SCHOOL OF
INFORMATION TECHNOLOGY
AND INNOVATION
BANGKOK UNIVERSITY

CS 410 Final Project

Comparative Study of Language Features

การศึกษาเปรียบเทียบและใช้งานคุณสมบัติของภาษา JAVA และ Prolog

นางสาว ภัทรวดี อุ่นระโลม รหัสนักศึกษา: 1650706441
นาย จักรพันธ์ เพลินทรัพย์ รหัสนักศึกษา: 1650703570
นาย ธนบูลย์ ทองประดา รหัสนักศึกษา: 1650706904

---

1. ฟังก์ชันคำนวณ แฟกทอเรียล

1.1 ประเภทข้อมูลและการประกาศตัวแปร

- JAVA

  ภาษา Java เป็นภาษา Static Typing ที่ต้องระบุประเภทตัวแปรล่วงหน้า และมีการตรวจสอบประเภทตอนคอมไพล์ ฟังก์ชัน factorial(int n) รับค่า int และคืนค่า int โดยใช้การเรียกซ้ำ และ Scanner ใช้เพื่อรับค่า n (ประเภท int) จากผู้ใช้งาน

สรุปการเปลี่ยนแปลงในคำอธิบาย:

- เปลี่ยนจาก "คืนค่า long" เป็น "คืนค่า int" เพื่อให้ตรงกับการประกาศประเภทการคืนค่าของฟังก์ชัน factorial

```
import java.util.Scanner;

public class Factorial {

    public static int factorial(int n) { // ประกาศตัวแปร n เป็นชนิด int
        // ...
    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in); // ประกาศตัวแปร scanner เป็นชนิด
Scanner
        int n = scanner.nextInt();                // ประกาศตัวแปร n เป็นชนิด int

        // ...
        scanner.close();
```

```
    }
}
```

(รูปที่ 1.1 ตัวอย่างการประกาศตัวแปรและประเภทข้อมูล)

- **Prolog**

  ไม่จำเป็นต้องประกาศตัวแปรหรือประเภทข้อมูลล่วงหน้า factorial(0, 1). เป็นข้อเท็จจริงที่กำหนดว่า แฟกทอเรียลของ 0 คือ 1 และส่วนถัดไปเป็นกฎที่อธิบายว่า เมื่อต้องการหาแฟกทอเรียลของจำนวนเต็ม N ใดๆ ที่มากกว่า 0 จะใช้กฎนี้ร่วมกับข้อเท็จจริงที่กำหนดไว้ก่อนหน้า เพื่อนิยามความสัมพันธ์แบบเรียกซ้ำ (recursive relationship)

```
factorial(0, 1).
factorial(N, Result) :-
    N > 0,
    N1 is N - 1,
    factorial(N1, Result1),
    Result is N * Result1.
```

(รูปที่ 1.2 ตัวอย่างการประกาศตัวแปรและประเภทข้อมูล)

หัวข้อ	Java	Prolog
ชนิดข้อมูล	- กำหนดชนิดข้อมูลแบบแน่นอน เช่น int, double,long เป็นต้น	- กำหนดชนิดข้อมูลแบบไม่ตายตัวแปรที่มีอาจแทนค่าหรือโครงสร้างก็ได้
การประกาศตัวแปร	- ต้องประกาศชนิดข้อมูลก่อน เช่น int i = 2 ตัวแปรควรอยู่ method or class ตามโครงสร้างภาษา	- ไม่มีการประกาศชนิดล่วงหน้า ตัวแปรขึ้นต้นด้วยตัวอักษรตัวใหญ่
การตรวจสอบ type	- ตรวจสอบระหว่างคอมไพล์	- ไม่มีการตรวจสอบ type แบบ strict

(ตารางที่ 1 สรุปเปรียบเทียบ)

การเปรียบเทียบการคำนวณแฟกทอเรียลระหว่างภาษา Java และ Prolog

1.รูปแบบการเขียนโปรแกรม

- **Java:** เป็นการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) โดยมีโครงสร้างคลาสชัดเจน มีเมธอด factorial() ที่ถูกเรียกใช้จากเมธอด main()
- **Prolog:** เป็นการเขียนโปรแกรมเชิงตรรกะ (Logic Programming) โดยกำหนดความสัมพันธ์ระหว่างข้อมูลผ่าน predicate ซึ่งไม่มีโครงสร้างคลาสหรือเมธอดแบบภาษาเชิงวัตถุ

2.โครงสร้างควบคุม (เช่น if-else, loop)

- **Java:** ใช้คำสั่ง if-else สำหรับการตรวจสอบเงื่อนไข มีการเรียกซ้ำแบบ recursive และมีการวนลูปโดยนัยผ่านกระบวนการเรียกซ้ำ
- **Prolog:** ไม่มีคำสั่ง if-else แบบชัดเจน แต่ใช้การตรวจสอบเงื่อนไขผ่านการจับคู่รูปแบบ (pattern matching) และการนิยามหลายกรณี (multiple clauses) ใช้การเรียกซ้ำโดยไม่มีการวนลูปแบบเดิมๆ

3.การรับข้อมูลและแสดงผล

- **Java:** มีการรับข้อมูลผ่าน Scanner และแสดงผลด้วย System.out.println() มีข้อความแนะนำผู้ใช้และแสดงผลลัพธ์พร้อมข้อความอธิบาย
- **Prolog:** ในโค้ดตัวอย่างไม่มีการรับข้อมูลและแสดงผลโดยตรง เป็นเพียงการนิยามความสัมพันธ์ของแฟกทอเรียลเท่านั้น

4.การทำงานเชิงทฤษฎี / ความแตกต่าง / ประสิทธิภาพ

Java:

- ทำงานแบบการประมวลผลตามลำดับ (imperative) เป็นขั้นเป็นตอน
- ใช้หน่วยความจำสำหรับ call stack ในการเรียกซ้ำ อาจเกิดปัญหา stack overflow เมื่อค่า n มากๆ
- มีการตรวจสอบข้อมูลนำเข้าว่าต้องไม่เป็นค่าลบ
- เหมาะกับนักพัฒนาที่คุ้นเคยกับภาษาที่มีโครงสร้างแบบ C-like

Prolog:

- ทำงานแบบการประมวลผลเชิงประกาศ (declarative) โดยบอกว่า "อะไร" มากกว่า "อย่างไร"
- ใช้การ unification และการ backtracking ในการหาคำตอบ
- มีการเขียนที่กระชับกว่า เน้นความสัมพันธ์ทางคณิตศาสตร์โดยตรง

- รูปแบบ factorial(0, 1) เป็นการระบุกรณีฐานแยกต่างหาก
  ทำให้อ่านง่ายสำหรับผู้เข้าใจแนวคิดการเขียนโปรแกรมเชิงตรรกะ
- ใช้ตัวแปร Result เพื่อเก็บผลลัพธ์แทนการ return ค่าแบบใน Java

2. อัลกอริทึม จัดเรียงข้อมูล bubble sort

2.1 ประเภทข้อมูลและการประกาศตัวแปร

- JAVA

  ในโปรแกรมนี้มีการใช้ประเภทข้อมูลและการประกาศตัวแปรดังนี้

  - **int** ใช้เก็บค่าตัวเลข เช่น ขนาดของอาร์เรย์ ค่าภายในอาร์เรย์ และตัวแปรควบคุมลูป เช่น

    int size, temp, i, j, low, high, pivot, pi; และอาร์เรย์เช่น int[] arr, arrBubble, arrQuick;

  - **boolean** ใช้ในการตรวจสอบว่ามีการสลับค่าหรือไม่ในแต่ละรอบของการเรียงลำดับแบบ Bubble Sort เช่น

    boolean swapped;

  - **Scanner** เป็นคลาสที่ใช้รับข้อมูลจากผู้ใช้ผ่านแป้นพิมพ์ เช่น

    Scanner scanner = new Scanner(System.in);

ตัวแปรทั้งหมดถูกประกาศและใช้งานภายในฟังก์ชันต่าง ๆ ได้แก่ main(), bubbleSort(), quickSort() และ partition() เพื่อทำการรับข้อมูล เรียงลำดับ และแสดงผลข้อมูลตามขั้นตอนของอัลกอริทึมการเรียงลำดับ

```
import java.util.Arrays;
import java.util.Scanner;

public class SortingAlgorithms {

    public static void bubbleSort(int[] arr) {
        int n;
        int i, j;
        boolean swapped;
        int temp;
    }

    public static void quickSort(int[] arr, int low, int high) {
        int pi;
    }

    public static int partition(int[] arr, int low, int high) {
        int pivot;
        int i, j;
        int temp;
        return 0; // placeholder return
    }

    public static void main(String[] args) {
        Scanner scanner;
```

```
        int size;
        int[] arr;
        int i;
        int[] arrBubble;
        int[] arrQuick;
    }
}
```

(รูปที่ 2.1 ตัวอย่างการประกาศตัวแปรและประเภทข้อมูล)

- **Prolog**

การเขียนโปรแกรมในภาษา Prolog ใช้การประกาศตัวแปรโดยขึ้นต้นด้วยตัวอักษรตัวใหญ่หรือขีดล่าง และไม่มีการระบุชนิดข้อมูลแบบตายตัว โดยตัวแปรจะถูกผูกค่าจากกระบวนการจับคู่ระหว่างการทำงาน โครงสร้างของโปรแกรมประกอบด้วยข้อกำหนดของกฎ (predicate) พร้อมอาร์กิวเมนต์ที่เป็นตัวแปร ซึ่งสามารถแทนข้อมูลประเภทต่าง ๆ ได้ เช่น ลิสต์ ตัวเลข หรือโครงสร้างอื่น ๆ โค้ดที่นำเสนอแสดงให้เห็นการใช้ตัวแปรเพื่อวางรูปแบบของฟังก์ชันจัดเรียงข้อมูล (Bubble Sort และ Quicksort) โดยเน้นที่โครงสร้างของตัวแปรและความสัมพันธ์ระหว่างกัน ซึ่งเป็นแนวทางสำคัญในการออกแบบโปรแกรมเชิงตรรกะในภาษา Prolog.

```
% Bubble Sort (แสดงเฉพาะโครงสร้างตัวแปร)
bubble_sort(List, Sorted).
bubble_sort(List, Acc, Sorted).
bubble([X], [], [], X).
bubble(X, [Y|T], [Y|NT], Max).
bubble(X, [Y|T], [X|NT], Max).

% Quicksort (แสดงเฉพาะโครงสร้างตัวแปร)
quicksort(List, Sorted).
quicksort(List, Sorted) :-
    partition(Rest, Pivot, Less, Greater),
    quicksort(Less, SortedLess),
    quicksort(Greater, SortedGreater),
    append(SortedLess, [Pivot|SortedGreater], Sorted).

partition([], Pivot, Less, Greater).
partition([X|T], Pivot, [X|Less], Greater).
partition([X|T], Pivot, Less, [X|Greater]).
```

(รูปที่ 2.2 ตัวอย่างการประกาศตัวแปรและประเภทข้อมูล)

หัวข้อ	Java	Prolog
ชนิดข้อมูล	- ต้องระบุชนิดข้อมูลอย่างชัดเจน เช่น int, boolean และไม่สามารถใช้ผิดชนิดได้ (static typing)	- ไม่มีการระบุชนิดข้อมูลแบบตายตัว (dynamic typing) - ใช้ข้อมูลได้หลายชนิดโดยไม่ต้องกำหนดล่วงหน้า
การประกาศตัวแปร	- ระบุชนิดก่อนใช้งานเช่น int x = 5;	- ไม่มีการประกาศตัวแปรแบบประจำ ตัวแปรเริ่มต้นด้วยอักษรใหญ่ และจับค่าระหว่างการรัน
การตรวจสอบ type	- ตรวจสอบตอนคอมไพล์ (compile-time type checking)	- ไม่มีการตรวจสอบแบบเข้มงวด (no strict type checking) - ใช้ pattern matching และ unification

(ตารางที่ 2.1 สรุปเปรียบเทียบ)

การเปรียบเทียบอัลกอริทึมการเรียงลำดับระหว่างภาษา Java และ Prolog

รูปแบบการเขียนโปรแกรม

Java:

- เป็นการเขียนโปรแกรมเชิงวัตถุที่มีโครงสร้างชัดเจน มีการสร้างคลาส SortingAlgorithms ที่ประกอบด้วยเมธอดต่างๆ
- มีการแบ่งแยกเมธอดตามหน้าที่การทำงาน เช่น bubbleSort(), quickSort() และ partition()
- ใช้รูปแบบการเขียนแบบ imperative ที่ระบุขั้นตอนการทำงานอย่างชัดเจน

Prolog:

- เป็นการเขียนโปรแกรมเชิงตรรกะที่กำหนดความสัมพันธ์ระหว่างข้อมูลผ่าน predicate
- ไม่มีโครงสร้างคลาส แต่แบ่งเป็นกลุ่มของ predicate ที่ทำงานร่วมกัน
- ใช้รูปแบบการเขียนแบบ declarative ที่เน้นการอธิบายความสัมพันธ์มากกว่าขั้นตอนการทำงาน

โครงสร้างควบคุม (เช่น if-else, loop)

Java:

- ใช้ loop แบบ for เพื่อวนซ้ำผ่านอาร์เรย์
- ใช้เงื่อนไข if-else สำหรับการตัดสินใจและเปรียบเทียบค่า
- มีการใช้การเรียกซ้ำ (recursion) ในอัลกอริทึม Quick Sort
- มีการใช้ตัวแปร boolean swapped เพื่อเพิ่มประสิทธิภาพการทำงานของ Bubble Sort

Prolog:

- ไม่มีการใช้ loop แบบ for หรือ while แต่ใช้การเรียกซ้ำ (recursion) แทน
- ไม่มีการใช้ if-else แบบดั้งเดิม แต่ใช้การจับคู่รูปแบบ (pattern matching) และการนิยามหลายกรณี
- แต่ละ predicate มีหลายเงื่อนไข (multiple clauses) สำหรับสถานการณ์ต่างๆ

การรับข้อมูลและแสดงผล

Java:

- มีการรับข้อมูลจากผู้ใช้ผ่าน Scanner
- มีการแสดงข้อความแนะนำผู้ใช้อย่างชัดเจน
- แสดงผลลัพธ์ด้วย System.out.println() และใช้ Arrays.toString() เพื่อแสดงอาร์เรย์
- สามารถเปรียบเทียบผลลัพธ์จากทั้งสองอัลกอริทึมในการรันครั้งเดียว

Prolog:

- ในโค้ดตัวอย่างไม่มีการรับข้อมูลและแสดงผลโดยตรง
- การใช้งานต้องเรียกผ่าน query ของ Prolog ซึ่งไม่ได้แสดงในโค้ด
- ไม่มีการแสดงผลลัพธ์เปรียบเทียบระหว่างอัลกอริทึม

การทำงานเชิงทฤษฎี / ความแตกต่าง / ประสิทธิภาพ

Java:

- Bubble Sort:
    - ใช้การเปรียบเทียบและสลับตำแหน่งของอาร์เรย์โดยตรง
    - มีการปรับปรุงประสิทธิภาพด้วยตัวแปร swapped เพื่อหยุดการทำงานเมื่อข้อมูลเรียงลำดับแล้ว
    - ทำงานบนข้อมูลชุดเดิมโดยไม่ต้องสร้างอาร์เรย์ใหม่ (in-place sorting)
- Quick Sort:
    - ใช้แนวคิด divide-and-conquer โดยเลือก pivot และแบ่งข้อมูลเป็นสองส่วน

- มีการเรียกตัวเองซ้ำเพื่อเรียงลำดับข้อมูลในแต่ละส่วนย่อย
- ประสิทธิภาพเฉลี่ยดีกว่า Bubble Sort สำหรับข้อมูลจำนวนมาก

Prolog:

- Bubble Sort:
    - ใช้การสร้างรายการใหม่ในแต่ละรอบแทนการสลับค่าในรายการเดิม
    - แบ่งการทำงานเป็นขั้นตอนย่อยๆ ผ่าน predicate หลายตัว
    - ใช้ตัวสะสม (accumulator) ในการสร้างรายการผลลัพธ์
- Quick Sort:
    - มีการเขียนที่กระชับและสะท้อนแนวคิดของอัลกอริทึมโดยตรง
    - ใช้ predicate partition แยกข้อมูลเป็นสองกลุ่มตาม pivot
    - ใช้ predicate append ในการรวมรายการที่เรียงลำดับแล้ว
    - มีการสร้างรายการใหม่ในแต่ละขั้นตอน ต่างจาก Java ที่ทำงานบนอาร์เรย์เดิม

ความแตกต่างที่สำคัญคือ Java ใช้แนวคิดเชิงคำสั่งที่ระบุขั้นตอนการทำงานชัดเจน ขณะที่ Prolog ใช้แนวคิดเชิงประกาศที่เน้นการอธิบายความสัมพันธ์ของข้อมูล ทำให้โค้ด Prolog มีความกระชับและเข้าใจง่ายในเชิงแนวคิด แต่อาจมีประสิทธิภาพด้อยกว่าใน Java เนื่องจากมีการสร้างรายการใหม่บ่อยครั้ง

3. เครื่องคิดเลขอย่างง่าย

3.1 ประเภทข้อมูลและการประกาศตัวแปร

- JAVA

ในโปรแกรมเครื่องคิดเลขที่พัฒนาโดยใช้ภาษา Java มีการประกาศตัวแปรหลายประเภทเพื่อรองรับการทำงานของระบบ GUI และกระบวนการคำนวณทางคณิตศาสตร์ โดยใช้ตัวแปรชนิด JTextField สำหรับแสดงค่าที่ป้อนและผลลัพธ์ (display), ชนิด JPanel สำหรับจัดวางปุ่มต่าง ๆ (buttonPanel), และ JButton[]
ซึ่งเป็นอาร์เรย์ของปุ่มที่ใช้แสดงตัวเลขและเครื่องหมายบนเครื่องคิดเลข (buttons)
โดยกำหนดข้อความป้ายของปุ่มเหล่านี้ไว้ในอาร์เรย์ชนิด String ชื่อ buttonLabels เช่น "1", "+", "C" เป็นต้น

สำหรับการคำนวณ มีการประกาศตัวแปรชนิด double ชื่อ num1 เพื่อเก็บตัวเลขตัวแรกที่ป้อนเข้ามา
โดยรองรับค่าทศนิยม และตัวแปร String ชื่อ operator เพื่อเก็บเครื่องหมายคำนวณที่เลือกใช้งาน เช่น บวก ลบ คูณ หาร
ส่วนตัวแปร boolean ชื่อ isNewNumber ใช้ตรวจสอบสถานะการป้อนตัวเลขใหม่ ว่ากำลังเริ่มต้นการคำนวณรอบใหม่หรือไม่
ซึ่งทั้งหมดนี้เป็นการประกาศตัวแปรโดยระบุประเภทข้อมูลให้เหมาะสมกับการใช้งานในแต่ละส่วนของโปรแกรม
เพื่อให้การประมวลผลและการแสดงผลทำงานได้อย่างถูกต้องและมีประสิทธิภาพ

```
private JTextField display;                 // ประกาศตัวแปรชนิด JTextField สำหรับแสดงผล
private JPanel buttonPanel;                 // ตัวแปร panel สำหรับวางปุ่ม
private JButton[] buttons;                  // อาเรย์ของปุ่ม (JButton)
private String[] buttonLabels = {           // อาเรย์ข้อความสำหรับ label ปุ่มต่างๆ
    "7", "8", "9", "/",
    "4", "5", "6", "*",
    "1", "2", "3", "-",
    "0", ".", "=", "+", "C"
};
private double num1 = 0;                    // ตัวแปรชนิด double สำหรับเก็บตัวเลขตัวที่ 1
private String operator = "";               // ตัวแปรชนิด String สำหรับเก็บเครื่องหมาย + - * /
private boolean isNewNumber = true;         // ตัวแปร boolean สำหรับตรวจว่าเป็นการป้อนตัวเลขใหม่หรือไม่
```

(รูปที่ 3.1 ตัวอย่างการประกาศตัวแปรและประเภทข้อมูล)

- Prolog

ใน Prolog ตัวแปรไม่มีการกำหนดประเภทข้อมูลล่วงหน้า แต่จะรับค่าที่เหมาะสมตามบริบทการใช้งานจริง เช่น X, Y, R มักจะเป็นตัวเลข (integer/float), Op เป็น atom, และค่าผลลัพธ์สามารถเป็นตัวเลขหรือ string

```
% ตัวแปรที่ใช้ในโปรแกรม
% X, Y      - ตัวเลขที่ผู้ใช้ป้อน (ชนิด: number)
% Op        - ตัวดำเนินการ (ชนิด: atom เช่น '+', '-', '*', '/')
% Result    - ผลลัพธ์จากการคำนวณ (ชนิด: number หรือ string ในกรณี error)

main :-
    read(X),          % X: number
    read(Op),         % Op: atom ('+', '-', '*', '/')
    read(Y),          % Y: number
    calculator(X, Op, Y, Result),
    write(Result).
```

(รูปที่ 3.2 ตัวอย่างการประกาศตัวแปรและประเภทข้อมูล)

หัวข้อ	Java	Prolog
ชนิดข้อมูล	- ต้องระบุชนิดข้อมูล เช่น int, double, char ก่อนใช้งาน	- ไม่มีการระบุชนิดข้อมูลล่วงหน้า - ใช้ symbol หรือค่าต่าง ๆ ได้โดยตรงตามบริบท
การประกาศตัวแปร	- ต้องระบุชนิดก่อนใช้ เช่น int x = 5;	- ตัวแปรถูกสร้างขึ้นโดยการใช้งานครั้งแรก เช่น calculator(X, '+', Y, R)
การตรวจสอบ type	- ตรวจสอบขณะคอมไพล์ หากชนิดข้อมูลไม่ตรงจะเกิดข้อผิดพลาด	- ไม่ตรวจสอบชนิดข้อมูลแบบเข้มงวด - หากเงื่อนไขไม่เป็นจริง เช่น / 0 จะ fail

(ตารางที่ 3.1 สรุปเปรียบเทียบ)

การเปรียบเทียบเครื่องคิดเลขระหว่างภาษา Java และ Prolog

รูปแบบการเขียนโปรแกรม

Java:

- เป็นการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) ที่มีโครงสร้างซับซ้อนกว่า โดยสร้างคลาส CalculatorGUI ที่สืบทอดจาก JFrame และใช้ ActionListener เพื่อจัดการการทำงานของปุ่ม
- มีการจัดการส่วนติดต่อผู้ใช้แบบกราฟิก (GUI) ด้วย Swing ซึ่งมีองค์ประกอบหลายส่วน เช่น หน้าจอแสดงผล (JTextField) และปุ่มกด (JButton)
- มีการออกแบบเชิงวัตถุที่ชัดเจน โดยกำหนดคุณสมบัติ (attributes) และพฤติกรรม (behaviors) ของเครื่องคิดเลขในคลาสเดียวกัน

Prolog:

- เป็นการเขียนโปรแกรมเชิงตรรกะ (Logic Programming) ที่มีโครงสร้างเรียบง่าย
- กำหนดความสัมพันธ์ของการคำนวณผ่าน predicate calculator/4 (รับ 4 พารามิเตอร์)
- ไม่มีส่วนติดต่อผู้ใช้แบบกราฟิก แต่ใช้การโต้ตอบผ่านคอนโซล (console) ด้วยคำสั่ง write/1 และ read/1
- สะท้อนแนวคิดการเขียนโปรแกรมแบบประกาศ (declarative) ที่เน้นการกำหนด "อะไร" มากกว่า "อย่างไร"

โครงสร้างควบคุม (เช่น if-else, loop)

Java:

- ใช้ if-else statements และ switch-case เพื่อตรวจสอบเงื่อนไขต่างๆ เช่น การตรวจสอบว่าปุ่มที่กดเป็นตัวเลข เครื่องหมาย หรือปุ่มพิเศษ
- มีการใช้ loop for เพื่อสร้างปุ่มตัวเลขและเครื่องหมายต่างๆ
- มีการจัดการเหตุการณ์ (event handling) ผ่านเมธอด actionPerformed() ที่ทำงานเมื่อผู้ใช้กดปุ่ม

Prolog:

- ไม่มีการใช้ if-else หรือ loop แบบในภาษาเชิงขั้นตอนวิธี แต่ใช้การกำหนดกฎ (rules) และข้อเท็จจริง (facts)
- ใช้เทคนิคการจับคู่รูปแบบ (pattern matching) และการทำงานแบบ backtracking
- มีโครงสร้างควบคุมเรียบง่าย ผ่านการกำหนดกรณีพิเศษ เช่น การหารด้วยศูนย์

การรับข้อมูลและแสดงผล

Java:

- มีส่วนติดต่อผู้ใช้แบบกราฟิก (GUI) ที่สมบูรณ์และใช้งานง่าย ประกอบด้วยช่องแสดงผลและปุ่มกดต่างๆ
- รับข้อมูลผ่านการกดปุ่มบนหน้าจอ

- แสดงผลลัพธ์ในช่องแสดงผล (JTextField) ที่อยู่ด้านบนของหน้าต่าง
- มีการจัดรูปแบบการแสดงผลอย่างเป็นระเบียบ เช่น การจัดตำแหน่งข้อความชิดขวา

Prolog:

- ใช้การโต้ตอบผ่านคอนโซลแบบพื้นฐาน
- รับข้อมูลผ่านคำสั่ง read/1 ที่รอให้ผู้ใช้ป้อนข้อมูล
- แสดงผลลัพธ์ผ่านคำสั่ง write/1 และ nl (new line)
- ไม่มีการจัดรูปแบบการแสดงผลพิเศษ

การทำงานเชิงทฤษฎี / ความแตกต่าง / ประสิทธิภาพ

Java:

- ทำงานแบบเชิงขั้นตอนวิธี (imperative) ที่กำหนดขั้นตอนการทำงานชัดเจน
- รองรับการคำนวณอย่างต่อเนื่อง โดยเก็บผลลัพธ์ก่อนหน้าไว้ใช้ในการคำนวณถัดไป
- มีการจัดการกรณีพิเศษหลายกรณี เช่น การป้อนจุดทศนิยม การล้างค่า และการตรวจสอบการหารด้วยศูนย์
- มีความยืดหยุ่นสูง สามารถขยายฟังก์ชันการทำงานได้ง่าย เช่น เพิ่มปุ่มหรือฟังก์ชันใหม่
- โค้ดมีความซับซ้อนกว่าและยาวกว่า แต่มอบประสบการณ์ผู้ใช้ที่ดีกว่า

Prolog:

- ทำงานแบบเชิงประกาศ (declarative) ที่เน้นความสัมพันธ์ของข้อมูลมากกว่าขั้นตอนการทำงาน
- จัดการเพียงการคำนวณครั้งเดียว ไม่มีการเก็บประวัติการคำนวณ
- จัดการกรณีพิเศษเฉพาะการหารด้วยศูนย์
- มีความยืดหยุ่นน้อยกว่า แต่โค้ดมีความกระชับและเข้าใจง่ายในเชิงตรรกะ
- เหมาะสำหรับการคำนวณแบบตรรกะและความสัมพันธ์

ความแตกต่างที่สำคัญคือ Java มุ่งเน้นการสร้างแอปพลิเคชันที่ใช้งานได้จริงและมีส่วนติดต่อผู้ใช้ที่ดี ในขณะที่ Prolog มุ่งเน้นการกำหนดความสัมพันธ์ทางตรรกะและการคำนวณแบบง่ายๆ โดยไม่ให้ความสำคัญกับส่วนติดต่อผู้ใช้ ทำให้ Java เหมาะสำหรับการพัฒนาแอปพลิเคชันสำหรับผู้ใช้ทั่วไป ในขณะที่ Prolog
เหมาะสำหรับการแก้ปัญหาที่ต้องการการใช้เหตุผลและตรรกะ

4. วิเคราะห์เชิงเปรียบเทียบ

Java เหมาะสำหรับงานทั่วไปที่ต้องการความควบคุมสูงและมีเครื่องมือพัฒนาเยอะ ส่วน Prolog เหมาะกับงานที่เน้นตรรกะ เช่น AI, expert systems, หรือการค้นหาความสัมพันธ์ที่ซับซ้อน

1. ประเภทข้อมูลและการประกาศตัวแปร

Java: ระบุประเภทตัวแปรชัดเจน เช่น int, double, char

Prolog: ตัวแปรไม่ต้องประกาศชนิด ใช้ dynamic typing โดยใช้การจับคู่ (unification)

2. ขอบเขตการใช้งานของตัวแปร

Java: มีขอบเขตในระดับเมธอด/คลาส

Prolog: ตัวแปรมีขอบเขตเฉพาะภายในแต่ละกฎหรือข้อคำถาม

3. วิธีการส่งพารามิเตอร์

Java: ส่งแบบค่าหรืออ้างอิงตามชนิดข้อมูล (pass-by-value / pass-by-reference)

Prolog: ใช้การจับคู่พารามิเตอร์ตามกฎ (unification)

4. โครงสร้างควบคุม (if-else, loop)

Java: ใช้ if, switch, for, while ตามแบบ imperatively

Prolog: ใช้การจับคู่, เงื่อนไขด้วย -> ;, และการวนซ้ำผ่าน recursion

5. การประมวลผลแบบขนาน (Parallelism)

Java: รองรับด้วย Thread, Executor, Streams

Prolog: บาง implementation รองรับ parallelism, แต่แบบพื้นฐานทำงานทีละกฎ

6. ความง่ายในการใช้งาน / การอ่านโค้ด / ประสิทธิภาพ

Java: อ่านง่ายสำหรับโปรแกรมเชิงขั้นตอน, มีประสิทธิภาพสูงสำหรับงานทั่วไป

Prolog: เข้าใจยากกว่าเล็กน้อย เหมาะกับงานด้านตรรกะ การพิสูจน์ และการวางเงื่อนไข