чергу стають операндами. Їх комбінація приводить до утворення більш **складніших виразів** і т.д.

Існують три **типи виразів**:

- **математичні** - результатом виразу є число;

- **текстові** – результатом являється рядок символів;
- **логічні** - результат є твердження “так” чи “ні”.

В залежності від виду операцій вирази поділяються на **первинні, унарні, бінарні, тернарні, присвоєння та приведення типу** відповідно до кількості операторів у даному виразі.

Присвоєння це вираз.

Символи та їх назви

+ додавання (<i>addition</i>)	“ подвійні кавички (<i>double quotes</i>)
– віднімання (<i>subtraction</i>)	. десяткова точка (<i>decimal point</i>)
/ ділення (<i>division</i>)	.. діапазон (<i>range</i>)
* множення (<i>multiplication</i>)	< менше чим (<i>less than</i>)
^ піднесення до степеню (<i>exponentiation</i>)	<= менше або рівно (<i>less than or equal</i>)
:= присвоєння (<i>assignment</i>)	> більше чим (<i>more than</i>)

= рівність (<i>exclamation mark</i>)	>= більше або рівно (<i>greater or equal</i>)
! знак оклику (<i>equality</i>)	<> нерівно (<i>unequal</i>)
: дві крапки (<i>colon</i>)	^ кришка (<i>cap</i>)
% процент (<i>percent</i>)	# номер (<i>number</i>)
конкатенація (<i>concatenation</i>)	@ композиція (<i>composition</i>)
<i>union</i> об'єднання множини	:: оголошення типу
<i>minus</i> різниця множин	-> створення функції
<i>intersect</i> перетин множин	

Дія символів:

- : фіксатор виразу, який запобігає виводу результату;
- ; фіксатор виразу, який формує вивод результату в окремий рядок;
- # покажчик коментаря;
- := оператор присвоєння;
- ‘ обмежувачі змінної, при вживанні яких ігнорують попередні дані;
- % повертає останній результат обчислення.

Види унарних (з одним операндом) операцій

Знак	Назва операцій	Приклади
- +	Унарні мінус і плюс виконують арифметичне віднімання чи додавання свого операнда	$x = -2;$ $y = +3.5;$

Види бінарних (з двома операндами) операцій

Знаки	Назва операцій	Приклади
* /	Мультиплікативні - множення, ділення	$z = x * y;$ $z = x / y;$
+ -	Адитивні - додавання і віднімання	$z = x + y;$ $z = x - y;$
> < <= == <>	Відношення - більше, більше-дорівнює, менше, менше-дорівнює, дорівнює, не дорівнює	$x < y$
<i>mod</i>	Залишок від ділення цілих чисел	$7 \bmod 5$
\$	Оператор послідовності	$\$1..5$
!	Факторіал	$10!$

Пріоритет впливає на порядок обчислення операцій у виразах

Операції	Назви
$() [] ->$	первинні
$* /$	мультиплікативні
$+ -$	адитивні
$< <= > =$	відношення
$:=$	присвоєння

Приклад. Розробити блок-схему, словесний алгоритм і написати на мові *Maple* програму обчислення довжини кола по формулі $l=2\pi r$.

Блок-схема	Словесний опис	Вихідний текст на <i>Maple</i>
<pre> graph TD Start([Початок]) --> Input[/Ввод Rad/] Input --> Process[Обчислення L] Process --> Output[/Вивод L/] Output --> End([Кінець]) </pre>	Почати роботу Введемо значення <i>Rad</i> Обчислимо довжину кола $L=2\pi r$ Вивести значення Закінчити роботу.	<pre> restart; Радіус кола: Rad := 30; Довжина кола: L := 2·Rad·Pi; Результат: L; evalf(%) = 188.4955592 </pre>

Аналіз математичних виразів

Операції аналітичних перетворень - приведення подібних, розкладання на множники, розкриття дужок тощо.

Команди синтаксичного аналізу: *whattype()*, *type()*, *nops()* і *op()*.

Вибір лівої або ж правої частини математичного виразу $A=B$ здійснюється операторами *lhs(A=B)*; *rhs(A=B)*

Для отримання чисельника і знаменника використовують оператори *numer(вираз)*; *denom(вираз)*.

Для виділення частини виразу або списку слугує оператор *op(i, вираз)*, де *i* - число, що визначає позицію у виразі (списку).

Команда *simplify()* призначена спрощення різноманітних виразів.

Найбільш часто використовувані при перетвореннях команди і функції *Maple* розташовуються в його системному ядрі.