

# Appelli Paradigmi di Programmazione - Domande a Scelta Multipla

## APPELLO 1 - Reactive Extensions e Sistemi Distribuiti

### Domanda 1

Nelle Reactive Extensions, l'Observable è concettualmente simile a:

- a) Un Iterator che permette di scorrere una collezione
- b) Un Stream che emette nel tempo una sequenza di valori
- c) Un Thread che esegue operazioni in parallelo
- d) Un Subject che può consumare e produrre eventi

**Risposta corretta: b**

### Domanda 2

Quale delle seguenti operazioni NON è necessario specificare quando si elaborano gli oggetti emessi da un Observable? [Scegli 1 o più alternative]

- a) Il comportamento alla ricezione di un oggetto
- b) Il numero di oggetti che l'Observable è autorizzato ad inviare
- c) Il comportamento alla richiesta di separazione di uno stream parallelo
- d) Il comportamento al termine dello stream di oggetti
- e) Il comportamento alla ricezione di un errore

**Risposte corrette: b, c**

### Domanda 3

Nel Reactive Manifesto, un sistema "Elastic" deve essere:

- a) In grado di rispondere sempre entro tempi prevedibili
- b) Capace di replicare i componenti per gestire i fallimenti
- c) In grado di consumare risorse variabili in funzione del carico
- d) Orientato alla comunicazione tramite messaggi

**Risposta corretta: c**

### Domanda 4

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

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

**Risposta corretta: a**

### Domanda 5

Lo Scheduler nelle Reactive Extensions permette di:

- a) Sincronizzare l'accesso a risorse condivise
- b) Gestire la terminazione degli Observable
- c) Indicare il tipo di concorrenza desiderato per ogni operatore
- d) Combinare più Observable in un unico flusso di dati

**Risposta corretta: c**

---

## APPELLO 2 - JIT vs AOT e GraalVM

### Domanda 6

Confrontando JIT (Just-In-Time) e AOT (Ahead-Of-Time), quale affermazione è corretta?

- a) AOT ha prestazioni a regime sempre superiori a JIT
- b) JIT ha un tempo di avvio più rapido rispetto ad AOT
- c) AOT ha un consumo di memoria inferiore durante l'esecuzione
- d) JIT ha una compatibilità inferiore rispetto ad AOT

**Risposta corretta: c**

### Domanda 7

Con i metodi di esecuzione nativa del codice Java (come AOT), si ottiene: [Scegli 1 o più alternative]

- a) Prestazioni a regime che possono essere superiori a quelle dell'esecuzione JIT
- b) Una diminuzione delle risorse necessarie durante l'esecuzione
- c) Un tempo di compilazione più lungo
- d) Un avvio più rapido dell'applicazione
- e) Prestazioni a regime che possono essere inferiori a quelle dell'esecuzione JIT

**Risposte corrette: c, d, e**

### Domanda 8

GraalVM include tutti i seguenti componenti ECCETTO:

- a) Un nuovo compilatore JIT per Java
- b) Un compilatore ahead-of-time per Java
- c) Una specifica di codice intermedio chiamata "Truffle"
- d) Un garbage collector specializzato per applicazioni native

**Risposta corretta: d**

### Domanda 9

CRaC (Coordinated Restore at Checkpoint) risolve il problema del tempo di startup:

- a) Compilando il codice Java direttamente in eseguibili nativi
- b) Salvando lo stato della memoria della JVM e riavviandolo successivamente
- c) Utilizzando un compilatore JIT più efficiente
- d) Riducendo la dimensione dell'applicazione attraverso ottimizzazioni

**Risposta corretta: b**

## **Domanda 10**

La compilazione ahead-of-time in Java ha come principale limitazione:

- a) L'impossibilità di ottimizzazioni runtime
- b) La necessità di limitare reflection e classloading dinamico
- c) L'incompatibilità con le librerie standard
- d) L'aumento significativo delle dimensioni dell'eseguibile

**Risposta corretta: b**

---

## **APPELLO 3 - Stream API e Programmazione Funzionale**

### **Domanda 11**

Le operazioni intermedie sugli Stream si dividono in stateless e stateful. Quale delle seguenti è un'operazione STATEFUL?

- a) filter
- b) map
- c) limit
- d) peek

**Risposta corretta: c**

### **Domanda 12**

Quale delle seguenti operazioni è TERMINALE in uno Stream?

- a) distinct
- b) sorted
- c) collect
- d) skip

**Risposta corretta: c**

### **Domanda 13**

Nel pattern di esecuzione degli Stream, le operazioni intermedie:

- a) Vengono eseguite immediatamente quando vengono chiamate
- b) Non devono modificare gli elementi dello stream
- c) Possono sempre mantenere uno stato interno
- d) Sono sempre più efficienti delle operazioni terminali

**Risposta corretta: b**

### Domanda 14

Un Stream può essere configurato per l'esecuzione parallela. Cosa accade se si applica sia `parallel()` che `sequential()` nella stessa pipeline? a) Viene generata un'eccezione a runtime b) L'ultima configurazione nella catena prevale c) Viene eseguita una versione mista parallela-sequenziale d) Viene automaticamente scelta la configurazione più efficiente

**Risposta corretta: b**

### Domanda 15

La classe Optional è stata introdotta per:

- a) Ottimizzare le performance degli Stream
- b) Rendere esplicita la rappresentazione di un valore che potrebbe non esserci
- c) Gestire eccezioni in modo funzionale
- d) Sostituire completamente l'uso di null in Java

**Risposta corretta: b**

---

## APPELLO 4 - Java Avanzato e Concorrenza

### Domanda 16

Date le seguenti classi, ordinare l'esecuzione delle strutture:

```
java

class Foo {
    Foo (int a) {
        //Costruttore Foo
    }
}

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

Ordine di esecuzione:

- a) Inizializzatore statico, Costruttore Foo, Inizializzatore, Costruttore Bar
- b) Costruttore Foo, Inizializzatore statico, Inizializzatore, Costruttore Bar
- c) Inizializzatore statico, Inizializzatore, Costruttore Foo, Costruttore Bar
- d) Inizializzatore, Inizializzatore statico, Costruttore Foo, Costruttore Bar

**Risposta corretta: a**

### **Domanda 17**

Una variabile di tipo ThreadLocal<T>: a) Possiede un valore differente per ogni Thread che vi accede b) Permette a più Thread di accedere rapidamente al valore che contiene c) Un solo Thread per volta può accedere al valore contenuto d) Più Thread possono accedere allo stesso valore senza interferire tra loro

**Risposta corretta: a**

### **Domanda 18**

Quale delle seguenti affermazioni riguardante le Lambda Expression è vera?

- a) Rendono Java un linguaggio funzionale perché possono essere passate come parametri
- b) Non rendono Java un linguaggio funzionale perché non sono un'entità del linguaggio, ma solo una convenienza sintattica
- c) Non rendono Java un linguaggio funzionale perché non sono facilmente componibili
- d) Rendono Java un linguaggio funzionale perché permettono di astrarre sul comportamento

**Risposta corretta: b**

### **Domanda 19**

Una classe Java dichiarata abstract è visibile:

- a) Solo dalle classi che la estendono
- b) Da tutte le classi dello stesso package
- c) abstract non è un modificatore di visibilità
- d) Da qualsiasi classe

**Risposta corretta: c**

### **Domanda 20**

La differenza principale tra Thread e Processo è:

- a) I Thread sono più veloci dei Processi
- b) I Thread condividono le risorse dello stesso processo
- c) I Processi possono essere eseguiti in parallelo, i Thread no
- d) I Thread sono gestiti dal programmatore, i Processi dal sistema operativo

**Risposta corretta: b**

### **Domanda 21**

TornadoVM permette di:

- a) Compilare Java in eseguibili nativi più veloci di GraalVM
- b) Eseguire codice Java su hardware eterogeneo come GPU e FPGA
- c) Gestire stream reattivi su sistemi distribuiti
- d) Ottimizzare automaticamente l'uso della memoria nelle applicazioni Java

**Risposta corretta: b**

### **Domanda 22**

KotlinNative si differenzia da GraalVM perché:

- a) Supporta solo Kotlin mentre GraalVM supporta tutti i linguaggi JVM
- b) Si appoggia ad LLVM per produrre eseguibili direttamente dal codice Kotlin
- c) Ha prestazioni sempre superiori agli eseguibili GraalVM
- d) Non richiede modifiche al codice sorgente per la compilazione nativa

**Risposta corretta: b**

### **Domanda 23**

Un programma è "logicamente monotono" se:

- a) Esegue sempre le stesse operazioni nello stesso ordine
- b) All'aumentare della dimensione del problema il risultato non cambia
- c) Utilizza solo operazioni matematiche crescenti
- d) Non presenta cicli o ricorsioni nella sua struttura

**Risposta corretta: b**

### **Domanda 24**

Quale vantaggio si può ottenere dall'adottare un framework per la costruzione di applicazioni distribuite?  
[Scegli 1 o più alternative]

- a) Maggiore sicurezza perché i dettagli sono gestiti da persone più competenti
- b) Aggiornamento continuo che apporta benefici automaticamente
- c) Estrema efficienza nello sfruttare le peculiarità dell'hardware
- d) Facilità di realizzazione perché i protocolli di comunicazione sono nascosti da API di livello più elevato
- e) Facilità di realizzazione perché le parti più strutturali sono già implementate

**Risposte corrette: a, b, d, e**

### **Domanda 25**

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

- a) L'aumento dei nodi della rete ha compensato il miglioramento tecnologico
- b) Le tecnologie hanno eliminato il problema ma la concorrenza l'ha reintrodotto
- c) Dipende da una grandezza fisica che non può essere eliminata
- d) I protocolli di rete moderni introducono overhead maggiori

**Risposta corretta: c**

---

## **APPELLO 6 - Approfondimenti e Casi Pratici**

### **Domanda 26**

Nel confronto tra paradigmi di programmazione, quale caratteristica distingue la programmazione funzionale?

- a) Utilizza preferibilmente dati immutabili
- b) Prevede sempre l'esecuzione parallela
- c) Non supporta la ricorsione
- d) Richiede necessariamente un garbage collector

**Risposta corretta: a**

### **Domanda 27**

Quando si implementa uno Stream personalizzato, Spliterator fornisce:

- a) Un meccanismo per dividere l'iterazione in vari rami paralleli
- b) Un modo per ottimizzare automaticamente le operazioni terminali
- c) Una strategia per ridurre il consumo di memoria
- d) Un sistema per gestire eccezioni durante l'elaborazione

**Risposta corretta: a**

### **Domanda 28**

OpenLiberty InstantOn si differenzia da CRaC perché:

- a) Utilizza la compilazione AOT invece del salvataggio dello stato JVM
- b) Lavora a livello di processo di sistema operativo invece che dall'interno della JVM
- c) È compatibile con tutti i sistemi operativi
- d) Produce eseguibili di dimensioni inferiori

**Risposta corretta: b**

### **Domanda 29**

Il Subject nelle Reactive Extensions:

- a) Può solo consumare Observable senza produrre eventi
- b) Può solo produrre eventi senza consumare Observable
- c) Può sia consumare che produrre eventi, comportandosi da Observable e Subscriber
- d) È utilizzato esclusivamente per gestire gli errori negli stream reattivi

**Risposta corretta: c**

### **Domanda 30**

In un framework serverless, il principale problema della JVM tradizionale è:

- a) L'incompatibilità con i container Docker
- b) Il tempo di startup elevato per servizi che esistono solo per una richiesta
- c) L'impossibilità di gestire richieste HTTP concorrenti
- d) La mancanza di supporto per la programmazione asincrona

**Risposta corretta: b**

---

### **Soluzioni Riassuntive**

**Appello 1: b, b+c, c, a, c**

**Appello 2: c, c+d+e, d, b, b**

**Appello 3: c, c, b, b, b**

**Appello 4: a, a, b, c, b**

**Appello 5: b, b, b, a+b+d+e, c**

**Appello 6: a, a, b, c, b**