Umberto Cangelosi

1-Descrivere una classe;

Una classe e’ un tipo di dato che sta alla base del paradigma di programmazione ad oggetti.

E’ una modo di rappresentare ad alto livello un insieme di oggetti, di istanze che condividono attributi quindi le caratteristiche in comune degli oggetti e i metodi che ne descrivono il comportamento. Quando si crea un oggetto ‘fisico’ di una classe, esso prende il nome di istanza della classe. Puo essere una classe astratta se si definiscono solo i metodi senza implementarli. Una classe si crea mediante la keyword ‘class’.

2-Spiegare cosa sono i metodi e spiegare la differenza tra overloading e override;

I metodi sono implementazioni che rappresentano il comportamento degli oggetti di una classe.

Permettono di riutilizzare porzioni di codice senza doverle reimplementare ogni volta.

Di un metodo si deve definire sicuramente i modificatori di visibilita’ o accesso, di solito “public”, si possono inserire altri modificatori come final che fa si che il metodo non possa essere sovrascritto da sottoclassi, oppure “static”, in quel caso il metodo diventa un metodo di classe e non relativo alla singola istanza, ed e’ accessibile direttamente dalla classe. In quel caso puo ricevere come parametro solo variabili anche esse “static”. Successivamente si definisce il tipo di return, che puo’ essere anche void se non abbiamo un return, il nome del metodo, i parametri tra parentesi tonde, e le istruzioni tra parentesi graffe.

Quando si riscrive il comportamento un metodo con un nome gia’ esistente all’interno di una classe (nel caso in cui si tratta di una metodo proveniente da una classe estesa non deve avere il modificatore ‘final’), si presenta un caso di override, in cui il metodo presentera’ in nuovo comportamento. E’una delle basi del concetto di polimorfismo.

L’overloading si presenta quando ci sono piu’ metodi con lo stesso nome, ma con differente firma, quindi metodi che hanno lo stesso nome ma cambiano i parametri (esattamente devono avere i tipi dei parametri differenti per essere riconosciuti come differenti).

3- Il garbage collector e’ un componente della JVM che permette di liberare memoria allocata nello heap della memoria, liberando spazio quando oggetti e variabili non sono piu’utilizzate all’interno del programma. Esso opera mediante cicli di controllo dove avviene una rilevazioni degli oggetti e variabili all’interno dell’heap della memoria per spostarle in altre zone della memoria deallocando lo heap. A un certo punto eliminerà definitivamente la memoria.Per via del garbage collector in java non c’e bisogno dei distruttori.

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.

Le variabili locali sono definite all’interno di un metodo, ed esistono nella memoria solo fino al termine del metodo.

Le variabili di istanza sono definite all’interno di una classe, ma fuori dai metodi. Si dichiarano con accesso private e questo concetto e’ alla base dell’incapsulamento

Le variabili di Classe sono variabili di istanza con il modificatore static, che fa si che quella variabile sia unica nella memoria per tutte le istanze, e sia utilizzabile dai metodi statici.

I parametri sono i primitivi o i riferimenti che passiamo ad un metodo tra le parentesi tonde.

Un'istanza di una classe(vedi risposta1) è un oggetto che è stato creato utilizzando il costruttore di quella classe. In altre parole, un'istanza rappresenta uno oggetto specifico di una classe che contiene i suoi attributi e comportamento (i metodi associati).

Quando viene creata un'istanza di una classe, viene allocato spazio di memoria

5-i dati di primitivi sono:

boolean: tipo di dato che rappresenta true o false, e occupa un bit di memoria

byte: rappresenta valori tra -128 e 128,

char: rappresenta i caratteri in unicode, occupa 16bit

float: rappresenta numeri in virgola mobile si usa la “f”in fase di inizializzaizione, occupa una memoria di 32 bit, ed e’ sconsigliato quando la precisione e’ molto importate. (esempio quando rappresentiamo ordini i grandezza molto piccoli oppure piccole variazioni sono molto rilevanti come nell’ambito della finanza).

double: simile al float ma con una doppia precisione, occupa 64 bit di memoria e si usa in fase di inizializzazione la lettera “d”.

int: si usa per rappresentare i numeri interi e occupa una memoria di 32 bit.

long: si usa per rappresentare numeri interi molto grandi, utilizza una memoria di 64 bit.

short: si usa per rappresentare numeri che non sono molto grandi, occupa una memoria di 16 bit.

e’molto importante scegliere con cura il tipo di dato per rappresentare i numeri perche’si puo’generare facilmente durante una fase di calcolo un errore di underflow o overflow.

5-il casting e’ l’operazione che permette di far passare una variabile da un tipo di dato a un altro.

Puo’ essere esplicito, quando indichiamo tra parentesi tonde prima della variabile da castare, il tipo di dato che vogliamo. Implicito quando il tipo di dato di partenza occupa una memoria minore o uguale al tipo di dato che vogliamo, in quel caso non dobbiamo esplicitare il casting ma avviene automaticamente.

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

B) Fornire un modello per la creazione di oggetti

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

A) String è immutabile, mentre StringBuilder è mutabile

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

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

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

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

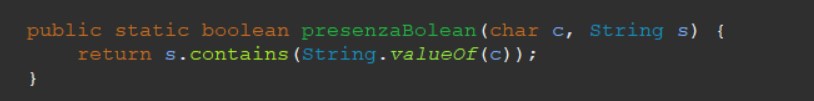
* In che modo StringBuilder è più efficiente di String quando si manipolano grandi quantità di dati?
* D) StringBuilder è mutabile, consentendo modifiche dirette senza creare nuove istanze
* Quale delle seguenti dichiarazioni è falsa riguardo alla classe Object in Java?

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

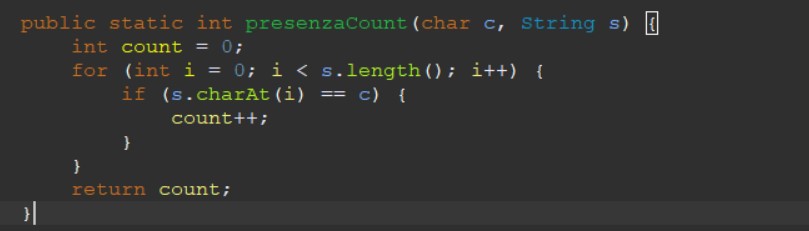
* Qual è il principale vantaggio dell'utilizzo di modificatori di accesso in Java?

B) Migliorare la leggibilità e la manutenibilità del codice

esercizio1



esercizio2



esercizio3

