

3.1) Quali oggetti vengono istanziati? Quali di questi hanno un'istanza esplicita?

Vengono creati cinque oggetti: Scanner, Posto, Display, Partecipante e Scrittore.

- Scanner è un'istanza esplicita, usata per leggere l'input dell'utente.
- Posto è rappresentato come un array che indica le sedie del gioco, ma non è un'istanza esplicita.
- Display è un'istanza esplicita che mostra il risultato del gioco.
- Partecipante è un'istanza esplicita che rappresenta i giocatori.
- Scrittore è un'istanza esplicita che serve per salvare i risultati su un file.

3.2) Esistono classi anonime? Definisci il concetto di classe anonima.

Non ci sono classi anonime in questo caso. Una classe anonima è una classe che viene definita e creata nello stesso momento. Spesso viene usata per implementare interfacce o estendere classi in modo rapido.

3.3) Quali sono le classi che lavorano in concorrenza? Quanti sono i thread in memoria?

Le classi che lavorano in concorrenza sono Partecipante e Display.

Il numero totale di thread dipende dal numero di partecipanti: c'è un thread per ogni sedia ($\text{NUMSEDIE} + 1$) e un thread per il Display.

3.4) Quali sono i metodi della classe Thread che vengono usati?

I metodi utilizzati sono start() e sleep(millis).

- start() avvia il thread e richiama il metodo run().
- sleep(millis) ferma temporaneamente l'esecuzione del thread per un certo periodo di tempo.

3.5) Qual è la classe Monitor? Perché e a cosa serve?

La classe Posto funziona come Monitor. Questo significa che usa il meccanismo synchronized per assicurarsi che solo un thread alla volta possa accedere a risorse condivise, come le sedie. Serve per evitare conflitti tra thread nella sezione critica.

3.6) Quali eccezioni sono usate nell'applicazione? Di che tipo sono?

L'eccezione InterruptedException, che è un'eccezione controllata, viene gestita nel metodo sleep().

3.7) Come si gestiscono le eccezioni? Individua le parti di codice interessate.

Le eccezioni si dividono in controllate e non controllate:

- Le eccezioni controllate devono essere gestite con un blocco try-catch, altrimenti il codice non viene compilato. Il blocco try contiene il codice "a rischio", mentre catch gestisce eventuali errori.
- Le eccezioni non controllate non sono obbligatorie da gestire, ma possono causare errori durante l'esecuzione (es. divisione per zero).

