

Il Modulo

Concetto Teorico di Modulo

In matematica e programmazione, il modulo si riferisce a:

1. **Valore assoluto:**

- In matematica, il modulo di un numero è la sua distanza dallo zero.

Operatore modulo (%):

- In programmazione, il modulo restituisce il **resto** di una divisione tra due numeri. Si utilizza quando vogliamo sapere quanto "avanza" dopo aver diviso un numero per un altro.

Utilizzo del Modulo negli Algoritmi

Il modulo è uno strumento potente per risolvere problemi che coinvolgono cicli, o verifiche specifiche.

Algoritmi Basati sul Modulo

Algoritmo 1: Verifica Pari o Dispari

1. Leggi un numero intero `n`.
2. Se `n % 2 == 0`, stampa "Il numero è pari".
3. Altrimenti, stampa "Il numero è dispari".

Algoritmo 2: Numeri Multipli di un Numero

1. Leggi un numero `m`.
2. Itera da 1 a un limite massimo (es., 50).
3. Se `i % m == 0`, stampa "`i` è multiplo di `m`".

Gli Operatori Booleani

Concetto Teorico di Operatori Booleani

Gli **operatori booleani** permettono di combinare o modificare condizioni logiche (vero/falso). Sono fondamentali per la programmazione, specialmente nei costrutti condizionali e nei cicli.

Tipi di Operatori Booleani

1. **AND (&&):**
 - Restituisce **true** se **entrambe** le condizioni sono vere.
 - Esempio: $A \ \&\& \ B$ è vero solo se AAA e BBB sono entrambi veri.
2. **OR (||):**
 - Restituisce **true** se **almeno una** delle condizioni è vera.
 - Esempio: $A \ || \ BA \ || \ BA \ || \ B$ è vero se AAA, BBB, o entrambi sono veri.
3. **NOT (!):**
 - Inverte il valore booleano.
 - Esempio: $!A!A!A$ è vero se AAA è falso.

Applicazioni Negli Algoritmi

Esempio 1: Controllo di Più Condizioni Un numero è positivo **e** pari:

1. Leggi un numero n .
2. Se $n > 0$ e $n \% 2 == 0$, stampa "Positivo e pari".
3. Altrimenti, stampa "Non soddisfa entrambe le condizioni".

Esempio 2: Verifica Alternativa Un numero è negativo **o** maggiore di 100:

1. Leggi un numero n .
2. Se $n < 0$ o $n > 100$, stampa "Condizione soddisfatta".
3. Altrimenti, stampa "Condizione non soddisfatta".

Implementazione in Java

Operatore Modulo (%) in Java

L'operatore % è integrato in Java e funziona su interi e numeri in virgola mobile.

Esempio: Verifica Pari o Dispari

```
public class ModuloExample {  
    public static void main(String[] args) {  
        int numero = 7;  
  
        if (numero % 2 == 0) {  
            System.out.println(numero + " è pari.");  
        } else {  
            System.out.println(numero + " è dispari.");  
        }  
    }  
}
```

Spiegazione:

- La condizione `numero % 2 == 0` verifica se il resto della divisione del numero per 2 è uguale a 0.
 - Se il resto è 0, il numero è divisibile per 2 ed è quindi pari.
 - Se il resto non è 0, il numero non è divisibile per 2 ed è quindi dispari.

Operatori Booleani in Java

Gli operatori booleani (&&, ||, !) sono utilizzati per combinare condizioni.

Esempio: Controllo Multiplo

```
public class BooleanExample {  
    public static void main(String[] args) {  
        int numero = 8;  
  
        if (numero > 0 && numero % 2 == 0) {  
            System.out.println(numero + " è positivo e pari.");  
        } else {  
            System.out.println("Il numero non soddisfa entrambe le  
condizioni.");  
        }  
    }  
}
```

Spiegazione:

- La condizione `numero > 0 && numero % 2 == 0` verifica che il numero sia:
 1. Maggiore di 0.
 2. Pari (cioè divisibile per 2 senza resto).
- Solo se entrambe le condizioni sono vere, viene stampato: *"8 è positivo e pari."*
- In caso contrario, verrà stampata la frase nel ramo `else`.

Esempio: Negazione

```
public class Negazione {  
    public static void main(String[] args) {  
        boolean condizione = false;  
  
        if (!condizione) {  
            System.out.println("La condizione è falsa, quindi il NOT  
la rende vera.");  
        } else {  
            System.out.println("La condizione era vera.");  
        }  
    }  
}
```

Spiegazione:

- L'operatore **!** inverte il valore booleano di **condizione**.
 - Se **condizione** è **false**, **!condizione** diventa **true**.
 - Se **condizione** è **true**, **!condizione** diventa **false**.
- In questo caso, dato che **condizione** è inizializzato a **false**, il ramo **if** viene eseguito e stampa il messaggio: *"La condizione è falsa, quindi il NOT la rende vera."*

Esempio: Controllo Alternativo (||)

```
public class OperatoreOr {  
    public static void main(String[] args) {  
        int numero = -3;  
  
        if (numero < 0 || numero % 2 == 0) {  
            System.out.println(numero + " è negativo oppure pari.");  
        } else {  
            System.out.println("Il numero non soddisfa nessuna delle  
due condizioni.");  
        }  
    }  
}
```

Spiegazione:

- L'operatore `||` verifica se **almeno una delle due condizioni** è vera:
 - `numero < 0`: controlla se il numero è negativo.
 - `numero % 2 == 0`: controlla se il numero è pari.
- **Comportamento:**
 - Se **una o entrambe** le condizioni sono vere, il ramo `if` viene eseguito.
 - Altrimenti, si passa al ramo `else`.