Test di Daniela Gangemi.

1. Descrivere una classe;  
   Risposta: una classe è un tipo di dato che sta alla base del paradigma ad oggetti. Essa è formata da attributi e metodi. Gli attributi descrivono le proprietà della classe mentre i metodi descrivono le azioni che le istanze della classe possono eseguire. Le istanze sono degli oggetti che hanno la stessa struttura della classe. Questo significa che essi possiedono gli stessi attributi e possono utilizzare gli stessi metodi.
2. Spiegare cosa sono i metodi e spiegare la differenza tra overloading e override;

Risposta: I metodi sono delle funzioni che descrivono le azioni che un oggetto può compiere. Ogni metodo è univoco ed è descritto dalla sua firma (signature) che è data dalla combinazione del nome del metodo e dai suoi parametri. Tramite l’overloading è possibile creare più metodi che abbiano lo stesso nome ma con parametri diversi. Un’applicazione di overloading riguarda il costruttore di una classe, il quale ha il compito di inizializzare gli oggetti. L’override, invece, permette di sovrascrivere metodi già definiti. Questo può capitare nel caso in cui volessimo implementare un’interfaccia (ossia un tipo di dato formato solo da firme) oppure se volessimo sovrascrivere un metodo ereditato da una classe padre.

1. Descrivere il funzionamento del Garbage Collection in Java e spiega l'importanza di questo meccanismo per la gestione della memoria.  
   La Garbage Collector è un meccanismo che si occupa della memoria tenendo traccia delle parti utilizzate. In particolare, esso si occupa di effettuare cicli di pulizia che permettono di deallocare la memoria eliminando le variabili non utilizzate. Grazie alla Garbage Collector, in Java non è necessario esplicitare alcun metodo distruttore. La Garbage Collector si occupa anche della frammentazione dell’Heap. L’Heap è un’area di memoria in cui risiedono gli oggetti istanziati. Essa si divide in Young Generation Heap e Old Generation Heap; nel primo vengono mantenute le nuove istanze mentre nel secondo quelle più vecchie. A sua volta, lo Young Generation Heap contiene l’Eden space e il Survivor space. La Garbage Collector effettua dei cicli di pulizia e sposta gli oggetti prima dall’Eden space al Survivor space e poi dal Survivor space all’Old Generation Heap. Nel momento in cui gli oggetti non vengono più utilizzati, la Garbage Collectore dealloca la pozione di memoria ad essi destinata.
2. Spiegare le differenze tra i le variabili (locali, di istanza, di classe, parametri). Spiegare il concetto di istanza e cosa cambia quando si utilizza il modificatore di accesso static.   
   Risposta: Le variabili locali sono definite all’interno di un metodo e il loro ciclo di vita coincide con esso. Le variabili di istanza sono definite all’interno di una classe ma si trovano al di fuori dei metodi della stessa. Le variabili di classe sono delle variabili di istanza ma utilizzano il modificatore “static”. Infine, i parametri sono variabili passate ai metodi, cioè vengono dichiarati dentro le parentesi tonde del metodo. I parametri si dividono, a loro volta, in formali e attuali. I parametri formali sono tipizzati e possono essere utilizzati all’interno del metodo mentre i parametri attuali sono quelli passati al metodo al momento della chiamata. Un’istanza è un oggetto di una classe specifica. Esso viene creato utilizzando la keyword “new” seguito dal nome della classe che si vuole istanziare. Il modificatore static può essere utilizzato su variabili e metodi e fa si che questi vengano caricati in memoria insieme alla classe che li definisce, appartenendo così alla classe stessa e non all’istanza.
3. Spiegare i tipi di dati primitivi e il casting   
   Risposta: I tipi di dati primitivi utilizzano una quantità di memoria ben definita e hanno dei valori di default. Ne esistono 8 tipi: boolean, byte, char, short, int, float, double, long.   
   - Il tipo boolean restituisce un valore tra “true” e “false”, occupa 1 bit di memoria e il suo valore di default è false.   
   - Il byte restituisce valori interi tra -128 e 127, utilizza 8 bit di memoria (uno per il segno e 7 per il valore) e il suo valore di default è 0.   
   - Il char è un carattere unicode, utilizza 16 bit di memoria e il suo valore di default è 0 in unicode.   
   - Lo short è un tipo di dato intero due volte più piccolo del tipo int, utilizza 16 bit di memoria e il suo valore di default è 0.   
   - L’int restituisce un intero che va da circa -2 miliardi a + 2 miliardi che utilizza 32 bit di memoria e il suo valore di default è 0.   
   - Il float restituisce un numero a virgola mobile a precisione singola, utilizza 32 bit di memoria e il suo valore di default è 0f.   
   - Il double è un tipo di dato a virgola mobile con precisione doppia, utilizza 64 bit e il suo valore di default è 0.0d.   
   - Il long restituisce un intero di grandi dimensioni, utilizza 64 bit di memoria e il suo valore di default è 0L.   
   Il casting è un’operazione che permette di passare da un tipo di dato all’altro. Esistono due tipi di casting: implicito ed esplicito. Quello implicito avviene senza alcuna dichiarazione mentre quello esplicito richiede una dichiarazione. È possibile fare il cast implicito quando il dato iniziale (da convertire) ha un range di valori incluso nel tipo di dato finale. Viceversa, è necessario fare il cast esplicito, che potrebbe portare a perdita di informazioni.

* Qual è il principale scopo di una classe in Java?

A) Implementare algoritmi complessi

B) Fornire un modello per la creazione di oggetti

C) Gestire eccezioni durante l'esecuzione del programma

D) Ottimizzare le performance del codice

* Quale delle seguenti affermazioni è vera riguardo a String e StringBuilder in Java?

A) String è immutabile, mentre StringBuilder è mutabile

B) Entrambe sono immutabili

C) Entrambe sono mutabili

D) String e StringBuilder sono sinonimi

* Qual è il concetto principale dell'ereditarietà in Java?

A) Creare oggetti senza dover dichiarare una classe

B) Permettere a una classe di ereditare le proprietà e i metodi di un'altra classe

C) Limitare l'accesso alle variabili di istanza

D) Creare classi che non possono essere istanziate

* Quando è appropriato utilizzare i getter e i setter in una classe Java?

A) Mai, poiché violano il principio dell'incapsulamento

B) Solo quando si lavora con classi astratte

C) Per fornire un modo controllato per accedere e modificare i dati di una classe

D) Esclusivamente in classi con membri statici

* In che modo StringBuilder è più efficiente di String quando si manipolano grandi quantità di dati?

A) StringBuilder è immutabile

B) StringBuilder è thread-safe

C) StringBuilder utilizza meno memoria grazie alla sua immutabilità

D) StringBuilder è mutabile, consentendo modifiche dirette senza creare nuove istanze

* Quale delle seguenti dichiarazioni è falsa riguardo alla classe Object in Java?

A) Tutte le classi Java ereditano implicitamente dalla classe Object

B) La classe Object contiene i metodi equals() e hashCode()

C) La classe Object è final e non può essere estesa

D) La classe Object fornisce il metodo toString() che può essere sovrascritto

* Qual è il principale vantaggio dell'utilizzo di modificatori di accesso in Java?

A) Aumentare la complessità del codice

B) Migliorare la leggibilità e la manutenibilità del codice

C) Ridurre le performance del programma

D) Eliminare la necessità di ereditarietà

1. Scrivere un metodo che, dati un carattere c ed una stringa s, restituisce true se c occorre in s, false altrimenti.  
     
   Immagine che contiene testo, schermata, Carattere, design

   Descrizione generata automaticamente  
     
   Commento: il metodo getOccurrance() prende in ingresso due parametri (una stringa e un carattere). Tramite un ciclo for, scorre tutti i caratteri che compongono la stringa, li compara con il carattere di cui vogliamo valutare l’occorrenza e, se la condizione è vera, interrompe il ciclo e ritorna true. Qualora la condizione non si verificasse, il metodo ritorna false.
2. Scrivere un metodo che, dati un carattere c ed una stringa s, restituisce il numero delle occorrenze di c in s.  
     
   Immagine che contiene testo, schermata, Carattere

   Descrizione generata automaticamente  
     
   Commento: il metodo getOccurranceNumber() prende in ingresso due parametri (una stringa e un carattere). Per prima cosa viene inizializzato un contatore a 0. Questa variabile serve per tenere conto di quante volte c occorre in s. Tramite un ciclo for, scorre tutti i caratteri che compongono la stringa, li compara con il carattere di cui vogliamo valutare l’occorrenza e, se la condizione è vera, incrementa il contatore. Il metodo ritorna il contatore che corrisponde al numero di occorrenze di c in s.
3. Scrivere un metodo in linguaggio Java che data una stringa s e due caratteri c1 e c2 determini se il numero di occorrenze di c1 in s sia uguale o meno al numero di occorrenze di c2.  
     
   Immagine che contiene testo, schermata, software

   Descrizione generata automaticamente

Commento: il metodo checkEqualOccurrance() prende in ingresso tre parametri (una stringa e due caratteri). Per prima cosa vengono inizializzat due contatori a 0. Queste variabili servono per contare quante volte c1 e c2 occorrono in s. Tramite un ciclo for, scorre tutti i caratteri che compongono la stringa, li compara sia con c1 che con c2 (in due controlli separati) e, se la condizione è vera, incrementa il rispettivo contatore. Infine, il metodo controlla se il numero delle occorrenze di c1 e c2 sia uguale, ritornando true se la condizione è verificata. Se non entra nell’if, il metodo ritorna false.