Appunti di Algoritmi e Strutture Dati

4 ottobre 2022

Rosso Carlo

Contents

1 appello 1 **2**

2 Appello 2 **9**

3 Appello 3 **13**

1 appello 1

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 Risposta:

1. statico
2. Costruttore Foo
3. Inizializzatore
4. Costruttore Bar

Nelle Reactive Èxtensions, quali di queste operazioni non è necessario (o possibile) specificare per elaborare gli oggetti emessi da un observable:
Scegli una o più alternative:

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

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:
Scegli un'alternativa:

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

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?
Scegli un'alternativa:

- L'implementazione può analizzare le operazione della catena, e decidere se eseguirle parallelamente o in serie.

- 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 operazioni della catena, e prendere decisioni su come applicarle in funzione delle loro caratteristiche.

//

Come molti altri linguaggi Object-Oriented, Java ha a disposizione un meccanismo di ereditarietà per estendere classi esistenti senza doverle modificare. L'ereditarietà in Java ha le seguenti caratteristiche:
Scegli una o più alternative:

- A causa dell'introduzione dei default methods, è possibile causare un Diamond Problem.
- Il Diamond Problem è impossibile per costruzione.
- Una classe può ereditare da una sola altra classe.
- Una interfaccia può ereditare da un'altra interfaccia.
- Una interfaccia può implementare una sola altra interfaccia.
- Una classe può implementare più interfacce.

La fallacia "Network is homogeneus" è stata aggiunta alle prime sette da Gosling, proprio in seguito alle prime esperienze con Java. Oggi, la sua rilevanza:
Scegli un'alternativa:

- È ancora maggiore, perché le tipologie e le caratteristiche delle reti sono sempre più varie.
- È tutto sommato un problema risolto dalla diffusione dei protocolli più recenti.
- Non è più rilevante dopo la diffusione delle connessioni wireless.
- È ancora rilevante, ma si tratta ormai di un problema ormai sotto controllo.

Lo scopo delle Reactive Extensions è:
Scegli un'alternativa:

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

Quando si dice che il compilatore Java ha delle capacità di Type Inference si intende che:
Scegli un'alternativa:

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

Quando si dice che la parola chiave syncronized introduce una relazione di happens-before nel codice, si intende che:
Scegli un'alternativa:

- Il compilatore viene istruito a garantire che il codice sorvegliato dalla parola chiave syncronized 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 syncronized 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 syncronized venga allineato al trasferimento di dati fra la memoria principale e la cache del microprocessore.
- Il compilatore viene istruito a garantire che il codice sorvegliato dalla parola chiave syncronized venga effettivamente eseguito prima del codice che lo segue, e da un solo Thread alla volta.

È importante gestire velocemente l'accettazione di un pacchetto da una DatagramSocket perché:
Scegli un'alternativa:

- Se non c'è un thread in attesa su DatagramSocket:receive(), nessun pacchetto viene ricevuto.
- Se non c'è un thread in attesa della ricezione di un pacchetto, questo viene scartato.
- Se c'è un thread in attesa su DatagramSocket::receive(), gli invii sono bloccati.
- I pacchetti ricevuti vengono conservati in buffer limitati; se si riempiono, i messaggi successivi sono scartati.

I membri di una interfaccia Java:
Scegli un'alternativa:

- Nessuno di essi deve specificare una implementazione.
- Sono immutabili.
- Sono tutti pubblici, senza necessità di indicarlo.
- Sono indipendenti dal tipo di appartenenza.

Perché nel linguaggio Java si è deciso di introdurre i metodi di default nelle Interfacce?
Scegli un'alternativa:

- Per inseguire una feature richiesta dal mercato.
- Per rendere più difficile modificare l'implementazione delle interfacce in modi non previsti.
- Per evitare una possibilità di realizzare un diamond problem.
- poter estendere delle interfacce consolidate senza richiedere l'aggiornamento del codice esistente.

Una operatore short-circuiting all'intemo di una catena di elaborazione di uno Stream può:
Scegli un'alternativa:

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

Selezionare quali delle seguenti sono condizioni di Coffman necessarie per l'instaurarsi di un deadlock.
Scegli una 0 più alternative:

- Ordinamento dell'esecuzione
- Possesso e attesa
- Salvataggio del contesto
- Temporizzazione dell'accesso
- Mutua Esclusione

- Risorse non riassegnabili
- Attesa circolare

Usando i Reactive Stream, la gestione più granulare della composizione della pipeline di elaborazione dello stream permette di:
Scegli un'alternativa:

- Scegliere algoritmi di suddivisione del lavoro più efficienti di quelli della libreria standard, perché più facili da aggiornare.
- Isolare la parte di pipeline che si desidera sia resa parallela; gli Stream della libreria standard sono 0 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.
- Ridurre la latenza dell'elaborazione dei vari componenti decidendo quante risorse dedicare a ciascuno di essi.

Se in un sistema distribuito i nodi non trovano un consenso sullo stato del sistema, può accadere che:
Scegli un'alternativa:

- 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 pace condition.
- Le risposte del sistema siano incoerenti e dipendano da quale nodo viene contattato.
- Le risposte del sistema non siano disponibili in quanto gli stati differenti si annullano.

Indicare quali fra le seguenti sono problematiche che rendono la serializzazione un processo complesso, che richiede molte attenzioni:
Scegli una o più alternative

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

Quando di un sistema reattivo si indicano le sue qualità come Responsive, Resilient, Elastic, Message-Oriented, con la qualità "Message-Oriented" sî intende:
Scegli un'alternativa:

- Il sistema risponde proporzionalmente alla quantità di richieste in ingresso.
- L'unica primitiva di comunicazione fra i componenti è il messaggio asincrono.
- Il sistema è disponibile anche in caso di guasto parziale.
- Il sistema fornisce delle funzionalità di coda di messaggi distribuita.

Nell'astrazione delle Reactive Extensions, un subject può:
Scegli un'alternativa:

- Osservare uno Stream esaminando solo alcuni elementi.
- Osservare altri subject e observable alterando la struttura dello stream fra di loro.

- Osservare diversi observable, e comportarsi da observable esso stesso, modificando la struttura dell'elaborazione dello stream.
- Osservare uno Stream in modalità parallela.

Il compilatore Java può riordinare le istruzioni di un blocco di codice in seguito a considerazioni di ottimizzazione ed efficienza. Per imporre un ordinamento preciso è possibile:
Scegli un'alternativa:

- Usare la parola chiave syncronized per introdurre un ordinamento specifico, con una relazione di happens-before.
- Usare la parola chiave final per prevenire modifiche di questo tipo durante la compilazione.
- Usare la parola chiave volatile per segnalare un blocco di codice che non va riordinato.
- Usare la parola chiave strictfp per richiedere una semantica precisa delle istruzioni riordinate.

Un Thread esce dallo stato blocked quando:
Scegli un'alternativa:

- Ottiene il lock che stava aspettando o viene interrotto.
- Viene interrotto (e solo in questo caso).
- Trascorre esattamente il tempo impostato (e solo în questo caso).
- Ottiene la risorsa di sistema che aveva richiesto.

Quale dei seguenti non è uno Stream Flag, cioè una caratteristica che uno Stream può dichiarare ed uno operatore (intermedio o terminale) utilizzare per organizzare l'esecuzione:
Scegli un'alternativa:

- SUBSIZED - lo stream può essere diviso in partizioni di dimensione nota.
- UNTYPED - non è noto a priori il tipo degli elementi.
- CONCURRENT - lo stream supporta l'elaborazione parallela.
- NONNULL - tutti gli elementi sono diversi da null.

Che tipo di rischi devono essere considerati nella scelta e nell'adozione di un framework per la costruzione di applicazioni distribuite?
Scegli una o più alternative:

- Errori operativi dovuti a bachi nel codice del framework.
- Indisponibilità di assistenza remota nella risoluzione degli errori applicativi.
- Direzione di sviluppo non allineata con le esigenze dell'evoluzione dell'applicazione.
- Errori operativi dovuti a configurazioni di default insicure o difficili da capire.
- Casi d'uso particolari non coperti dalle funzionalità del framework.
- Difficoltà a rimuovere gli errori dal proprio codice una volta introdotto il framework.

Quale di queste caratteristiche è propria della sintassi switch-case come espressione:
Scegli un'alternativa:

- I risultati devono essere tutti valori della stessa interfaccia.
- Ogni caso deve produrre un risultato diverso.
- È possibile il fall-through da un caso all'altro.

- L'elenco delle opzioni deve essere esaustivo.

Il modello dei Thread permette ad un Processo di organizzare più linee di esecuzione al suo interno incorrendo in una minore penalità di cambiamento del contesto durante l'esecuzione. Tuttavia, si ritrova a dover gestire:
Scegli un'alternativa:

- L'accesso e la condivisione delle risorse
- La ricezione degli eventi di rete
- L'invio dei byte alle periferiche di I/O
- L'organizzazione della memoria esterna

Quali dei seguenti possono essere indicati come vantaggi dell'uso dei Datagram:
Scegli una o più alternative:

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

Quali vantaggi si cercano nel distribuire lo stato di un sistema su più nodi:
Scegli una 0 più alternative:

- Elaborazione più rapida delle richieste distribuite a più nodi.
- possibilità di gestire uno stato più grande della capacità di una singola macchina.
- Maggiore sicurezza dei dati gestiti dal sistema di consenso.
- Accesso più rapido da località differenti.

Le classi del package java.concurrent.atomic
Scegli un'alternativa:

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

Nel linguaggio Java, una variabile final:
Scegli un'alternativa:

- Può contenere un valore di un solo tipo.
- 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.

Una variabile di tipo threadLocal<T>:
Scegli un'alternativa:

- Possiede un valore differente per ogni Thread che vi accede.
- Permette ad più Thread di accedere rapidamente al valore che contiene.
- Un solo Thread per volta può accedere al valore contenuto.

Più Thread possono accedere allo stesso valore senza interferire fra loro.
Una classe Java dichiarata abstract è visibile:
Scegli un'alternativa:

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

Quale delle seguenti affermazioni riguardante le Lambda Expression è vera:

- Rendono Java un linguaggio funzionale perché possono essere passate come parametri diuna chiamata ad un metodo.
- Non rendono Java un linguaggio funzionale perché non sono una entità del linguaggio, masolo 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 comportamentodi un metodo chiamato.

Nelle Reactive Extensions, quali di queste operazioni nonè necessario (o possibile) specificare per elaborare glioggetti 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.

La fallacia "Latency is zero" è ancora rilevante perché:

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

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 modoquasi 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 APIdi livello più elevato.
- Aggiornamento continuo che non richiede intervento da parte dello sviluppatore.
- Facilità di realizzazione perché le parti più strutturali sono giù implementate.

L'interfaccia Collector permette di eseguire la riduzionead un risultato di uno Stream parallelo in modo piùefficiente perché:

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

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 timeoutimpostato.
- Viene interrotto (e solo in questo caso)

2 Appello 2

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

Implementare un programma di rete usando l'astrazione dei Channels della libreria standard di Java permette di nonoccuparsi di molti dettagli riguardanti l'interazione con ilmezzo di comunicazione, ma:

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

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 deiProcessi i Thread devono direttamente controllare il loro accesso. Le Fiber rendono esplicitala concorrenza con lo scopo di essere ancora più leggere dei Thread.

- I Processi sono raggruppamenti logici di risorse che nei Thread vengono associati a risorsefisiche. 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 sottrratta, mettendolo in stato waiting. Le Fiber rendono la gestione dellaconcorrenza più esplicita.
- I Processi evitano il deadlock attraverso l'ordinamento delle priorità. I Thread invece nonsono ordinati e devono essere manualmente controllati per evitare conflitti nella gestionedelle risorse. Le Fiber sono una evoluzione più efficiente dei Thread.

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
- È meno costosa in termini di cicli di CPU che nel caso di accesso esclusivo.
- È più performante nel caso generale

È 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 sulSocket non riprendono.
- Finché non c'è un thread che attende in ServerSocket::accept(), le nuove richieste diconnessione 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 sulSocket non riprendono.
- Finché non c'è un thread che attende in ServerSocket::accept(), i buffer di invio e ricezionedati del sistema operativo non sono sorvegliati e potrebbero riempirsi

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

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

- 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

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

In questo codice di esempio:

```
try(DatagramSocketsocket=newDatagramSocket(8080){
    byte[] buf=newbyte[16];
    DatagramPacketpacket=newDatagramPacket(buf, 16);//
    //A
    Stringinput=newString(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

Associare ciascuna struttura di gestione della concorrenza al suo ambito di applicazione:

- Semaphore — Gestione di un insieme omogeneo di risorse,
- Lock Gestione — esplicita, senza legami sintattic, del blocco e dello sblocco della sezione critica.
- syncronized — Gestione della concorrenza tramite la struttura sintattica del codice.
- condition — Gestione dell'accesso alla stessa sezione critica in condizioni di blocco e/o sblocco dilferenti.
- object: :Wait() — Gestione esplicita dell'accesso ad un singolo oggetto. six

Quali di queste caratteristiche sono richieste per parlare di programmazione concorrente:
Scegli una o più alternative:

- Comunicazione tramite messaggi
- Processi separati
- Uniformità tecnologica fra i processi
- Coesistenza sullo stesso nodo di calcolo
- Condivisione di risorse
- Esecuzione contemporanea

L'astrazione channel della libreria standard di Java permette di non occuparsi di molti dettagli riguardanti l'interazione con il mezzo di comunicazione. 1 vari metodi che l'astrazione richiede di implementare sono accomunati dall'uso di un parametro attachment. Il suo scopo è:
Scegli un'alternativa:

- Rendere accessibile in ogni momento il canale sottostante la comunicazione.
- Trasportare in sicurezza il contesto della conversazione, fra metodi che saranno richiamati da thread differenti.
- Distribuire fra i vari metodi i dati globali del programma.
- Fornire il dato letto dal canale di comunicazione.

Perché l'elaborazione dia un risultato corretto, è richiesto che le operazioni interne siano:
Scegli una o più alternative:

- Prive dî uno stato interno.
- Implementate da oggetti diversi.
- Prive di allocazioni di nuovi oggetti.
- Non invasive: non devono modificare o interferire con gli elementi dello Stream.

Come molti altri linguaggi Object-Oriented, Java ha a disposizione un meccanismo di ereditarietà per estendere classi esistenti senza doverle modificare. L'ereditarietà in Java ha le seguenti caratteristiche:
Scegli una o più alternative:

- A causa dell'introduzione dei default methods, è possibile causare un Diamond Problem.
- Una interfaccia può implementare una sola altra interfaccia.
- Una classe può implementare più interfacce.
- Il Diamond Problem è impossibile per costruzione.
- Una classe può ereditare da una sola altra classe.
- Una interfaccia può ereditare da un'altra interfaccia.

La fallacia "Bandwidth is infinite":
Scegli un'alternativa:

- È ancora rilevante perché anche se è cresciuta la banda mediamente disponibile, è anche molto aumentata la quantità di dati trasmessi.
- È ancora rilevante in quanto dipendente da limiti fisici.
- Non è più rilevante in quanto anche in mobilità la banda disponibile è ubiquitaria ed elevata.

Quale delle seguenti affermazioni è corretta riguardo allo stato di un Attore:
Scegli un'alternativa:

- Lo stato di un Attore viene modificato da ogni messaggio ricevuto.
- Lo stato di un Attore può essere modificato solo dall'Attore stesso o dall'Attore che lo ha creato.
- Lostato di un Attore può essere modificato da più messaggi contemporaneamente, e va gestito con variabili thread-safe.
- Lo stato di un Attore è privato, e può essere solo modificato alla ricezione di un messaggio.

Nel linguaggio Java, un metodo final:
Scegli un'alternativa:

- Il suo valore di ritomo non può essere modificato.
- Deve essere reimplementato da una classe derivata.
- Viene richiamato al momento della cancellazione dell'oggetto.
- Non può essere reimplementato da una classe derivata.

La parola chiave volatile fornisce un particolare tipo di garanzie di concorrenza. È necessaria a causa dî:
Scegli un'alternativa:

- Effetti quantistici di interferenza fra celle vicine di memoria.
- Peculiarità della struttura della memoria di sistemi multiprocessore.

- Gestione della permanenza dei dati in seguito al riavvio della JVM.
- Effetti termici delle memorie a velocità superiori ad una certa soglia.

Uno stream rappresenta una sequenza di elementi, potenzialmente infinita. L'obiettivo di questa astrazione è:
Scegli un'alternativa:

- Permettere di descrivere l'elaborazione in termini i sugli i, e non dell delliterazione.
- Fornire l'API per attraversare più rapidamente la collezione.
- Uniformare l'accesso a più collezioni di struttura differente.
- Permettere di controllare più finemente l'avanzamento dell'ierazione.

3 Appello 3

In questo codice di esempio:

```
        try(ServerSocket serverSocket= new ServerSocket(8030);
        Socket socket serverSocket.accept();
        Printuiriter out = new PrintWriter(socket.getOutputStream(), true);
        An = new Buff Inpi den(socket..getInpi 05) {
        String inputtine;
        while ((inputline = in.readLine()) != null) {
        System.out.printIn("Received:-" + inputlLine);
        out.printin(*Hello * + inputline);
}
```

Una eccezione lanciata dall'oggetto out ha come effetto:
Scegli un'alternativa:

- La chiusura della sola risorsa in.
- La chiusura delle risorse serverSocket e socket, ma non di in.
- La chiusura di tutte le risorse e l'uscita dal blocco.
- L'uscita dal blocco con perdita delle risorse allocate.

Quali dei seguenti possono essere indicati come vantaggi dell'uso dell'oggetto Socket per la comunicazione:
Scegli una o più alternative:

- La connessione con un Socket non è legata ad una specifica porta del nodo connesso.
- La comunicazione può essere bidirezionale.
- L'orientamento alla connessione del Socket permette di trasferire in modo affidabile quantità di dati rilevanti.
- l'buffera disposizione per le connessioni su Socket sono più ampi di quelli per i Datagram.

Quali vantaggi si vogliono ottenere nel distribuire lo stato di un sistema su più nodi:
Scegli una o più alternative:

- Suddivisione del lavoro fra più nodi per una elaborazione più rapida delle richieste.
- Affidabilità rispetto al guasto fisico di uno 0 più nodi.
- Maggiore sicurezza dei dati gestiti dal sistema di consenso.
- Gestione di una mole di dati più grande della capacità di una singola macchina.

Un Thread può trovarsi in diversi stati di attesa: waiting, timed waiting, blockes. In cosa si distinguono?
Scegli un'alternativa:

- Dipendono da questioni tecnologiche mediate dalla JVM.
- Dipendono dalla chiamata di libreria standard usata per richiedere una risorsa.
- Sono gestiti da strutture sintattiche differenti.
- È diverso il motivo dell'attesa, la sua durata ed il modo in cui si esce dallo stato.

Una classe Java dichiarata senza modificatori di visibilità:
Scegli un'alternativa:

- E visibile da ogni classe del sistema.
- Un modificatore di visibilità è obbligatorio.
- Da ogni classe dello stesso package.
- Solo dalle classe che la estendono.