Contents

ความยาก: **

เนื้อหาต่อจากนี้ถูกสร้างขึ้นโดย AI โดยใช้ข้อมูลจาก Presentation Slides เรื่อง Variable and Data types

---

2.1 การเขียนโปรแกรม Java เบื้องต้น

- **Class (คลาส)**: โครงสร้างพื้นฐานของโปรแกรม Java ทุกโปรแกรมจะต้องเริ่มต้นด้วยคลาส
- **Main Method (เมธอดหลัก)**: จุดเริ่มต้นของการทำงานในโปรแกรม Java

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

2.2 ตัวแปร (Variables)

- **ตัวแปร** คือ กล่องที่ใช้เก็บข้อมูลในหน่วยความจำ
- การประกาศตัวแปรใน Java ต้องระบุ **ชนิดข้อมูล (Type)** ก่อนเสมอ เช่น `int`, `double`, `String`

```
int age = 25;
double salary = 50000.0;
```

2.3 ชนิดข้อมูลพื้นฐาน (Primitive Data Types)

ชนิดข้อมูล	ขนาด (บิต)	คำอธิบาย
`byte`	8	จำนวนเต็มขนาดเล็ก
`short`	16	จำนวนเต็มขนาดกลาง
`int`	32	จำนวนเต็มทั่วไป
`long`	64	จำนวนเต็มขนาดใหญ่
`float`	32	จำนวนทศนิยม
`double`	64	จำนวนทศนิยมที่มีความแม่นยำสูง
`char`	16	ตัวอักษร Unicode
`boolean`	1	ค่าจริงหรือเท็จ (`true`/`false`)

2.4 นิพจน์และการกำหนดค่าให้ตัวแปร (Expressions & Variable Assignment)

- การกำหนดค่าให้ตัวแปรใช้เครื่องหมาย `=`

```
int x = 10;
double interest = principal * rate;
```

2.5 เมธอด (Subroutines)

- **Subroutine** คือ ชุดคำสั่งที่รวบรวมไว้ภายใต้ชื่อเดียว และสามารถเรียกใช้ซ้ำได้

```
public static int add(int a, int b) {
    return a + b;
}
```

2.6 สายอักขระ (Strings) **

- **String** คือ ลำดับของตัวอักษร ซึ่งจัดการโดยคลาส `String`
- **เมธอดที่ใช้บ่อย**:
  - `length()`: คืนค่าความยาวของสายอักขระ
  - `charAt(index)`: คืนค่าตัวอักษรที่ตำแหน่งที่ระบุ
  - `toUpperCase()`, `toLowerCase()`: แปลงสายอักขระเป็นตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็ก

2.7 การใช้ Enum (Enumerations) *

- **Enum** คือ ชนิดข้อมูลที่มีค่าคงที่ที่กำหนดไว้ล่วงหน้า

```
enum Season { SPRING, SUMMER, FALL, WINTER }
Season vacation = Season.SUMMER;
```

- **เมธอด `ordinal()`**: ใช้คืนค่าลำดับของค่าใน Enum

```
System.out.println(Season.SUMMER.ordinal()); //     : 1
```

2.8 ตัวอย่างการใช้ Enum *

```
public class EnumDemo {
    enum Day { SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY }
    public static void main(String[] args) {
        Day tgif = Day.FRIDAY;
        System.out.println(tgif + " is the " + tgif.ordinal() + "-th day of the week.");
    }
}
```

2.9 กฎการตั้งชื่อใน Java (Syntax Rules)

- **Identifiers (ตัวระบุ)** คือ ชื่อที่ใช้เรียกตัวแปร คลาส หรือเมธอด
- ต้องเริ่มต้นด้วยตัวอักษรหรือ `_` และไม่มีช่องว่างระหว่างชื่อ
- ตัวพิมพ์ใหญ่และตัวพิมพ์เล็กถือว่าแตกต่างกัน เช่น `HelloWorld` ไม่เหมือนกับ `helloworld`
- ห้ามใช้ **Reserved Words (คำสงวน)** เช่น `class`, `public`, `static`, `if`, `else`

2.10 โครงสร้างโปรแกรม Java

- โครงสร้างพื้นฐานของโปรแกรม Java:

```
public class ProgramName {
    public static void main(String[] args) {
        //
    }
}
```

- ชื่อคลาสต้องตรงกับชื่อไฟล์ เช่น คลาส `HelloWorld` ต้องบันทึกในไฟล์ `HelloWorld.java`

2.11 ชนิดข้อมูล Math และเมธอดที่สำคัญ *

- คลาส **Math** มีเมธอดที่ใช้คำนวณต่างๆ เช่น
  - `Math.abs(x)`: ค่าสัมบูรณ์
  - `Math.pow(x, y)`: ยกกำลัง
  - `Math.random()`: สุ่มตัวเลขระหว่าง 0 ถึง 1

2.12 การวัดเวลาในโปรแกรม *

- ใช้เมธอด `System.currentTimeMillis()` เพื่อวัดเวลาปัจจุบันในหน่วยมิลลิวินาที
- สามารถนำไปใช้วัดเวลาการทำงานของโปรแกรมได้

```
long startTime = System.currentTimeMillis();
//
long endTime = System.currentTimeMillis();
System.out.println("Run time: " + (endTime - startTime) + " ms");
```

2.13 การรับค่า Program Arguments

- โปรแกรม Java สามารถรับค่า argument จากคอมมานด์ไลน์ได้

```
public static void main(String[] args) {
    System.out.println(args[0]); //      argument
}
```

- ใช้ `Integer.parseInt()` หรือ `Double.parseDouble()` เพื่อแปลงค่าจาก String เป็นตัวเลข

2.14 ตัวอย่างโปรแกรม Java

- **โปรแกรม HelloWorld**

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

- **โปรแกรมคำนวณดอกเบี้ย (Interest.java)**

```
public class Interest {
    public static void main(String[] args) {
        double principal = 17000;
        double rate = 0.07;
        double interest = principal * rate;
        principal = principal + interest;

        System.out.println("The interest earned is $" + interest);
        System.out.println("The value after one year is $" + principal);
    }
}
```

Contents

1 Java

2 สรุปเนื้อหาเรื่อง Operations, Packages, and Programming Styles (พร้อมตัวอย่างโค้ด)

เนื้อหาต่อจากนี้ถูกสร้างขึ้นโดย AI โดยใช้ข้อมูลจาก Presentation Slides เรื่อง Operations, Packages, and Programming Styles

---

2.1 การดำเนินการทางคณิตศาสตร์ใน Java (Arithmetic Operations)

- **ตัวอย่างการใช้งานตัวดำเนินการพื้นฐาน**

```
int a = 10;
int b = 3;

System.out.println(a + b);   //     : 13
System.out.println(a - b);   //     : 7
System.out.println(a * b);   //     : 30
System.out.println(a / b);   //     : 3
System.out.println(a % b);   //     : 1
```

2.2 ตัวดำเนินการแบบย่อ (Shortcut Operators)

```
int num = 5;

num += 3;  // num = num + 3;
System.out.println(num);  //     : 8

num *= 2;  // num = num * 2;
System.out.println(num);  //     : 16
```

2.3 การเพิ่มและลดค่า (Increment & Decrement Operators)

```
int x = 1;

// Post-Increment (x++             )
System.out.println(x++);  //     : 1
System.out.println(x);    //     : 2

// Pre-Increment (++x              )
System.out.println(++x);  //     : 3
```

2.4 ตัวดำเนินการเปรียบเทียบ (Relational Operators)

```
int a = 10;
int b = 5;

System.out.println(a == b);   //     : false
System.out.println(a != b);   //     : true
System.out.println(a > b);    //     : true
System.out.println(a <= b);   //     : false
```

2.5 ตัวดำเนินการแบบเงื่อนไข (Conditional Operator) *

```
int n = 3;
int next = (n % 2 == 0) ? (n / 2) : (3 * n + 1);
System.out.println(next);  //     : 10
```

2.6 การแปลงชนิดข้อมูล (Type Casting) *

```
int a = 10;
double b = a;  //      int    double

double c = 9.99;
int d = (int) c;  //      double    int       cast
System.out.println(d);  //     : 9
```

ข้อควรระวัง: การแปลงจาก `double` เป็น `int` จะตัดทศนิยมทิ้ง

2.7 การใช้แพ็กเกจใน Java (Java Packages) *

- **ตัวอย่างการนำเข้าแพ็กเกจ**

```
import java.util.*; //       *        packages

public class InputExample {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter your name: ");
        String name = input.nextLine();
        System.out.println("Hello, " + name);
        input.close();
    }
}
```

2.8 การจัดการเอกสารโค้ดด้วย Javadoc *

- **ตัวอย่าง Javadoc Comment**

```
/**
 * This class demonstrates how to use Javadoc.
 * @author Oh
 * @version 1.0
 */
public class JavadocExample {
```

```
    /**
     * This method adds two numbers.
     * @param a the first number
     * @param b the second number
     * @return the sum of a and b
     */
    public static int add(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {
        System.out.println("Sum: " + add(5, 10));
    }
}
```

Contents

ความยาก: **

เนื้อหาต่อจากนี้ถูกสร้างขึ้นโดย AI โดยใช้ข้อมูลจาก Presentation Slides เรื่อง Methods

---

2.1 ตัวอย่างโครงสร้าง Method พื้นฐาน

```
public class MethodsExample {
    // Method    add
    public static int add(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {
        int result = add(5, 10);
        System.out.println("Result: " + result);  //     : 15
    }
}
```

2.2 ตัวอย่างการประกาศและนิยาม Method

```
//       Method
static int max(int num1, int num2);

//      Method
static int max(int num1, int num2) {
    if (num1 > num2) {
        return num1;
    } else {
        return num2;
    }
}
```

2.3 การเรียกใช้งาน Method (Method Call)

```
public static void main(String[] args) {
    int i = 5;
    int j = 2;
    int result = max(i, j);
    System.out.println("Max value is: " + result);  //     : Max value is: 5
}
```

2.4 การส่งค่าพารามิเตอร์ (Passing Parameters)

```
// Method
static void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
    System.out.println("Inside swap method: a = " + a + ", b = " + b);
}

public static void main(String[] args) {
    int m = 2;
    int n = 3;
    System.out.println("Before swap: m = " + m + ", n = " + n);
    swap(m, n);
    System.out.println("After swap: m = " + m + ", n = " + n);  //
}
```

2.5 การ Overloading Method **

```
//    Overloading Method
static int max(int num1, int num2) {
    return (num1 > num2) ? num1 : num2;
}

static double max(double num1, double num2) {
    return (num1 > num2) ? num1 : num2;
}

public static void main(String[] args) {
    System.out.println(max(5, 10));        //     : 10
    System.out.println(max(5.5, 10.1));    //     : 10.1
}
```

2.6 Method ที่เรียกตัวเอง (Recursive Method) *

```
//      Factorial       Recursive Method
static long factorial(int n) {
    if (n == 0) { //
        return 1;
    } else {
        return n * factorial(n - 1);
    }
}

public static void main(String[] args) {
    int number = 5;
    System.out.println("Factorial of " + number + " is " + factorial(number));  //     : 120
}
```

2.6.1 Recursive Method Stopping Condition

ถ้า Recursive Method ไม่มีเงื่อนไขหยุด จะทำให้เกิด **StackOverflowError**

2.7 การใช้งาน Method จากแพ็กเกจมาตรฐาน (API) *

```
import java.util.Scanner;

public class InputExample {
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = input.nextInt();
        System.out.println("You entered: " + number);
        input.close();
    }
}
```

Contents

ความยาก: *

เนื้อหาต่อจากนี้ถูกสร้างขึ้นโดย AI โดยใช้ข้อมูลจาก Presentation Slides เรื่อง Control Statements

---

2.1 if Statement

```
int number = 10;
if (number > 0) {
    System.out.println("The number is positive.");  //     : The number is positive.
}
```

2.2 if...else Statement

```
int number = -5;
if (number > 0) {
    System.out.println("The number is positive.");
} else {
    System.out.println("The number is negative.");  //     : The number is negative.
}
```

2.3 Nested if Statement

```
int score = 85;

if (score >= 90) {
    System.out.println("Grade: A");
} else if (score >= 80) {
    System.out.println("Grade: B");  //     : Grade: B
} else if (score >= 70) {
    System.out.println("Grade: C");
} else {
    System.out.println("Grade: F");
}
```

2.4 switch Statement **

```
int day = 3;

switch (day) {
    case 1:
        System.out.println("Monday");
        break;
    case 2:
        System.out.println("Tuesday");
        break;
    case 3:
        System.out.println("Wednesday");  //     : Wednesday
        break;
    default:
        System.out.println("Invalid day");
}
```

2.5 while Loop

```
int i = 0;
while (i < 5) {
    System.out.println("Count: " + i);
    i++;
}
//     :
// Count: 0
// Count: 1
// Count: 2
// Count: 3
// Count: 4
```

2.6 do...while Loop

```
int i = 0;
do {
    System.out.println("Count: " + i);
    i++;
} while (i < 5);
//           while loop
```

2.7 for Loop

```
for (int i = 1; i <= 5; i++) {
    System.out.println("Iteration: " + i);
}
//     :
// Iteration: 1
// Iteration: 2
// Iteration: 3
// Iteration: 4
// Iteration: 5
```

2.8 Nested for Loop

```
for (int i = 1; i <= 2; i++) {
    for (int j = 1; j <= 4; j++) {
        System.out.print(j + " ");
    }
    System.out.println();
}
//     :
// 1 2 3 4
// 1 2 3 4
```

2.9 break Statement *

```
for (int i = 1; i <= 5; i++) {
    if (i == 3) {
        break;
    }
    System.out.println("Iteration: " + i);
}
//     :
// Iteration: 1
// Iteration: 2
```

2.10 continue Statement *

```
for (int i = 1; i <= 5; i++) {
    if (i == 3) {
        continue;
    }
    System.out.println("Iteration: " + i);
}
//     :
// Iteration: 1
// Iteration: 2
// Iteration: 4
// Iteration: 5
```

2.11 การใช้ Scanner รับค่าจากผู้ใช้ **

```
import java.util.Scanner;

public class UserInput {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter your age: ");
        int age = scanner.nextInt();

        if (age >= 18) {
            System.out.println("You are an adult.");
        } else {
            System.out.println("You are not an adult.");
        }

        scanner.close();
    }
}
```

Contents

1 Java

2 สรุปเนื้อหาเรื่อง Arrays (พร้อมตัวอย่างโค้ด)

ความยาก: **

เนื้อหาต่อจากนี้ถูกสร้างขึ้นโดย AI โดยใช้ข้อมูลจาก Presentation Slides เรื่อง Arrays

---

2.1 การสร้างและใช้งาน Array

```
//     Array         5
int[] numbers = new int[5];

//       Array
numbers[0] = 10;
numbers[1] = 20;
numbers[2] = 30;
numbers[3] = 40;
numbers[4] = 50;

//
for (int i = 0; i < numbers.length; i++) {
    System.out.println("Element " + i + ": " + numbers[i]);
}
```

2.2 Array Initialization (การกำหนดค่าเริ่มต้น)

```
{English Text}
//                Array
int[] list = {1, 2, 3, 4, 5};

//
for (int num : list) {
    System.out.println(num);
}
```

2.3 Frequents Errors About Array *

หากพยายามเข้าถึง index ที่ไม่มีใน array จะเกิดข้อผิดพลาด `ArrayIndexOutOfBoundsException`

```
int[] list = {1, 2, 3};
System.out.println(list[3]);  //     ArrayIndexOutOfBoundsException
```

2.4 Property `length` ใน Array หลายมิติ

- `A.length` ให้จำนวนแถวของ array
- `A[0].length` ให้จำนวนคอลัมน์ในแถวแรก

```
int[][] A = {
    {1, 2, 3, 4},
    {5, 6, 7, 8},
    {9, 10, 11, 12}
};

System.out.println("Number of rows: " + A.length);        //     : 3
System.out.println("Number of columns in row 0: " + A[0].length);  //     : 4
```

2.5 Multi-dimensional Arrays (Array หลายมิติ) *

```
//     2D Array (3    4     )
int[][] matrix = {
    {1, 2, 3, 4},
    {5, 6, 7, 8},
    {9, 10, 11, 12}
};

//
for (int i = 0; i < matrix.length; i++) {
    for (int j = 0; j < matrix[i].length; j++) {
        System.out.print(matrix[i][j] + " ");
    }
    System.out.println();
}
```

2.6 การค้นหาค่าใน Array (Linear Search) *

```
public class LinearSearch {
    public static int search(int[] array, int target) {
        for (int i = 0; i < array.length; i++) {
            if (array[i] == target) {
                return i;  //     index
            }
        }
        return -1;  //     -1
    }

    public static void main(String[] args) {
        int[] numbers = {10, 20, 30, 40, 50};
        int index = search(numbers, 30);
```

```
        System.out.println("Found at index: " + index);  //     : Found at index: 2
    }
}
```

2.7 การคัดลอกค่าใน Array (Array Copy) *

- **วิธีที่ 1: การคัดลอกแบบ Manual**

```
int[] source = {1, 2, 3, 4, 5};
int[] destination = new int[source.length];

//
for (int i = 0; i < source.length; i++) {
    destination[i] = source[i];
}

//
for (int num : destination) {
    System.out.println(num);
}
```

- **วิธีที่ 2: การคัดลอกด้วย System.arraycopy()**

ความยาก: *

```
int[] source = {1, 2, 3, 4, 5};
int[] destination = new int[source.length];

//    System.arraycopy()
System.arraycopy(source, 0, destination, 0, source.length);

//
for (int num : destination) {
    System.out.println(num);
}
```

2.8 การเรียงลำดับค่าใน Array (Sorting) **

- การเรียงลำดับค่าใน Array ด้วย Selection Sort

ความยาก: **

```
public class SelectionSort {
    public static void sort(int[] array) {
        for (int i = 0; i < array.length - 1; i++) {
            int minIndex = i;
            for (int j = i + 1; j < array.length; j++) {
                if (array[j] < array[minIndex]) {
                    minIndex = j;
                }
            }
            int temp = array[minIndex];
            array[minIndex] = array[i];
            array[i] = temp;
        }
    }

    public static void main(String[] args) {
        int[] numbers = {64, 25, 12, 22, 11};
        sort(numbers);
        for (int num : numbers) {
            System.out.print(num + " ");  //     : 11 12 22 25 64
        }
    }
}
```

- **การเรียงลำดับด้วย Insertion Sort**

ความยาก: **

```
public class InsertionSort {
    public static void sort(int[] array) {
        for (int i = 1; i < array.length; i++) {
            int key = array[i];
            int j = i - 1;

            while (j >= 0 && array[j] > key) {
                array[j + 1] = array[j];
                j--;
            }
            array[j + 1] = key;
        }
    }

    public static void main(String[] args) {
        int[] numbers = {5, 2, 9, 1, 5, 6};
        sort(numbers);

        for (int num : numbers) {
            System.out.print(num + " ");  //     : 1 2 5 5 6 9
        }
    }
}
```

2.9 การค้นหาด้วย Binary Search (Binary Search) *

```
public class BinarySearch {
    public static int binarySearch(int[] array, int target) {
        int low = 0;
        int high = array.length - 1;

        while (low <= high) {
            int mid = (low + high) / 2;

            if (array[mid] == target) {
                return mid;  //     index
            } else if (array[mid] < target) {
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }
        return -1;  //     -1
```

```
    }

    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9};
        int index = binarySearch(numbers, 5);
        System.out.println("Found at index: " + index);  //     : Found at index: 4
    }
}
```

2.10 การใช้ Arrays Utility ใน Java *

```
import java.util.Arrays;

public class ArrayUtilityExample {
    public static void main(String[] args) {
        int[] numbers = {5, 2, 9, 1, 5, 6};

        //       Array
        Arrays.sort(numbers);

        //
        System.out.println(Arrays.toString(numbers));  //     : [1, 2, 5, 5, 6, 9]

        //       Binary Search
        int index = Arrays.binarySearch(numbers, 5);
        System.out.println("Found at index: " + index);  //     : 2
    }
}
```
