TEST (Parte1) di 28/03/2024: **FILIPPO ROTOLO**

1. **Descrivere una classe;**  
   Una classe è uno dei concetti principali della programmazione in Java.  
   Una classe è rappresenta da un blocco di codice che definisce un modello, ovvero una rappresentazione di una entità del mondo reale, per la creazione di oggetti (cioè istanze di una classe). Una classe definisce stato e comportamento di entità simili tra loro (ad es: l’entità Persona→ classe Persona)
2. **Spiegare cosa sono i metodi e spiegare la differenza tra overloading e override;**  
   i metodi definiscono il comportamento degli oggetti di una classe, hanno un valore di ritorno, un nome e una lista di parametri e l’insieme di queste clausole definisce la cosiddetta firma del metodo.  
   Si parla di override quando all’interno di una stessa classe si definiscono metodi con lo stesso nome ma firme diverse.  
   Si parla invece di override quando si ridefinisce il comportamento di un metodo, ad esempio per un metodo ereditato da una superclasse per la quale vogliamo specificarne il comportamento.
3. **Descrivere il funzionamento del Garbage Collection in Java e spiega l'importanza di questo meccanismo per la gestione della memoria.**  
   Il Garbage Collector è uno dei componenti principali della JVM e permette di deallocare automaticamente la memoria occupata da oggetti non piu in uso.   
   Questo meccanismo permette di rimuovere la responsabilità della gestione della memoria direttamente al programmatore, permettendo allo stesso tempo una migliore gestione delle risorse.
4. **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.**-Una variabile è detta locale quando la sua dichiarazione e inizializzazione avviene all’interno di un blocco di codice (generalmente all’interno dei metodi); terminata l’esecuzione del blocco la variabile verrà deallocata e quindi questa cesserà di esistere. Una variabile locale non è visibile al di fuori del blocco in cui è stata dichiarata.  
   -Una variabile di istanza è una variabile definita all’interno di una classe ma fuori dai metodi e in particolare appartiene ad un oggetto che è istanza di una classe.  
   -Una variabile di classe è una variabile definita come “static”, quindi appartenente direttamente alla classe e non alle singole istanze.  
   -Per istanza si intende l’allocazione di memoria durante la “creazione” di un oggetto.  
   Una istanza di una classe si effettua con l’operatore new (Es Persona p = new Persona(); dove p sarà un oggetto istanza della classe Persona)  
   -Il modificatore static permette di rendere trasversale l’utilizzo di metodi e/o attributi dichiarati all’interno di una classe.   
   Con questo modificatore è possibile utilizzare metodi e/o attributi di una classe SENZA la necessità di instanziare un oggetto della classe con l’operatore new, ma richiamando semplicemente il nome della classe seguito dall’operatore dot ( NomeClasse.metodo / NomeClasse.attributo)
5. **Spiegare i tipi di dati primitivi e il casting**In java sono presenti i dati di tipo primitivo, che rappresentano variabili di tipo numerico, di tipo booleano o caratteri. Per questo motivo Java è detto “orientato” agli oggetti e non a oggetti puro.  
   In particolare esistono i tipi byte, short, int, long per la rappresentazione dei numeri interi, float e double per la rappresentazione dei numeri in virgola mobile, boolean per rappresentare lo stato vero/falso e char per i caratteri (Ascii Unicode).  
   Il casting rappresenta la forzatura da un tipo di dato ad un altro. Può essere implicito se passiamo un tipo di dato piu piccolo ad un tipo di dato piu grande, esplicito quando passiamo un tipo di dato piu grande ad uno piu piccolo. Il casting esplicito si indica con “(tipo\_dato\_primitivo) nomeVariabile”

* 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.
2. Scrivere un metodo che, dati un carattere c ed una stringa s, restituisce il numero delle 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.

/\*1)

\* Scrivere un metodo che, dati un carattere c ed una stringa s,

\* restituisce true se c occorre in s, false altrimenti.\*/

public boolean trovaOccorrenza(String s, char c){

for(int i=0; i<s.length();i++){

if(s.charAt(i)==c) {

return true;

}

}

return false;

}

/\*2)

public int contaOccorrenze(String s, char c){

int counter=0;

for(int i=0; i<s.length();i++){

if(s.charAt(i)==c) {

counter++;

}

}

return counter;

}

/\*3)

public boolean contaOccorrenze(String s, char c1, char c2){

int counter1=0;

int counter2=0;

for(int i=0; i<s.length();i++){

if(s.charAt(i)==c1) {

counter1++;

}

if(s.charAt(i)==c2) {

counter2++;

}

}

if(counter1<=counter2){

return true;

}

return false;

}