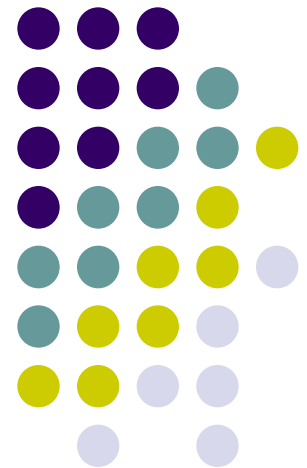


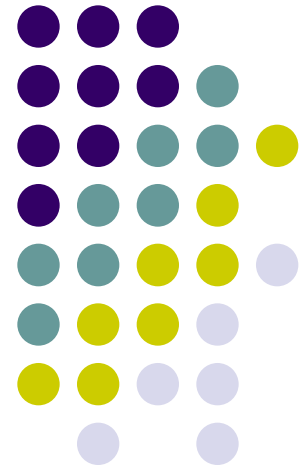
# Комп'ютери та комп'ютерні технології

Лектор: доцент Болбот Ігор Михайлович



**Спеціальний оператор  
припинення розрахунку *break*  
в пакеті *Mathcad***

---





# Оператор *break*

- В MathCAD є спеціальний оператор **break**, що дозволяє вийти із циклу або призупинити виконання програми при виконанні заданої в операторі **break** умови.

# Оператор *break*



- Для введення оператора **break** необхідно клацнути на кнопці **break** панелі **Програмування** (не можна вводити цей оператор із клавіатури по символах).



# Оператор *break*



- Оператор ***break*** використовується в лівому полі введення умовного оператор ***if***, а в правому розміщується умова, при виконанні якого відбувається припинення роботи циклу або програми, у нижнім полі - оператор, що виконується якщо умова не виконана.
- Тому спочатку вводиться оператор ***if***, а потім заповнюються поля цього оператора.



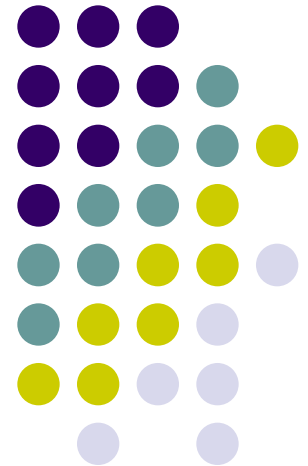
# Оператор *break*



```
break if █  
█
```

***Додатковий оператор  
програмування циклів  
continue в пакеті Mathcad***

---



# Оператор *continue*



- Оператор ***continue***, зазвичай використовується для продовження виконання циклу шляхом повернення в початок тіла циклу.





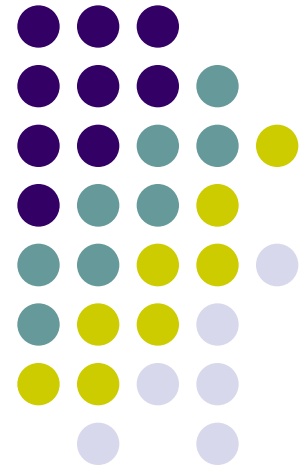
# Оператор *continue*



```
| continue if █  
| █
```

# *Додатковий оператор програмування циклів **return** в пакеті Mathcad*

---



# Оператор *return*



- Оператор ***return*** перериває виконання програми-функції та повертає значення операнда, що знаходиться за ним.



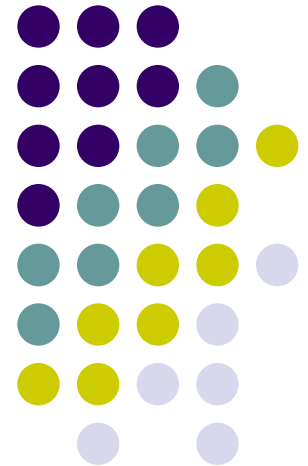
# Оператор *return*



```
| return ■ if ■  
| ■
```

**Додатковий оператор  
програмування циклів  
*on error* та функція *error* в  
пакеті *Mathcad***

---



# Оператор *on error*



- Оператор *on error*, цей оператор обробляє помилки, що виникають при виконанні тих або інших обчислень і записується у вигляді:
- $\langle \text{вираз 1} \rangle \text{ on error } \langle \text{вираз 2} \rangle$ .
- Виконується  $\langle \text{вираз 1} \rangle$ , якщо при виконанні  $\langle \text{виразу 2} \rangle$  виникає помилка. Якщо помилка не виникає, то виконується  $\langle \text{вираз 2} \rangle$ .



**Приклад:** Використаємо оператор *on error* для запобігання появи помилки "ділення на нуль" при обчисленні функції *angl(x,y)*.



$$\text{angl}(x, y) := \frac{x}{y}$$

$$\text{angl}(2, 0) =$$

Ділення на нуль

$$\text{angl}(x, y) := 0 \text{ on error } \text{angl}(x, y)$$

$$\text{angl}(2, 0) = 0$$

Результат при діленні на нуль

# Функція *error*



- Функція **error** використовується для висновку діагностичних повідомлень при виникненні в обчисленнях помилки і записується у вигляді:
- **error** ("*< діагностичне повідомлення користувача >*").
- Функція використовується в лівому полі умовного оператора **if**, як показано в наступному прикладі.





**Приклад: Програмування висновку  
діагностичного повідомлення при спробі  
спроєктувати вектор  $v$  на нульовий вектор  $w$ .**

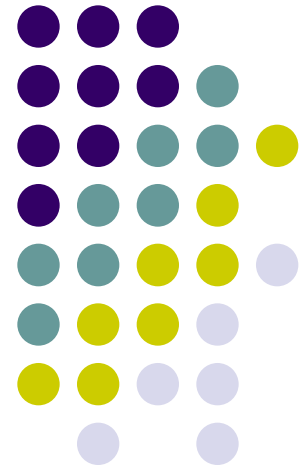


$$\text{proj}(v, w) := \begin{cases} \text{error}(\text{" You cannot project onto the 0 vector" }) & \text{if } |w| = 0 \\ \frac{w}{|w|} \cdot (v \cdot w) & \text{otherwise} \end{cases}$$



***Модульне програмування в  
межах одного документа  
Mathcad***

---



# Модульне програмування



- Загальна ідея модульного програмування полягає в наступному:
  - - реалізації обчислювальних процесів у вигляді окремих програмних одиниць - модулів;
  - - звертанні до цих модулів в інших програмах з передачею даних, необхідних для обчислювального процесу.



# Модульне програмування



- Модульне програмування в одному документі характеризується тим, що:
  - - для реалізації простих обчислень використовуються локальні функції, а більш складних - програми - функції;
  - - опис локальних функцій, програм-функцій та їхній виклик (тобто звернення до них) перебувають у межах одного документа та зберігаються в одному файлі. При цьому часто всередині однієї програми-функції перебувають виклики локальних функцій, вбудованих функцій MathCAD та іншої програми-функції.

# Модульне програмування

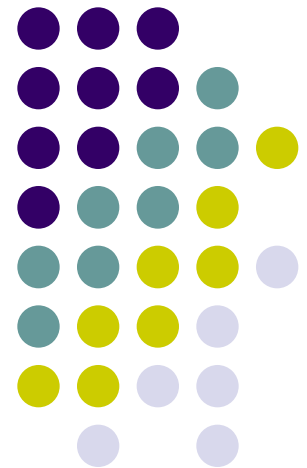


- *Модульне програмування дозволяє зменшити об'єм вихідних текстів програм, зробити їх більше простішими, прискорити написання та тестування програм, зменшити витрати на супровід (експлуатацію) програм.*



***Модульне програмування у  
декількох документах  
MathCAD***

---



# Модульне програмування



- *У попередньому способі реалізації модульного програмування опис модулів (функцій користувача та програм-функцій) і їхній виклик перебуває в одному документі. Такий спосіб має ряд недоліків.*

# Модульне програмування



- *Недолікі:*
  - - неможливість паралельної розробки програм декількома розроблювачами;
  - - неможливість "автономного" налагодження програм-функцій та їхньої модифікації в процесі експлуатації програмного забезпечення;
  - - неможливість використання розробленої програми-функції в декількох документах без дублювання опису програми-функції.



# Модульне програмування



- *Для подолання цих недоліків опис програми-функції виконують в одному документі MathCAD, а її виклик розміщується в іншому документі (цей прийом широко використовується в сучасних алгоритмічних мовах високого рівня).*

# Модульне програмування



- Однак при цьому виникає питання: як при виклику програми-функції в одному документі "приєднати" файл із іншим документом MathCAD, у якому перебуває опис програми-функції, що викликається?
- Для такого приєднання існує спеціальний оператор **Reference**, що записується у вигляді, показаному на рисунку.

# Структура оператора Reference



Значок оператора  
Reference



Reference:D:\MathCad\_Apl\Adapt\_Integration.mcd

Повне ім'я файлу, у якому перебуває  
документ, що містить опис програми-функції,  
яка викликається

# Оператор *Reference*



- Оператор ***Reference*** вставляється в текст документа, у якому викликається програма-функція перед її викликом. Для вставки цього оператора необхідно виконати наступні кроки:



# Оператор *Reference*



- *Крок 1. Клацнути лівою кнопкою миші в тім місці, куди буде вставлений оператор **Reference**.*
- *Крок 2. Звернутися до пункту меню **Insert** і виконати команду **Reference**.*
- *Крок 3. У поле діалогового вікна, що з'явилося, ввести повне ім'я файлу, що містить документ із описом функції, що викликається. Для введення імені необхідно клацнути кнопку **Browse** і у діалоговому вікні, що з'явилося, указати диск, папку та ім'я файлу (у поле вводимо повне ім'я файлу).*
- *Крок 4. Після виконаних установок натиснути кнопку **Ok**.*



# Оператори

## *File Output та File Input.*

- Оператор **Reference** дозволяє зчитати інформацію з файла, що має розширення *\*.tcd*.
- Для зчитування та запису файла з іншим розширенням використовуються оператори **File Output** та **File Input**.



# Оператори

## *File Output та File Input.*



- Оператор **File Output** вставляється в текст документа, для зчитування інформації з файлу.
- Оператор **File Input** вставляється в текст документа, для запису інформації в файл.
- Для вставки цих операторів необхідно виконати наступні кроки:



# Оператори

## *File Output та File Input.*



- *Крок 1. Клацнути лівою кнопкою миші в тім місці, куди буде вставлений оператор.*
- *Крок 2. Звернутися до пункту меню **Insert**→**Data** і виконати команду **File Output** чи **File Input**.*
- *Крок 3. У поле діалогового вікна, що з'явилося, ввести повне ім'я файлу, що містить документ із даними, які зчитуються чи записуються. Необхідно вибрати формат файлу. Для введення шляху до файлу необхідно клацнути кнопку **Browse** і у діалоговому вікні, що з'явилося, указати диск, папку та ім'я файлу (у поле вводимо повне ім'я файлу).*
- *Крок 4. Після виконаних установок натиснути кнопку **Готово**.*



# Оператори

## *File Output та File Input.*



- Після виконання цих кроків у документі з'явиться оператор **File Output** чи **File Input** і привласнюємо йому ім'я змінної див. рис.

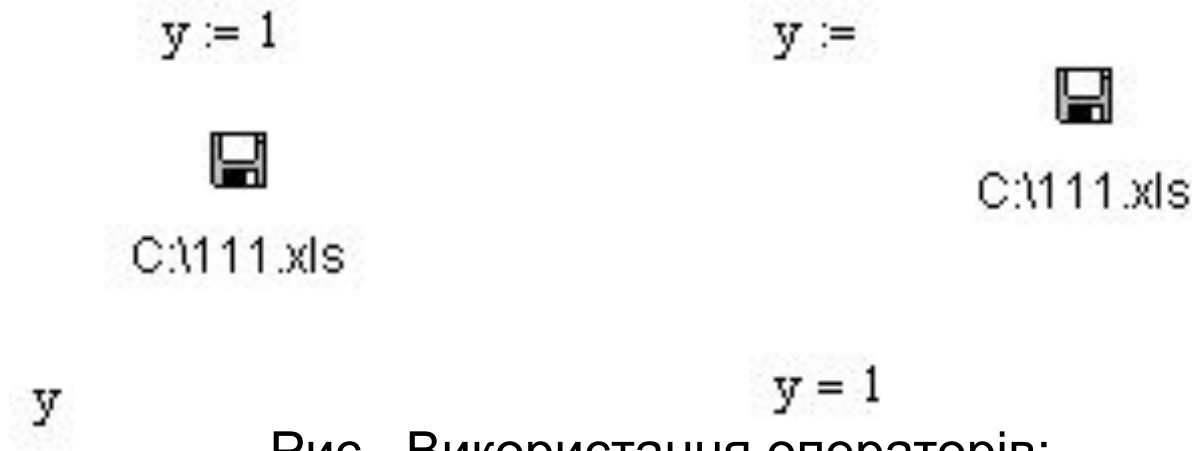


Рис. Використання операторів:  
а). для запису **File Output**; б). для зчитування **File Input**.