Il pattern **Singleton** è un modello di progettazione creazionale che assicura che una classe abbia una sola istanza e fornisce un punto di accesso globale a quell'istanza. È utile quando è necessario un oggetto globale che controlla risorse condivise o mantiene uno stato condiviso.

Componenti del Pattern Singleton

1. **Istanza Unica:** La classe singleton ha una sola istanza, e questa istanza è accessibile globalmente.
2. **Costruttore Privato:** Il costruttore della classe è privato, impedendo la creazione di nuove istanze dall'esterno.
3. **Metodo di Accesso Statico:** Un metodo statico fornisce un punto di accesso all'istanza unica della classe.

Come Funziona

- **Istanziazione Pigra (Lazy Initialization):** L'istanza viene creata solo quando è necessaria per la prima volta.
- **Istanziazione Eager (Eager Initialization):** L'istanza viene creata subito quando la classe viene caricata.

Esempio con Pokémon

Supponiamo di voler creare un registro di battaglie Pokémon che tenga traccia di tutte le battaglie combattute. Utilizzeremo il pattern Singleton per assicurare che esista un solo registro di battaglie.

Implementazione Singleton

Registro di Battaglia

java — Copia codice

```java
public class RegistroBattaglie {
    // L'unica istanza dell'oggetto RegistroBattaglie
    private static RegistroBattaglie instance;

    // Costruttore privato per impedire l'istanza diretta
    private RegistroBattaglie() {}

    // Metodo statico per ottenere l'istanza unica
    public static synchronized RegistroBattaglie getInstance() {
        if (instance == null) {
            instance = new RegistroBattaglie();
        }
        return instance;
    }

    // Metodo per registrare una battaglia
    public void registraBattaglia(String battaglia) {
        System.out.println("Registrazione della battaglia: " + battaglia);
    }
}
```

Applicazione Principale

Utilizzo del Singleton

java — Copia codice

```java
public class MainApp {
    public static void main(String[] args) {
        // Ottenere l'unica istanza del RegistroBattaglie
        RegistroBattaglie registro1 = RegistroBattaglie.getInstance();
        RegistroBattaglie registro2 = RegistroBattaglie.getInstance();

        // Verifica che entrambe le variabili puntino alla stessa istanza
        if (registro1 == registro2) {
            System.out.println("RegistroBattaglie è un singleton: entrambe le variabili puntano alla stessa istanza.");
        }

        // Registrare alcune battaglie
        registro1.registraBattaglia("Pikachu vs Charmander");
        registro2.registraBattaglia("Bulbasaur vs Squirtle");
    }
}
```

Come Funziona Questo Esempio

1. **Costruttore Privato:** Il costruttore del `RegistroBattaglie` è privato, quindi non può essere chiamato dall'esterno per creare nuove istanze.
2. **Istanza Statica:** La variabile `instance` è una variabile statica che mantiene l'unica istanza del `RegistroBattaglie`.
3. **Metodo Statico:** Il metodo `getInstance` fornisce un punto di accesso globale all'unica istanza del `RegistroBattaglie`. Utilizza la sincronizzazione (`synchronized`) per garantire che l'istanza sia creata in modo thread-safe.

Benefici del Pattern Singleton

- **Accesso Controllato:** Assicura che ci sia una sola istanza di una classe.
- **Punto di Accesso Globale:** Fornisce un punto di accesso globale a quell'istanza.
- **Gestione delle Risorse:** Può essere utilizzato per gestire risorse condivise come connessioni a database, file di configurazione, ecc.

Quando Utilizzare il Pattern Singleton

- Quando è necessario assicurarsi che ci sia una sola istanza di una classe.
- Quando l'oggetto singleton deve essere accessibile globalmente.
- Quando è necessario un punto di accesso controllato per risorse condivise.

Conclusione

Il pattern Singleton è un modello semplice ma potente che garantisce l'esistenza di una sola istanza di una classe, fornendo un punto di accesso globale a essa. Nell'esempio del registro di battaglie Pokémon, il Singleton assicura che tutte le battaglie siano registrate nello stesso oggetto, facilitando la gestione e la consultazione delle battaglie combattute.