

Array - while - for

Leggere i dati in input

Per leggere i dati in input in Java, si utilizza l'oggetto `Scanner`. Prima di tutto, bisogna importare la libreria necessaria e poi creare un'istanza di `Scanner`:

```
import java.util.Scanner;

Scanner in = new Scanner(System.in);
```

Vettori

Creazione del Vettore

Per creare un vettore di interi con 10 elementi:

```
int[] arr = new int[10];
```

Accesso agli Elementi

- **Indice:** Utilizzare `i` per accedere all'indice dell'elemento.
- **Valore:** Utilizzare `arr[i]` per accedere al valore dell'elemento.

Per accedere o iterare sugli elementi del vettore, si usa un ciclo:

```
for(int i = 0; i < arr.length; i++){
    // operazioni sugli elementi
}
```

Riempire il Vettore

Per riempire un vettore con `N` elementi letti da input:

```
for(int i = 0; i < N; i++){  
    arr[i] = in.nextInt();  
}
```

Stampa degli Elementi

Per stampare tutti gli elementi del vettore:

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

Stampa in Ordine Inverso

Per stampare gli elementi di un vettore in ordine inverso (esempio con 15 elementi):

```
int[] inverso = new int[15];  
for(int i = 0; i < 15; i++){  
    inverso[14 - i] = arr[i];  
}  
  
for(int i = 0; i < 15; i++){  
    System.out.println("Elemento inverso " + i + ": " + inverso[i]);  
}
```

Calcolo della Media degli Elementi

Per calcolare la media degli elementi di un vettore:

```
int somma = 0;  
for(int i = 0; i < arr.length; i++){  
    somma += arr[i];  
}  
  
int media = somma / arr.length;  
System.out.println("La media degli elementi è: " + media);
```

Condizioni

Numeri Dispari

Per verificare se un numero è dispari:

```
if(arr[i] % 2 == 1){  
    // il numero è dispari  
}  
  
// oppure  
  
if(arr[i] % 2 != 0){  
    // il numero è dispari  
}
```

Numeri Pari

Per verificare se un numero è pari:

```
if(arr[i] % 2 == 0){  
    // il numero è pari  
}
```

Differenza tra while e for

Ciclo while

Il ciclo `while` viene utilizzato quando si vuole iterare fino a quando una condizione rimane vera. Non è necessario conoscere il numero esatto di iterazioni in anticipo.

```
while(condizione){  
    // blocco di codice  
}
```

Ciclo for

Il ciclo `for` viene utilizzato quando si conosce il numero esatto di iterazioni. È ideale per iterare attraverso array o liste con una lunghezza definita.

```
for(inizializzazione; condizione; incremento){  
    // blocco di codice  
}
```

```
}
```

Esempi di Utilizzo

- **Ciclo `while`** : utile quando si attende un input specifico dall'utente.

```
int input;
while((input = in.nextInt()) != -1){
    // continua a leggere input fino a quando l'utente inserisce -1
}
```

- **Ciclo `for`** : utile per iterare su un array o una lista con una lunghezza nota.

```
for(int i = 0; i < arr.length; i++){
    System.out.println(arr[i]);
}
```
