## Test di: Federico Mollica

1. **Descrivere una classe;**

Una classe è una rappresentazione astratta di un’entità reale o immaginaria, è composta dalle sue azioni dette metodi e dalle sue caratteristiche dette attributi. Una classe può essere ereditata da altre classi dette “figlie”, seguendo ad uno schema ad albero possiamo dire che le classi poste in alto sono classi generalizzanti, invece quelle poste in basso specializzanti.

1. **Spiegare cosa sono i metodi e spiegare la differenza tra overloading e override;**

Per overload si intende il processo di riscrittura di un metodo mantenendo inalterata la firma ma cambiando il suo comportamento. Per override si intende la riscrittura di un metodo cambiando anche i parametri formali

1. **Descrivere il funzionamento del Garbage Collection in Java e spiega l'importanza di questo meccanismo per la gestione della memoria.**

Java alloca nell’heap memory ogni istanza di una classa, quindi un oggetto, ma non permette la gestione diretta della memoria fornendo un distruttore. Quindi il Garbage Collector si occupa di eliminare dalla memoria gli oggetti con verifiche periodiche in modo tale da rimuovere tutti gli oggetti inutilizzati. Nel caso in cui la memoria sia quasi piena viene effettuato un ciclo di pulizia più profondo e più impattante nell’utilizzo delle risorse. Invece se la memoria messa a disposizione dalla jvm dovesse esaurirsi si provocherebbe un errore di OutOfMemory.

1. **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.**

Le variabili locali sono variabili che hanno il loro scope (quindi il loro “ciclo di vita”) legato al metodo o al costrutto che le dichiara e vengono allocate nella memoria statica detta stack.

Le variabili di istanza sono variabili legate alla singola istanza (quindi allocazione) di una classe. Le istanze di una classe sono tutti quegli attributi non static che appartengono ad uno oggetto e che non sono condivise con le altre istanze.

Le variabili di classe, invece, a differenza delle variabili di istanza si indicano con il modificatore static e sono delle variabili condivise tra tutte le istanze di una determinata classe.

I parametri (divisi tra attuali e formali) sono le variabili che vengono inserite tra le parentesi tonde di un metodo quando viene invocato. I parametri formali sono le variabili che verranno inviate al metodo e che dovranno essere dichiarate nella firma dello stesso, ad es: a.met(b,c) <- b,c parametri attuali, public void met(int b, int c) <- b,c parametri formali.

Per istanza di una classe si intende la creazione effettiva di un oggetto all’interno del programma, allocando quindi nell’heap, attraverso l’operatore new.

1. **Spiegare i tipi di dati primitivi e il casting**

Per tipo primitivo si intende quel tipo di dato semplice il quale non ha quindi nessun tipo di metodo associato a differenza dei tipi derivati come String. I tipi primitivi possono essere ad esempio: int, long, double, short, float, char, boolean.

Per casting si intende l’operazione di conversione di tipo di una variabile, esistono due tipi di casting, quello esplicito e quello implicito. In quello esplicito si scrive chiaramente a quale tipo si voglia fare la conversione. In quello implicito invece la conversione viene fatta in automatico senza necessità di indicare il tipo di festinazione. Ad es:

char c = ‘c’;

int b = (int) c; <- cast esplicito da char ad int

double d = b; <- cast implicito da int a double

* 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?

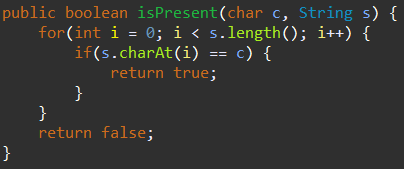
A) Aumentare la complessità del codice

B) Migliorare la leggibilità e la manutenibilità del codice

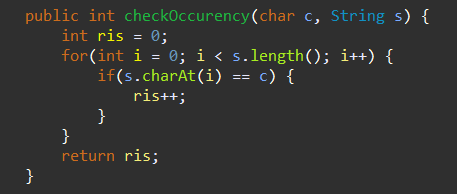
C) Ridurre le performance del programma

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.



1. Scrivere un metodo che, dati un carattere c ed una stringa s, restituisce il numero delle occorrenze di c in s.



1. 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.

