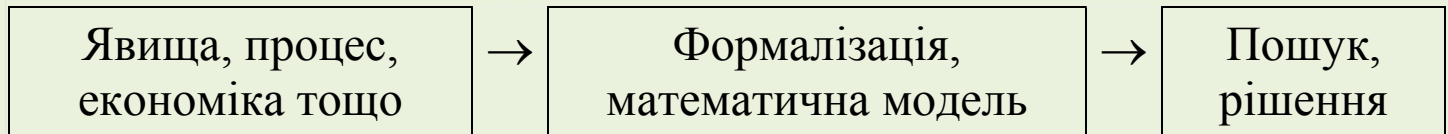


Тема 3. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ. АЛГОРИТМИ. МОВИ ПРОГРАМУВАННЯ. СИСТЕМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ

Математичне моделювання

Розрізняють **фізичні** та **математичні** моделі.

Математична модель – сукупність математичних співвідношень, рівнянь, нерівностей, що описують основні закономірності досліджуваного об'єкту.



Математична модель базується на деякому **спрощенні** (ідеалізації) та є **наближеним** описом реальності.

Критерієм **адекватності** (відповідності) математичної моделі і реального процесу є **практика** (натурний експеримент).

Аналіз математичної моделі дозволяє **прогнозувати** поведінку реального об'єкта.

Розрізняють математичні моделі:

- лінійні або нелінійні моделі;
- детерміновані або стохастичні;
- статичні або динамічні та інші.

Приклад. Математична модель траєкторії тіла:

$$x = t v \cos(\alpha), y = t v \sin(\alpha) - \frac{gt^2}{2}.$$

Виразимо $t = \frac{x}{v \cos(\alpha)}$ та отримаємо: $y = x \operatorname{tg}(\alpha) - \frac{x^2 g}{2v^2 \cos^2(\alpha)}.$

При $y = 0$, маємо відстань льоту: $x = \frac{v^2 \sin(2\alpha)}{g}.$

Імітаційне моделювання (*simulation*) передбачає представлення моделі у вигляді алгоритму та комп'ютерної програми, яка дозволяє відтворити поведінку об'єкту.

Імітаційні моделі розглядаються як **обчислювальні експерименти**, що імітують поведінку реальних об'єктів.

Етапи розв'язку задачі на ПК

1. **Постановка задачі:** повний зміст задачі, мета її розв'язку.
2. **Математична модель розв'язку задачі:** опис задачі в математичних поняттях, визначення вихідних даних; форми виводу одержаних результатів; точність розв'язку задачі.
3. **Вибір методу розв'язку задачі:** по точності, швидкості і об'єму обчислення; вимог до можливостей ПК тощо.
4. **Розробка алгоритму розв'язку задачі:** встановлення логічної послідовності операцій (дій), запис і аналіз алгоритму.
5. **Написання програми на мові:** перевід алгоритму на мову програмування.
6. **Відлагодження програми:** знаходження синтаксичних та логічних помилок.
7. **Обчислювальний експеримент:** здійснення розрахунків та їх аналіз.

Алгоритм - точна і зрозуміла послідовність дій (кроків), які приводять до вирішення задачі.

Вимоги до алгоритму:

- **масовість** - є корисним для цілого класу однотипних задач.
- **конкретність** (визначеність) - виключати довільне тлумачення.
- **результативність** - рішення задачі має кінцеву кількість кроків.

Блок-схеми запису алгоритму розв'язку задачі

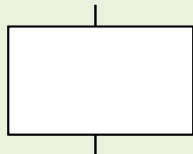
Форми запису алгоритму: блок-схеми, Рі-схеми, словесний опис.

Вимоги до складання блок-схеми алгоритму:

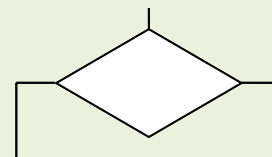
- співвідношення ширини до висоти графічного блоку 3:2;
- блоки потрібно брати розміром кратним 5 од. (10, 15, 20, 25, ...);
- записи можна використовувати як в середині графічного боку, так і зовні за допомогою горизонтальних ліній та квадратних дужок;
- кожен блок має одну точку входу.

Геометричні позначення і пояснення блоків запису алгоритму

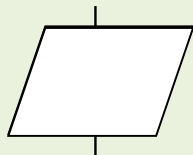
Процес - виконання
обчислювальних дій



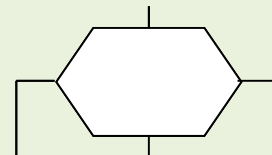
Вирішення - перевірка умови



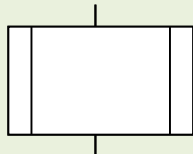
Ввід-вивід - ввід-вивід даних



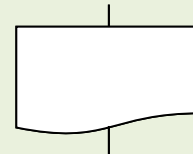
Модифікація - початок циклу



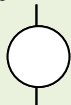
Визначений процес - виконання
функції



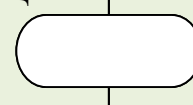
Документ - ввід-вивід
результатів



З'єднувач - розрив ліній потоку

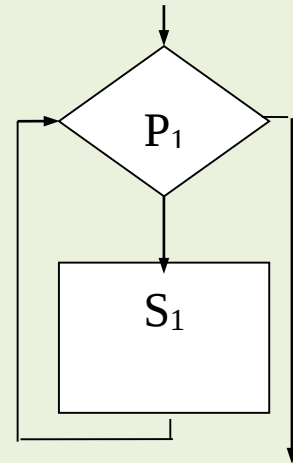
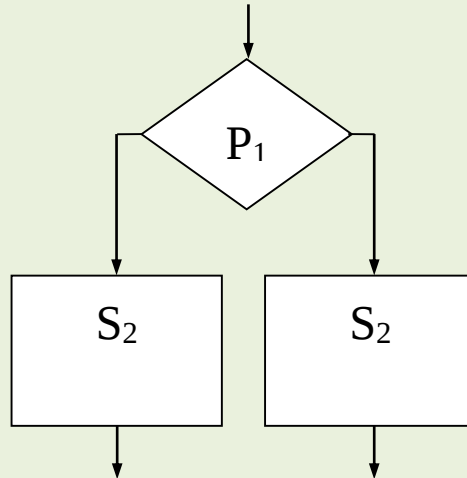
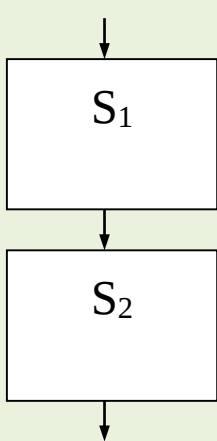


Початок, кінець



Базові алгоритмічні конструкції:

- **лінійна** - складається із послідовних дій;
- **розгалужена (умова)** - послідовність виконання залежить від певних умов;
- **циклічна** - одна і та ж послідовність виконується два і більше раз.



Мови та середовища програмування

Мови програмування - системи позначень для точного запису алгоритму, яка зрозуміла для ПК.

Мови програмування багато в чому нагадують звичайні мови - англійську чи українську і складаються із слів, фраз, синтаксичних правил. Написати програму - перекласти алгоритм мовою програмування, сприйнятливою ПК. Один і той же алгоритм може бути переведений на різні мови програмування.

Мови програмування поділяються на:

- **алгоритмічні** (процедурні) - мови, які описують алгоритм вирішення задачі за допомогою переліку процедур, наприклад, *Basic, Pascal, C, Logo* та інші;
- **функціональні** - опис моделі предметної області засобами ієрархії функцій, наприклад, мова *Lisp*;
- **логічні** - опис моделі предметної області засобами математичної логіки, наприклад, мови *Prolog, SmallTalk*.

Програмування умовно можна поділити на:

- **візуальне** програмування - засоби, за допомогою яких можна швидко створити додаток шляхом візуального проектування його макета в графічному вигляді, наприклад, *Visual C*, *Visual Basic*, *Delphi*.
- мови високого рівня - складається з операторів, схожих на звичайні слова;
- мови низького рівня (асемблер) - переводить зрозумілі людині символи (мнемоніку) в машинні коди.

Переклад з мови програмування у машинні коди здійснюється **трансляторами** - програмами-перекладами.

Можливі два варіанти трансляції:

- **компілятор** - здійснює переклад всієї програми, після чого вона виконується;
- **інтерпретатор** - здійснює переклад частини програми (рядка) і відразу виконує.

Мова Сі та її середовища

Мова Сі була створена в 1972 р Денісом Рітчі для розробки операційної системи *UNIX*.

На мові Сі розроблена графічна система *AutoCAD*, операційна оболонка *Windows* тощо.

Переваги мови Сі:

• потужність	• переносність	• популярність	• розвиток
--------------	----------------	----------------	------------

Система програмування (платформа, інтегроване середовище, IDE) - сукупність технічних і програмних засобів, які реалізують можливість запису операторів на мові програмуванні, їх відлагодження, компіляцію і виконання програми.

Структурний підхід до програмування

Кожній підзадачі відповідає незалежний **програмний модуль** – який, реалізує одну функцію.

Вихідний текст програми потрібно розділити на блоки:

- коротке пояснення призначення модуля, спосіб його виклику, вихідні дані і результат його роботи, дата створення тощо;
- опис всіх змінних, які використовуються в даному модулі;
- алгоритм вирішення задачі;
- обробка результатів вирішення задачі.

Об'єктно-орієнтоване програмування

ООП - метод програмування, який дозволяє розглядати концептуальну модель як **набір об'єктів**.

ООП реалізовано в *Visual C++*, *Delphi* та в інших середовищах.

В основу ООП покладено новий об'єкт (тип даних) - *class*.

Клас складається із набору змінних (даних) і операцій (методів=функцій-членів класу), які працюють лише з цими змінними.

Маніпулювати одержаними класами можна лише тільки за допомогою **повідомлень**.

Базові терміни ООП:

- **Інкапсуляція (*encapsulation*)** - спосіб об'єднання даних і функцій (методів) в єдину структуру. В класах C++ для управління функціональними елементами об'єкту використовуються дескриптори *private*, *public*, *protected*.
- **Ієрархія класів.** Об'єкти, які створені на основі опису класу, називають екземплярами (*instance*) класу. Можна побудувати ієрархію класів, які складаються із базового класу батьків (*parent*) і декількох класів потомків (*child*). Батьківські класи представляють загальні задачі, а похідні класи реалізують більш конкретні задачі.
- **Наслідування (*inheritance*)** - здібність класу наслідувати особливості інших класів.
- **Поліморфізм (*polimorphism*)** - концепція ООП при якій одні і ті ж повідомлення можна посилати батьківському класу і об'єктам похідних класів. Кожний об'єкт підкласу реагує деяким образом на формат повідомлення, відповідно його опису.

Системи комп'ютерної математики

Системи комп'ютерної математики (СКМ) **замінюють** роботу математиків - теоретиків та аналітиків високого рівня.

Є відкриття, зроблені за допомогою таких систем.

Поняття **«комп'ютерна математика»** включає сукупність як теоретичних і методичних засобів, так і сучасних програмних і апаратних засобів, що дозволяють здійснювати всі математичні обчислення з високою точністю і продуктивністю, будувати складні ланцюжки обчислювальних алгоритмів з широкими можливостями візуалізації процесів і даних при їх обробці.

Всі СКМ умовно можна поділити на дві групи:

1. для числових обчислень - *Eureka, MathCAD, MatLAB, Excel*;
2. символьних обчислень - *Maple, MuPAD, Derive, Mathematica*.

Кожна із цих СКМ має характерні особливості внутрішньої побудови та реалізації інтерфейсу.

Система арифметичних обчислень *MathCAD*

MathCAD – СКМ для автоматизації вирішення загальноприйнятих математичних задач в різних областях науки, техніки, у навчанні.

Назва *MathCAD* походить від слів: *MATHeuristic* (математика) і *CAD* (*Computer Aided Design* - система автоматизованого проектування). Розробник - *MathSoft Inc* (<http://www.mathsoft.com>).

На сьогодні використовується *MathCAD 15*. Ціна близько 800\$.

Склад системи:

- **редактор** документів - ввод тексту, формул, графіків тощо;
- *MathConex* - інтегратор з іншими системами;
- **довідкова** система - тематичний та індексований каталог допомоги;
- **швидка шпаргалка** - короткі приклади з мінімальним коментарем.

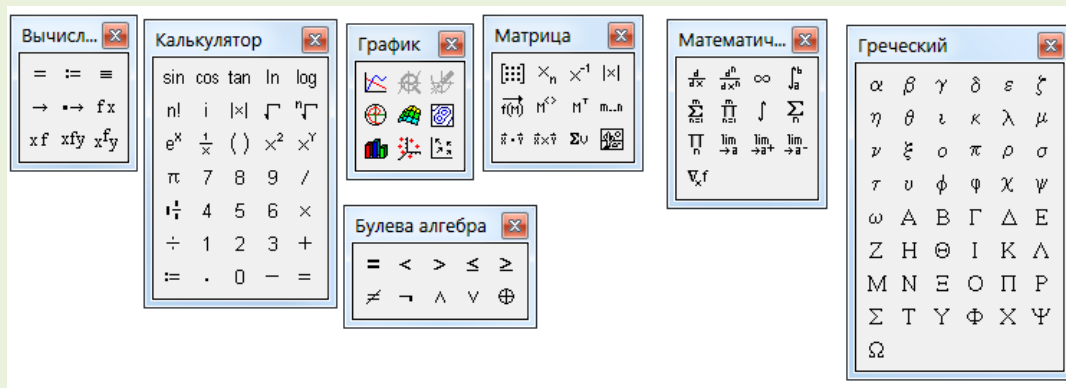
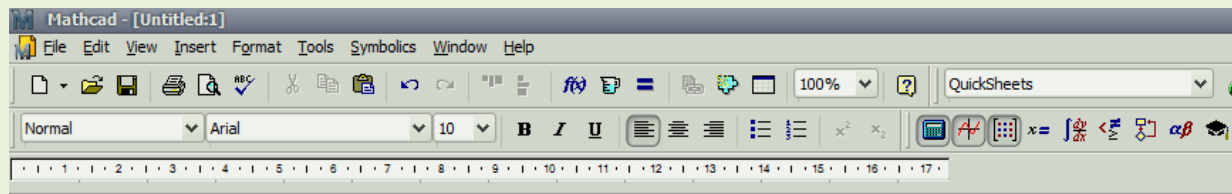
Обчислення в *MathCAD* можна виконати одним із трьох способів:

- вибором операції в меню;
- за допомогою панелей інструментів;

- зверненням до відповідних функцій.

Документ в *MathCAD* називається *WorkSheets* (робочі листки) об'єднує код, який написаний на візуально-орієнтованій мові, що майже не відрізняється від звичайної мови математики.

Інтерфейс *MathCAD* - сукупність засобів, які забезпечують управління системою *MathCAD* через клавіатуру та мишу.



Система інженерних розрахунків *MatLAB*

MatLAB (скорочено від *MATrix LABoratory* - лабораторія матриць) є інтерактивною системою для використання **інженерних** і наукових розрахунків з орієнтацією на роботу з масивами даних.

Створена *MatLAB* фірмою *MathWork Inc* (<http://www.mathwork.com> США). Широко почалася використовуватися наприкінці 80-х років після появи на ринку *MatLAB* 4.0. Ціна близько 2940 \$.

Серед основних областей застосування *MatLAB* - математичні розрахунки, **аналіз** даних і візуалізація, цифрову обробку сигналів та зображень, проектування систем управління, розробка додатків.

Вбудовані універсальні інтерфейси дозволяють здійснювати інтеграцію з процедурами на мовах *C*, *Java* та ін.

Система відрізняється від інших подібних завдяки:

- призначення для інженерних розрахунків;
- використовується мова програмування, яка близька до *BASIC*;
- легкість інтегрування з іншими системами.

Система наукових досліджень *Mathematica*

Mathematica - компанії Wolfram Research, Inc. (<http://www.wolfram.com/>) має широкий набір засобів, що переводять складні математичні алгоритми в програми. Ціна - близько 1460 \$.

Mathematica 1 (1988р.) стала першою системою комп'ютерної алгебри, яка викликала інтерес з боку інженерів і науковців.

Всі алгоритми, що містяться в курсі вищої математики ВНЗ, закладені в середовище системи *Mathematica*.

Недоліки *Mathematica*: погано зрозуміла мова функціонального програмування системи, незвичайність фіксації введення клавішами *Shift+Enter* (зазвичай введення фіксується клавішею *Enter*), незвичні імена функцій, наприклад *Sin[x]*.

Переваги *Mathematica*: наявність динамічних змінних, інтерактивного інтерфейсу, швидкість обчислення завдяки підтримці можливостей сучасних багатоядерних процесорів.

Система символьної алгебри *Maple*

Maple створено групою символьних обчислення, організовану видатним математиком Гоне, в 1980 році в університеті *Waterloo*, Канада. Остання версія є *Maple 2017*. Ціна *Maple 18* - близько 3000 \$.

Maple - це Комп'ютерна Алгебраїчна Система (*CAS - Computer Algebra System*), що означає її дію в рамках законів алгебри в розділах **Обчислення, Диференціальні Рівняння і Лінійна Алгебра** та інші.

Система увібрала в себе більшу частину математичних функцій і правила їх перетворення, які є в математиці.

Maple використовується для вирішення задач аеродинаміки, теорії поля, теплопровідності, теоретичної механіки та ін.

Ядро *Maple* використовується в системах *MathCAD*, *MatLAB*.

Система *Maple* об'єднує:

- мову програмування (є мовою інтерактивної взаємодії з системою);
- редактор для підготовки і редагування документів і програм;

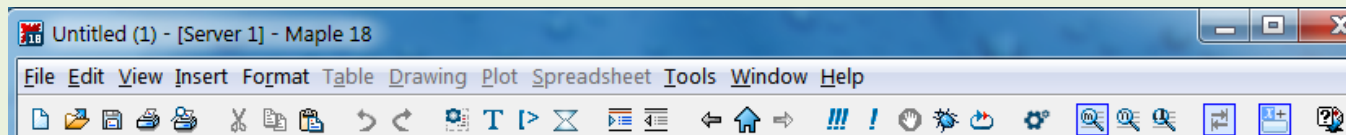
- сучасний багато віконний інтерфейс користувача;
 - потужну довідкову систему з багатьма тисячами прикладів;
 - ядро алгоритмів і правила перетворення математичних виразів;
 - бібліотеки вбудованих функцій (більше 3000 для *Maple 6*) та інше.
- Maple* є потужною системою програмування, яка включає мови:
- **вхідну** – яка використовується для завдання питань (вхідних даних для їх обробки). Ця мова інтерпретуючого типу і має велику кількість визначених математичних і графічних функцій.
 - **реалізації** - *Ci*-мови. На цій мові реалізовано ядро системи.
 - **програмування** - *Maple*-мову. Ця мова має традиційні засоби для вирішення типових задач програмування на основі принципів структурного підходу: цикли, умовні і безумовні переходи, логічні оператори, процедури і т.п.

Більша частина бібліотечних функцій розроблені на *Maple*-мові, що дозволяє користувачам доповнювати чи їх модифікувати.

Інтерфейс системи і робота в діалоговому режимі

Система *Maple* має стандартний графічний інтерфейс користувача для програм, що працюють під управлінням *Windows*.

Файли документів включають повний текст програми, результати обчислення, графічну інформацію тощо.



При завантаженні програми на екрані монітора відображається вікно. У верхній частині, безпосередньо під заголовком вікна, знаходиться головне меню системи:

- **File** – робота з файлами і друком документів;
- **Edit** – команди редагування документа;
- **View** – управління видом інтерфейсу;
- **Insert** – операції вставки;
- **Format** – операції завдання форматів;

- **Spreadsheet** – операції завдання таблиць;
- **Options** – завдання та зміна параметрів системи;
- **Window** – управління вікнами;
- **Help** – робота з довідковою системою.

Робота в діалоговому режимі

Робочий лист складається з області **введення** та області **виводу**.

Діалог з системою *Maple* відбувається в режимі запитань: користувач вводить команди і натисканням [*Enter*] передає їх на виконання ядру системи *Maple*. Результат роботи команди відображаються в області виведення.

Для **введення формули** в стандартній математичній формі треба виконати команду *Insert -Math Input*. При цьому в полі введення буде відображатися команда у формі синтаксису *Maple*, а в області виведення ця ж команда буде відображена в математичній нотації.

Введення текстових коментарів - режим, при якому будь-яка інформація розуміється як текст (невиконуюча). Для вставки текстового коментаря слід виконати команду *Insert -Text* або натиснути кнопку на основній панелі інструментів або ж *F5*.



В *Maple 18* на панелі інструментів введена кнопка **!!!**, яка дублює команду меню *Edit -Execute -Worksheet* (Виконати всю сторінку).

Основні елементи синтаксису *Maple*:

- / мигаюча нахилена риска - маркер введення формули.
- | мигаюча вертикальна риска - маркер введення тексту.
- ; знак фіксації кінця виразу з виведенням результату на екран.
- : знак фіксації кінця виразу, що запобігає виводу результату.
- := оператор присвоювання (наприклад, $x:=4$).
- = оператор рівності для завдання тотожностей і логічних умов ($a=b$), областей змінних ($i=1..5$) і значень опцій ($color=black$).

Перенесення команди на новий рядок (без виконання) здійснюється натисканням комбінації клавіш [*Shift+Enter*].

Для вставки нового рядка: зверху *Ctrl+Shift+K*; знизу *Ctrl+Shift+ J*.
Основні операції редагування містяться в головному меню *Edit*.

- *Delete Paragraph* або ж *Ctrl+Del* - стирання виділених об'єктів.
- *Find ...* або ж *Ctrl+F5* - знайти текстову або математичну формулу.
- *Input Mode* або ж *F5* - включення / вимикання введення текстів.

Робочі листи *Maple* зберігаються у файлах з розширенням *mw*.

Існує можливість зберегти робочий лист у файлах інших форматів, вказавши в діалозі команді *Save As* відповідний формат, або скористатися командами підміню *Export As*:

- *Plain Text .txt* - звичайний текстовий файл.
- *Maple Text .txt* - текстовий формат *Maple*.
- *RTF .rtf* - формат *RTF* (дозволяє вставляти робочі листи в документи *MS Word* із збереженням кольорової палітри).