

# Altri Paradigmi di Programmazione

---

1° Appello - 26 Gennaio 2021

---

1) Quale di queste caratteristiche è propria della sintassi `switch-case` come espressione:

- ☐ I risultati devono essere tutti valori della stessa interfaccia.
- ☐ E' possibile il fall-through da un caso all'altro.
- ☒ L'elenco delle opzioni deve essere esaustivo.
- ☐ Ogni caso deve produrre un risultato diverso.

2) Se in un sistema distribuito i nodi non trovano un consenso sullo stato del sistema, può accadere che:

- ☒ Le risposte del sistema siano incoerenti e dipendano da quale nodo viene contattato.
- ☐ Le risposte del sistema siano molteplici e conflittuali perché raccolgono i dati da più nodi.
- ☐ Le risposte del sistema siano inefficienti perché le differenti versioni dello stato si accavallano in una `race condition`.
- ☐ Le risposte del sistema non siano disponibili in quanto gli stati differenti si annullano.

3) Lo scopo delle Reactive Extensions è:

- ☐ Fornire una API per definire elaborazioni di sequenze di oggetti.
- ☒ Fornire una semantica per definire elaborazioni asincrone di sequenze di oggetti.
- ☐ Fornire un insieme di componenti per l'elaborazione distribuita di stream di valori.
- ☐ Fornire un modello di esecuzione di elaborazioni parallele di insiemi di oggetti.

4) Quale delle seguenti affermazioni riguardante le Lambda Expression è vera:

- ☐ Rendono Java un linguaggio funzionale perché possono essere passate come parametri di una chiamata ad un metodo.
- ☒ Non rendono Java un linguaggio funzionale perché non sono una entità del linguaggio, ma solo una convenienza sintattica risolta dal compilatore.
- ☐ Non rendono Java un linguaggio funzionale perché non sono facilmente componibili.
- ☐ Rendono Java un linguaggio funzionale perché permettono di astrarre sul comportamento di un metodo chiamato.

5) Nelle Reactive Extensions, quali di queste operazioni *non* è necessario (o possibile) specificare per elaborare gli oggetti emessi da un `Observable`:

- ☐ Il comportamento alla ricezione di un oggetto.
- ☐ Il comportamento al termine dello stream di oggetti.
- ☒ Il numero di oggetti che l'`Observable` è autorizzato ad inviare.
- ☒ Il comportamento alla richiesta di separazione di uno stream parallelo.
- ☐ Il comportamento alla ricezione di un errore.

## 6) Nella implementazione degli Stream della libreria standard, che vantaggio si ottiene dal fatto che la API consente di costruire la catena di elaborazione separatamente dalla sua esecuzione?

- ☒ L'implementazione può analizzare le operazioni della catena, e prendere decisioni su come applicarle in funzione delle loro caratteristiche.
- ☐ L'implementazione può sempre sapere se dovrà eseguire un numero finito o meno di elaborazioni in funzione unicamente delle operazioni intermedie.
- ☐ L'implementazione può analizzare le operazione della catena, e decidere se eseguirle parallelamente o in serie.

## 7) La fallacia "Latency is zero" è ancora rilevante perché:

- ☐ L'aumento dei nodi della rete ha compensato il miglioramento tecnologico, mantenendo il problema sostanzialmente uguale.
- ☐ Le tecnologie di comunicazione hanno eliminato il problema presente in passato, ma le esigenze di concorrenza l'hanno reintrodotto in altra forma.
- ☒ Dipende da una grandezza fisica

## 8) Che tipo di vantaggi si possono avere dall'adottare un framework per la costruzioni di applicazioni distribuite?

- ☒ Maggiore sicurezza perché i dettagli sono gestiti da persone più competenti.
- ☒ Aggiornamento continuo che apporta benefici a tutte le parti dell'applicazione, in modo quasi automatico.
- ☐ Estrema efficienza nello sfruttare le peculiarità dell'hardware a disposizione.
- ☐ Assistenza remota nella risoluzione degli errori applicativi.
- ☒ Facilità di realizzazione perché i dettagli dei protocolli di comunicazione sono nascosti da API di livello più elevato.
- ☐ Aggiornamento continuo che non richiede intervento da parte dello sviluppatore.
- ☒ Facilità di realizzazione perché le parti più strutturali sono già implementate.

## 9) L'interfaccia `Collector` permette di eseguire la riduzione ad un risultato di uno Stream parallelo in modo più efficiente perché:

- ☐ Perché non necessita di sapere la lunghezza dello Stream.
- ☒ Perché gestisce un accumulatore mutabile, che riduce la pressione sulla Garbage Collection.
- ☐ Perché combina i risultati intermedi più velocemente.

- ☐ Perché mantiene il parallelismo dello Stream.

## 10) Una operatore *short-circuiting* all'interno di una catena di elaborazione di uno Stream può:

- ☐ Cambiare l'ordine degli elementi dello Stream.
- ☒ Produrre il risultato prima che lo Stream sia stato interamente consumato.
- ☐ Rendere seriale l'elaborazione di uno Stream parallelo.
- ☐ Ottenere uno Stream infinito da una funzione di trasformazione.

## 11) Un Thread esce dallo stato `timed waiting` quando:

- ☐ Trascorre esattamente il tempo impostato (e solo in questo caso).
- ☐ Ottiene il lock che stava aspettando o viene interrotto.
- ☒ Ottiene il lock che stava aspettando, oppure viene interrotto o trascorre il timeout impostato.
- ☐ Viene interrotto (e solo in questo caso)

## 12) In reazione alla ricezione di un messaggio, un attore può:

- ☒ Creare nuovi attori.
- ☒ Modificare il suo stato interno.
- ☒ Inviare messaggi ad attori di cui ha un riferimento.
- ☐ Modificare lo stato di un attore di cui ha un riferimento.
- ☐ Inviare un messaggio ad un attore di un altro nodo del sistema.
- ☐ Eliminare un attore di cui ha un riferimento.
- ☒ Modificare il suo comportamento per la ricezione dei prossimi messaggi

## 13) Usando i Reactive Stream, la gestione più granulare della composizione della pipeline di elaborazione dello stream permette di:

- ☐ Ridurre la latenza dell'elaborazione dei vari componenti decidendo quante risorse dedicare a ciascuno di essi.
- ☒ Isolare la parte di pipeline che si desidera sia resa parallela; gli Stream della libreria standard sono o completamente paralleli, o completamente seriali.
- ☐ Distribuire i singoli componenti dell'elaborazione su più nodi, indicando su quali nodi aumentare il parallelismo e su quali accumulare i risultati da elaborare serialmente.
- ☐ Scegliere algoritmi di suddivisione del lavoro più efficienti di quelli della libreria standard, perché più facili da aggiornare

## 14) Nel linguaggio Java, una variabile `final`:

- ☐ Richiama un metodo al momento della cancellazione.
- ☐ Richiama un metodo quando viene modificata.
- ☒ Deve essere inizializzata contestualmente alla definizione, ed il suo valore non può essere cambiato.

- ☐ Può contenere un valore di un solo tipo

## 15) Implementare un programma di rete usando l'astrazione dei `Channels` della libreria standard di Java permette di non occuparsi di molti dettagli riguardanti l'interazione con il mezzo di comunicazione, ma:

- ☐ E' sufficiente sostituire il codice che gestisce la ricezione di un pacchetto di dati.
- ☐ E' necessario riscrivere le parti che interagiscono con il mezzo di comunicazione per gestire in modo diverso la concorrenza.
- ☐ E' sufficiente sostituire il codice che gestisce una nuova connessione entrante.
- ☒ E' necessario ristrutturare il nostro codice riorganizzando in metodi che vengono richiamati all'avvenire di specifici eventi di I/O

## 16) Quando si dice che la parola chiave `synchronized` introduce una relazione di *happens-before* nel codice, si intende che:

- ☐ Il compilatore viene istruito a garantire che il codice sorvegliato dalla parola chiave `synchronized` venga eseguito contemporaneamente al codice che viene collegato dall'argomento dell'espressione.
- ☒ Il compilatore viene istruito a garantire che il codice sorvegliato dalla parola chiave `synchronized` venga effettivamente eseguito prima del codice che lo segue, e da un solo Thread alla volta.
- ☐ Il compilatore viene istruito a garantire che il codice sorvegliato dalla parola chiave `synchronized` venga effettivamente eseguito dopo il codice che lo precede, indipendentemente da quanti Thread lo attraversino.
- ☐ Il compilatore viene istruito a garantire che il codice sorvegliato dalla parola chiave `synchronized` venga allineato al trasferimento di dati fra la memoria principale e la cache del microprocessore.

## 17) Le classi del package `java.concurrent.atomic`:

- ☐ Sono particolarmente efficienti in caso di modifica concorrente del dato che rappresentano perché usano nel modo migliore i lock.
- ☐ Non sono adatte nel caso di modifica concorrente perché permettono una sola modifica alla volta.
- ☐ Non sono adatte nel caso di modifica concorrente perché permettono di leggere un dato che in realtà è già stato modificato.
- ☒ Sono particolarmente efficienti in caso di modifica concorrente del dato che rappresentano perché usano (se disponibili) delle funzionalità fornite direttamente dall'hardware.

## 18) Quale delle seguenti affermazioni riguardo ai rapporti fra Processi, Thread e Fiber è vera:

- ☒ Le risorse dei Processi sono controllate dal Sistema Operativo, mentre all'interno dei Processi i Thread devono direttamente controllare il loro accesso. Le Fiber rendono esplicita la concorrenza con lo scopo di essere ancora più leggere dei Thread.

- ☐ I Processi sono raggruppamenti logici di risorse che nei Thread vengono associati a risorse fisiche. Le Fiber sono un miglioramento dei Thread.
- ☐ Una volta allocata una risorsa, non può essere sottratta ad un Processo. Ad un Thread invece può essere sottratta, mettendolo in stato waiting. Le Fiber rendono la gestione della concorrenza più esplicita.
- ☐ I Processi evitano il deadlock attraverso l'ordinamento delle priorità. I Thread invece non sono ordinati e devono essere manualmente controllati per evitare conflitti nella gestione delle risorse. Le Fiber sono una evoluzione più efficiente dei Thread.

### 19) Una classe Java dichiarata ***abstract*** è visibile:

- ☐ Da qualsiasi classe.
- ☐ Solo dalle classe che la estendono.
- ☐ Da tutte le classi dello stesso package.
- ☒ **abstract** non è un modificatore di visibilità.

### 20) Quale delle seguenti frasi è falsa per una struttura dati non Thread-Safe in caso di accesso concorrente:

- ☐ Può dare un risultato errato o venirsi a trovare in uno stato inconsistente
- ☐ Può lanciare un'eccezione per segnalare un accesso non consentito o pericoloso
- ☒ E' meno costosa in termini di cicli di CPU che nel caso di accesso esclusivo.
- ☐ E' più performante nel caso generale

### 21) Quali vantaggi si cercano nel distribuire lo stato di un sistema su più nodi:

- ☐ Maggiore sicurezza dei dati gestiti dal sistema di consenso.
- ☒ Disponibilità anche in caso di guasti.
- ☒ Accesso più rapido da località differenti.
- ☐ Elaborazione più rapida delle richieste distribuite a più nodi

### 22) Indicare quali fra le seguenti sono problematiche che rendono la serializzazione un processo complesso, che richiede molte attenzioni:

- ☐ E' necessario disporre di un metodo per riordinare le parti del messaggio ricevute.
- ☒ Mittente e ricevente possono avere versioni diverse dell'oggetto serializzato.
- ☒ E' necessario disporre di un metodo per verificare integrità e affidabilità dei dati ricevuti.
- ☒ Il processo di serializzazione deve essere efficiente nel tempo e nello spazio impiegati.
- ☐ E' necessario disporre di un protocollo che delimiti correttamente le varie parti del messaggio.
- ☒ Un oggetto può contenere altri oggetti, che potrebbero non essere rappresentabili

### 23) Quando si dice che il compilatore Java ha delle capacità di Type Inference si intende che:

- ☐ E' in grado di indicare se il grafo dell'ereditarietà genera un diamond problem.
- ☐ E' in grado di trasformare un tipo in un'altro senza indicazioni esterne.
- ☒ E' in grado di dedurre il tipo di alcune espressioni senza che sia necessario indicarlo esplicitamente.
- ☐ E' in grado di calcolare la corretta indentazione del codice e correggerla

## 24) Quali dei seguenti possono essere indicati come vantaggi dell'uso dei Datagram:

- ☐ La consegna di un singolo Datagram ha una latenza inferiore all'invio su Socket.
- ☒ Il singolo Datagram è isolato, quindi non è necessario introdurre nel protocollo dei separatori fra messaggi diversi.
- ☒ Un singolo Datagram può essere inviato a molti indirizzi con una sola istruzione.
- ☐ La consegna di un Datagram non è legata ad una specifica porta

## 25) E' importante gestire velocemente l'accettazione di una nuova connessione su di un ServerSocket perché:

- ☐ Finché non c'è un thread che attende in `ServerSocket::accept()`, le operazioni di scrittura sul Socket non riprendono.
- ☒ Finché non c'è un thread che attende in `ServerSocket::accept()`, le nuove richieste di connessione si accodano su di un buffer del sistema operativo che può avere una lunghezza molto limitata.
- ☐ Finché non c'è un thread che attende in `ServerSocket::accept()`, le operazioni di lettura sul Socket non riprendono.
- ☐ Finché non c'è un thread che attende in `ServerSocket::accept()`, i buffer di invio e ricezione dati del sistema operativo non sono sorvegliati e potrebbero riempirsi

## 26) Un oggetto *Future* rappresenta:

- ☒ Un calcolo che potrebbe produrre un risultato dopo un certo tempo.
- ☐ Una generica esecuzione concorrente.
- ☐ Il risultato di un calcolo parallelo terminato correttamente.
- ☐ Un calcolo concorrente terminato in modo errato

## 27) Associare ad ogni condizione di Coffmann una strategia utile per rimuoverla. Una delle strategie indicate non è rilevante.

La risposta corretta è: Attesa circolare → Ordinamento dell'acquisizione, (non rilevante) → Esecuzione fuori ordine, Mutua Esclusione → Algoritmi lock-free, Risorse non riassegnabili → Pre-emption, Possesso e attesa → Assegnazione delle risorse transazionale

## 28) Perché nel linguaggio Java si è deciso di introdurre i metodi di default nelle Interfacce?

- ☐ Per rendere più difficile modificare l'implementazione delle interfacce in modi non previsti.
- ☐ Per inseguire una feature richiesta dal mercato.

- ☐ Per evitare una possibilità di realizzare un diamond problem.
- ☒ Per poter estendere delle interfacce consolidate senza richiedere l'aggiornamento del codice esistente

## 29) Date le seguenti classi:

```
class Foo {  
    Foo(int a) {  
        // Costruttore Foo  
    }  
}  
  
class Bar extends Foo {  
    static {  
        // Inizializzatore statico  
    }  
    {  
        // Inizializzatore  
    }  
    Bar(int a, String b) {  
        super(a);  
        // Costruttore Bar  
    }  
}
```

Ordinare le strutture indicate secondo la sequenza con cui verranno eseguite

La risposta corretta è: Inizializzatore statico → 1, Costruttore Foo → 2, Costruttore Bar → 4, Inizializzatore → 3

## 30) La comunicazione su Socket ha diversi vantaggi, ma il fatto che i dati vengano presentati come uno stream di byte ha i seguenti svantaggi:

- ☒ Deve essere definito un protocollo con cui i due lati della comunicazione riconoscono l'inizio e la fine dei messaggi.
- ☐ Nel caso di comunicazione fra più di due parti sullo stesso socket, il protocollo deve permettere l'identificazione del mittente di ogni messaggio.
- ☒ Le due parti devono concordare in qualche modo l'encoding delle stringhe all'interno del protocollo di comunicazione.
- ☐ Le due parti della comunicazione devono inviare i propri dati il più velocemente possibile per diminuire la latenza della comunicazione

## 31) In questo codice di esempio:

```
try(DatagramSocket socket= new DatagramSocket(8080)) {  
    byte[] buf = new byte[16];  
    DatagramPacket packet = new DatagramPacket(buf, 16);  
    // A  
    String input = new String(packet.getData(), 0, packet.getLength());  
    System.out.println(input);  
    // B  
}
```

in che punto va inserita la chiusura della risorsa DatagramSocket:

- ☐ In un blocco finally da aggiungere in coda al blocco try.
  - ☐ Nel punto A.
  - ☐ Nel punto B.
  - ☒ Non è necessaria
-







[illegible]

[L'Intelligenza Emotiva](#)

[illegible]

- Il **processo di sviluppo** è un processo continuo che genera una nuova conoscenza continua.

La **teoria scientifica** è il tentativo di costruire una teoria che spieghi i fenomeni osservati che vengono utilizzati per creare dei modelli predittivi.

**Teoria** è un modello di realtà che spiega i fenomeni osservati, che è un tentativo di costruire una teoria che spieghi i fenomeni osservati che vengono utilizzati per creare dei modelli predittivi.

**Teoria** è un modello di realtà che spiega i fenomeni osservati, che è un tentativo di costruire una teoria che spieghi i fenomeni osservati che vengono utilizzati per creare dei modelli predittivi.

- Il **processo di sviluppo** è un processo continuo che genera una nuova conoscenza continua.
- Il **processo di sviluppo** è un processo continuo che genera una nuova conoscenza continua.
- Il **processo di sviluppo** è un processo continuo che genera una nuova conoscenza continua.

La risposta corretta è: **l'implementazione può analizzare l'esecuzione della subtask esportando dati da un server applicativo in funzione della loro caratteristiche**

Come molti altri linguaggi Object-Oriented, Java dispone di un meccanismo di ereditarietà per permettere classi minori ereditando le caratteristiche di una tra le seguenti superclassi:

**Gerghi sono più alternativi**

- Il meccanismo di ereditarietà del titolo di eredità è possibile creare un **Custom Problem** ✓
- Il Custom Problem è implementato in Java ✓
- Non è necessario ereditare da una sola alternativa ✓
- Non è necessario ereditare da una sola alternativa ✓

- `String dataCognomeImplementazione` non può essere interfaccia.
- `String dataCognomeImplementazione` può essere classe. 

La risposta corretta non è una (non può essere né classe né interfaccia), è possibile implementare più interfacce, o usare l'implementazione di un'altra interfaccia, è possibile usare un `Common Problem`.

La risposta "Tendenzialmente non implementa" è sbagliata perché si può implementare con Java. Oggi, forse decano.

**Taggi (7/7/2018)**

- E' un'area soggetta, perché la logica e le caratteristiche delle sue sono sempre variate. 
- La logica convergente analizza, senza la dilatazione dei processi (ci sono esempi).

- Monitorare gli indicatori degli obiettivi strategici definiti
- Il processo dovrebbe essere molto flessibile e con alcuni controlli

La risposta corretta è 1) ancora maggiore, perché la strategia non è un'attività che deve restare sempre la stessa.

La logica della risposta è la seguente:

Perché un'alternativa

- Fornire un'alternativa all'organigramma per l'implementazione dell'attività di ricerca e sviluppo
- Fornire una alternativa per definire l'implementazione della strategia di sviluppo
- Fornire una alternativa per definire l'implementazione della strategia di sviluppo

☐ Accettare una APT per definire informazioni di sequenza di oggetti.  
☐ Accettare una APT per definire informazioni di sequenza di oggetti.

La risposta corretta è: Accettare una APT per definire informazioni di sequenza di oggetti.

Quando il file del computer (o la rete) è il tipo italiano di Windows?

Seleziona un'alternativa:

- ☐ Il grado di calcolo e l'installazione del codice sorgente.
- ☐ Il grado di calcolo e l'installazione del codice sorgente.
- ☐ Il grado di calcolo e l'installazione del codice sorgente.
- ☐ Il grado di calcolo e l'installazione del codice sorgente.

1.4. Segnala almeno 4 (4) tipi di disegni organizzativi senza che dovessero indicarli esplicitamente.

- Il computer deve essere a garanzia che il codice sorgente della parte critica **non** venga effettivamente eseguito prima del controllo e segue il suo corso finché non viene eseguito.
- La risposta corretta è il computer deve essere a garanzia che il codice sorgente della parte critica **non** venga effettivamente eseguito prima del controllo che lo segue o che un solo thread elabora.

**E' importante notare attentamente l'esclusione di un processore da una CriticalSection perché:**

Dovrà un idempotenza

- Se due o più thread in attesa suCriticalSection entrano, nessuno processore viene escluso.

[illegible]

☐ poco indipendenti dal tipo di appartamento.

☐ La dipendenza è forte per tutti pubblici senza nessuna distinzione.

Perché nel frattempo non si è deciso di introdurre metodi di dibattito più sofisticati?

☐ Non si è riusciti.

☐ Per la mancanza di informazioni e di dati necessari.

☐ Per la natura più difficile, più dura, più impegnativa delle richieste (non si sono potuti evitare compromessi e soluzioni unilaterali).

☐ Per la scarsa capacità di coinvolgere un numero maggiore di persone.

[illegible]

La risposta corretta è: Produttore di risultato grezzo. Se il Cliente vuole il risultato (completato), il Produttore di risultato grezzo deve fornire il risultato grezzo.

**Domanda 5.1**  
 Qual è la funzione del cliente?  
 A) Fornire il risultato grezzo  
 B) Fornire il risultato  
 C) Fornire il risultato grezzo  
 D) Fornire il risultato

La risposta corretta è: Fornire il risultato grezzo. Il cliente fornisce il risultato grezzo al Produttore di risultato grezzo.

**Domanda 5.2**  
 Qual è la funzione del cliente?  
 A) Fornire il risultato grezzo  
 B) Fornire il risultato  
 C) Fornire il risultato grezzo  
 D) Fornire il risultato

La risposta corretta è: Fornire il risultato grezzo. Il cliente fornisce il risultato grezzo al Produttore di risultato grezzo.

☐ Risposta corretta (questo è il tuo punteggio) ☒ Risposta errata (questo è il tuo punteggio)

La dipendenza economica è la relazione tra:

☒ il datore di lavoro e il lavoratore ☐ il datore di lavoro e il datore di lavoro ☐ il datore di lavoro e il datore di lavoro ☐ il datore di lavoro e il datore di lavoro

- **Esclusione** dell'azienda dal subcontratto del componente Decisione: questa azione Decide a decisione stessa.

La risposta corretta è l'azienda parte di progetto che si decide a subcontratto (il lavoro dell'azienda stessa non è completamente parato, o completamente parato).

Devo un subcontratto (il modo non lavorare con nessuno solo stato del sistema può essere che)

Singoli per l'azienda

- In risposta del subcontratto e condizioni generali vengono determinati.
- In risposta del subcontratto (l'azienda) per la decisione (azienda) di lavoro (azienda) di lavoro (azienda) di lavoro.

[illegible]

☐ Il ministero-Dipartimento (Ministry) per work, intelligence (intelligence) del del servizi. ✓  
☐ Il ministero-Dipartimento (Ministry) per work, intelligence (intelligence) del del servizi. ✓  
☐ Il ministero-Dipartimento (Ministry) per work, intelligence (intelligence) del del servizi. ✓  
☐ Il ministero-Dipartimento (Ministry) per work, intelligence (intelligence) del del servizi. ✓

La risposta corretta è: Il ministero-Dipartimento (Ministry) per work, intelligence (intelligence) del del servizi. ✓

Perché il ministero-Dipartimento (Ministry) per work, intelligence (intelligence) del del servizi. ✓

- ☐ Funzione **predefinita** di conversione da fra. a componenti di 1 messaggio sistema
- ☐ Funzione **integrabile** anche in caso di output peraltro
- ☐ Funzione **funzione** data-funzione di 1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61/62/63/64/65/66/67/68/69/70/71/72/73/74/75/76/77/78/79/80/81/82/83/84/85/86/87/88/89/90/91/92/93/94/95/96/97/98/99/100/101/102/103/104/105/106/107/108/109/110/111/112/113/114/115/116/117/118/119/120/121/122/123/124/125/126/127/128/129/130/131/132/133/134/135/136/137/138/139/140/141/142/143/144/145/146/147/148/149/150/151/152/153/154/155/156/157/158/159/160/161/162/163/164/165/166/167/168/169/170/171/172/173/174/175/176/177/178/179/180/181/182/183/184/185/186/187/188/189/190/191/192/193/194/195/196/197/198/199/200/201/202/203/204/205/206/207/208/209/210/211/212/213/214/215/216/217/218/219/220/221/222/223/224/225/226/227/228/229/230/231/232/233/234/235/236/237/238/239/240/241/242/243/244/245/246/247/248/249/250/251/252/253/254/255/256/257/258/259/260/261/262/263/264/265/266/267/268/269/270/271/272/273/274/275/276/277/278/279/280/281/282/283/284/285/286/287/288/289/290/291/292/293/294/295/296/297/298/299/300/301/302/303/304/305/306/307/308/309/310/311/312/313/314/315/316/317/318/319/320/321/322/323/324/325/326/327/328/329/330/331/332/333/334/335/336/337/338/339/340/341/342/343/344/345/346/347/348/349/350/351/352/353/354/355/356/357/358/359/360/361/362/363/364/365/366/367/368/369/370/371/372/373/374/375/376/377/378/379/380/381/382/383/384/385/386/387/388/389/390/391/392/393/394/395/396/397/398/399/400/401/402/403/404/405/406/407/408/409/410/411/412/413/414/415/416/417/418/419/420/421/422/423/424/425/426/427/428/429/430/431/432/433/434/435/436/437/438/439/440/441/442/443/444/445/446/447/448/449/450/451/452/453/454/455/456/457/458/459/460/461/462/463/464/465/466/467/468/469/470/471/472/473/474/475/476/477/478/479/480/481/482/483/484/485/486/487/488/489/490/491/492/493/494/495/496/497/498/499/500/501/502/503/504/505/506/507/508/509/510/511/512/513/514/515/516/517/518/519/520/521/522/523/524/525/526/527/528/529/530/531/532/533/534/535/536/537/538/539/540/541/542/543/544/545/546/547/548/549/550/551/552/553/554/555/556/557/558/559/560/561/562/563/564/565/566/567/568/569/570/571/572/573/574/575/576/577/578/579/580/581/582/583/584/585/586/587/588/589/590/591/592/593/594/595/596/597/598/599/600/601/602/603/604/605/606/607/608/609/610/611/612/613/614/615/616/617/618/619/620/621/622/623/624/625/626/627/628/629/630/631/632/633/634/635/636/637/638/639/640/641/642/643/644/645/646/647/648/649/650/651/652/653/654/655/656/657/658/659/660/661/662/663/664/665/666/667/668/669/670/671/672/673/674/675/676/677/678/679/680/681/682/683/684/685/686/687/688/689/690/691/692/693/694/695/696/697/698/699/700/701/702/703/704/705/706/707/708/709/710/711/712/713/714/715/716/717/718/719/720/721/722/723/724/725/726/727/728/729/730/731/732/733/734/735/736/737/738/739/740/741/742/743/744/745/746/747/748/749/750/751/752/753/754/755/756/757/758/759/760/761/762/763/764/765/766/767/768/769/770/771/772/773/774/775/776/777/778/779/780/781/782/783/784/785/786/787/788/789/790/791/792/793/794/795/796/797/798/799/800/801/802/803/804/805/806/807/808/809/810/811/812/813/814/815/816/817/818/819/820/821/822/823/824/825/826/827/828/829/830/831/832/833/834/835/836/837/838/839/840/841/842/843/844/845/846/847/848/849/850/851/852/853/854/855/856/857/858/859/860/861/862/863/864/865/866/867/868/869/870/871/872/873/874/875/876/877/878/879/880/881/882/883/884/885/886/887/888/889/890/891/892/893/894/895/896/897/898/899/900/901/902/903/904/905/906/907/908/909/910/911/912/913/914/915/916/917/918/919/920/921/922/923/924/925/926/927/928/929/930/931/932/933/934/935/936/937/938/939/940/941/942/943/944/945/946/947/948/949/950/951/952/953/954/955/956/957/958/959/960/961/962/963/964/965/966/967/968/969/970/971/972/973/974/975/976/977/978/979/980/981/982/983/984/985/986/987/988/989/990/991/992/993/994/995/996/997/998/999/1000/1001/1002/1003/1004/1005/1006/1007/1008/1009/1010/1011/1012/1013/1014/1015/1016/1017/

- **Costo medio variabile** ( $CMV$ ), è composto da **costi variabili** sostenuti, modificando la struttura dell'investimento dello stesso **o**
- **Costo medio variabile** ( $CMV$ ) **costo medio variabile**, modificando la struttura dell'investimento dello stesso.

**La risposta corretta è:** Costo medio variabile ( $CMV$ ), è composto da **costi variabili** sostenuti, modificando la struttura dell'investimento dello stesso.

**Domanda 10**  
 Risposta corretta  
 Attribuzione: 100%  
 Risposta: 100%

Il **costo medio variabile** ( $CMV$ ) è ottenuto da **costi variabili** sostenuti, modificando la struttura dell'investimento dello stesso **o** **costo medio variabile** ( $CMV$ ) **costo medio variabile**, modificando la struttura dell'investimento dello stesso.

**Domanda 11**  
 Risposta corretta  
 Attribuzione: 100%  
 Risposta: 100%

Il **costo medio variabile** ( $CMV$ ) è ottenuto da **costi variabili** sostenuti, modificando la struttura dell'investimento dello stesso **o** **costo medio variabile** ( $CMV$ ) **costo medio variabile**, modificando la struttura dell'investimento dello stesso.

- **Esare-fuqurqurqur-Naam** **re-fo-fo** per **ripetere** un'azione o il suono di "fo" fatto volutamente.
- **Esare-fuqurqurqur-Naam** **re-fo-fo** per **ripetere** un'azione o il suono di "fo" fatto volutamente.

La risposta corretta è **Esare-fuqurqurqur-Naam** **re-fo-fo** per **ripetere** un'azione o il suono di "fo" fatto volutamente.

Le risposte errate sono:

- **Esare-fuqurqurqur-Naam** **re-fo-fo** per **ripetere** un'azione o il suono di "fo" fatto volutamente.
- **Esare-fuqurqurqur-Naam** **re-fo-fo** per **ripetere** un'azione o il suono di "fo" fatto volutamente.

[illegible][illegible]

- **Importare** significa **importare** configurazioni di strumenti esistenti o strumenti da copiare.
- **Casi d'uso** particolari non supporta tutte le funzionalità del framework.
- **Difficoltà** a mantenere gli utenti del proprio codice su un'architettura di framework.

La dipendenza connessa alla limitazione di funzionalità influenzata con la maggiore difficoltà di implementazione. **Casi d'uso** particolari non supporta tutte le funzionalità del framework, **Importare** significa **importare** configurazioni di strumenti esistenti o strumenti da copiare, **Difficoltà** a mantenere gli utenti del proprio codice su un'architettura di framework.

**Quale di queste caratteristiche ha il proprio da dire?** **Quali sono le sue espressioni?**

**Scegli le alternative:**

- **Importare** significa **importare** configurazioni di strumenti esistenti o strumenti da copiare.
- **Casi d'uso** particolari non supporta tutte le funzionalità del framework.
- **Difficoltà** a mantenere gli utenti del proprio codice su un'architettura di framework.

[illegible][illegible]

- ☐ Se l'istogramma evidenzia la presenza di dati anomali, è opportuno che l'istogramma presenti:
  - ☐ La presenza di dati anormali (istogrammi che evidenziano l'inflessione all'estremità dei dati).
- La risposta corretta/erronea è: Istantanea (Istantanea è corretta, quindi non è necessario introdurre nel protocollo della risposta) Se l'istogramma evidenzia la presenza di dati anormali, è opportuno che l'istogramma presenti:
  - ☐ La presenza di dati anormali (istogrammi che evidenziano l'inflessione all'estremità dei dati).

Quasi tutti gli istogrammi non distribuiscono le statistiche in determinati punti esati.

Tutti gli istogrammi sono distribuiti:

- ☐ Distribuzione più rapida delle frequenze distribuite a più modi.
- ☐ Possibilità di prendere una data più grande della capacità di una singola macchina.

[illegible]

- ☐ **Numero adulti** nel cui modulo sono inseriti genitori e/o genitori separati.
- ☐ **Numero bambini** nel cui modulo sono inseriti i bambini.

La seguente struttura è stata particolarmente utile anche per il modulo concernente dati che rappresentano genitori separati (ex allegato) della Fondazione. (Bontempo-Fall'Industria)

[illegible][illegible]

# Simulazione Altri Paradigmi di Programmazione

---

## 1) In Java, i metodi di una interfaccia sono:

- ☐ a. Pubblici solo se preceduti dalla parola chiave `public`.
- ☒ b. Pubblici senza necessità di specificarlo.
- ☐ c. Privati a meno che non sia specificato diversamente.
- ☐ d. Hanno la stessa visibilità dell'interfaccia che li contiene.

## 2) La comunicazione via socket è:

- ☐ a. unidirezionale, da punto a punto
- ☐ b. bidirezionale, verso un insieme di indirizzi
- ☐ c. unidirezionale, verso un indirizzo o verso un insieme di indirizzi
- ☒ d. bidirezionale, da punto a punto

## 3) In uno stream Java, una operazione *short-circuiting* può:

- ☐ a. causare una esecuzione seriale anche se viene richiesta l'elaborazione parallela
- ☐ b. modificare gli elementi dello stream mentre li elabora
- ☐ c. elaborare lo stream con un consumo di memoria costante
- ☒ d. interrompere l'elaborazione prima di consumare lo stream in ingresso

## 4) Quali di queste sono problematiche dell'esecuzione concorrente di programmi:

Scegli una o più alternative:

- ☐ preemption
- ☐ instruction ordering
- ☒ deadlock
- ☒ starvation
- ☒ race conditions
- ☒ indeterminismo
- ☐ clock skew
- ☐ Amhdal's law

## 5) I valori in un programma Java possono essere:

- ☒ a. Oggetti, cioè istanze di una classe, oppure valori primitivi, oppure il valore `null`.
- ☐ b. Solo oggetti, cioè istanze di una classe.

- ☐ c. Solo classi.
- ☐ d. Solo valori primitivi e null.

**6) Quale di queste azioni un attore *non* può compiere durante l'elaborazione di un messaggio:**

- ☐ a. creare un altro attore
- ☒ b. interrompere l'elaborazione di un altro attore
- ☐ c. modificare il proprio stato
- ☐ d. inviare un numero finito di messaggi

**7) Le variabili Thread local:**

- ☐ a. Per ciascun Thread hanno lo stesso valore, inizializzato separatamente
- ☐ b. Per ciascun Thread hanno lo stesso valore, inizializzato uguale per tutti e costante
- ☐ c. Per ciascun Thread mantengono un valore distinto, inizializzato uguale per tutti
- ☒ d. Per ciascun Thread mantengono un valore distinto, inizializzato separatamente

**8) L'affidabilità del Datagram dipende, fra le altre cose, dalla frammentazione dei pacchetti. Ne dipende perchè:**

- ☐ a. al diminuire della frammentazione, aumenta la probabilità che i pacchetti giungano fuori ordine, invalidando il messaggio
- ☐ b. al diminuire della frammentazione, diminuisce la probabilità che l'intero messaggio arrivi a destinazione
- ☐ c. all'aumentare della frammentazione, diminuisce la probabilità che i pacchetti giungano in ordine, invalidando il messaggio
- ☒ d. all'aumentare della frammentazione, diminuisce la probabilità che l'intero messaggio arrivi a destinazione

**9) In Java, ai fini della compilazione classica del codice, un file sorgente deve trovarsi, ed essere denominato, nel seguente modo:**

- ☐ a. In una sottodirectory della directory di lavoro, denominato come una delle classi contenute.
- ☐ b. Nel percorso corrispondente alla struttura del package, con un nome qualsiasi ed estensione .java
- ☒ c. Nel percorso corrispondente alla struttura del package, e denominato come la classe pubblica contenuta.
- ☐ d. Nella directory di lavoro, con un nome qualsiasi.

**10) L'oggetto Socket rappresenta una connessione attiva, e propone, come interfaccia di invio e ricezione di dati, degli stream di byte. La difficoltà nel costruire un protocollo di comunicazione risiede:**

- ☐ a. nel fatto che possono essere trasmessi solo caratteri ASCII
- ☐ b. nel fatto che i caratteri non stampabili possono interferire con la suddivisione delle righe
- ☒ c. nel fatto che il separatore dei messaggi deve essere incluso nel contenuto
- ☐ d. nel fatto che il server può ricevere più messaggi contemporaneamente, e li deve separare

**11) I *Conflict-Free Replicated Data Type* consentono di riunire modifiche contemporanee da molte sorgenti. Tuttavia la loro utilità è limitata dal fatto che:**

- ☐ a. sono onerosi dal punto di vista del tempo di trasmissione
- ☒ b. possono rappresentare un insieme limitato di casi d'uso
- ☐ c. sono onerosi dal punto di vista computazionale
- ☐ d. sono complicati dalla necessità di scrivere codice ad-hoc

**12) La fallacia "Transport cost is zero" è ancora valida ma con un significato diverso. Il motivo è:**

- ☐ a. è cambiato il significato di trasporto: si intendeva senza condizioni, oggi è insicuro
- ☐ b. il costo del trasporto oggi tende progressivamente a zero
- ☐ c. oggi il trasporto è su tecnologie differenti, quindi la metrica del costo ora include la latenza
- ☒ d. è cambiato il significato di costo: si intendeva costo monetario, ora si intende costo energetico

**13) Storicamente, la programmazione concorrente è stata introdotta per:**

- ☐ a. Affrontare più problemi contemporaneamente
- ☒ b. Sfruttare pienamente le risorse degli elaboratori ed ottenere prestazioni più elevate
- ☐ c. Separare il lavoro di interazione con l'esterno dal puro calcolo
- ☐ d. Contrastare la legge di Moore e la differenza di velocità fra CPU e I/O

**14) In relazione al modo in cui il linguaggio Java viene di norma eseguito, quale di queste affermazioni è corretta:**

- ☐ a. Un insieme di sorgenti in linguaggio Java deve essere ricompilato per ogni JVM specifica per ciascun sistema operativo.
- ☐ b. Un insieme di sorgenti viene eseguito direttamente, in forma interpretata, senza necessità di compilazione.
- ☐ c. Un sorgente per volta viene trasformato e aggiunto ad un eseguibile binario nativo.
- ☒ d. Un insieme di sorgenti in linguaggio Java viene trasformato in bytecode, ed è quest'ultimo ad essere eseguito dalla JVM.

**15) I Reactive Stream aggiungono, al modello semantico delle Reactive Extensions, i seguenti concetti:**

Scegli una o più alternative:

- ☐ a. i componenti della pipeline possono dichiarare quanti elementi possono emettere ai componenti successivi
- ☐ b. i componenti della pipeline possono lanciare eccezioni
- ☐ c. i componenti della pipeline possono avere uno stato distribuito
- ☒ d. I componenti della pipeline possono esercitare una *back-pressure* nei confronti dei componenti che li precedono
- ☒ e. I componenti della pipeline possono essere distribuiti
- ☐ f. i componenti della pipeline possono chiudere in anticipo l'elaborazione dello stream