**Test Object Oriented**

1. *Due oggetti (istanze della stessa classe) possono avere attributi con lo stesso valore?*

* SI
* NO

1. *Specificare vari approcci nel mondo di SW Commerciale.*

Nel mondo dei software commerciali ci sono due approcci, cioè quello del software proprietario e quello del software libero. La commercializzazione del primo prevede può essere fatta sotto forma di bene proprietario mentre il secondo no.

1. *Specificare la fabbricazione di un’auto con un approccio Object Oriented.*

Per prima cosa si definisce la classe Auto che descrive gli attributi dell’auto e i vari metodi che deve possedere (il costruttore, i vari getter/setter, etc…). Una volta fatto ciò è possibile istanziare oggetti Auto utilizzando il costruttore che sia definito oppure quello di default

1. *Che cosa è un diagramma UML?*

Un diagramma UML è una rappresentazione della classe con i suoi attributi e metodi. Viene usato in fase di progettazione

1. *È necessario distruggere gli oggetti creati se non sono utilizzati?*

* SI
* NO

1. *Com’è gestita la area di memoria per due oggetti che hanno gli stessi attributi?*

Anche se i due oggetti hanno gli stessi attributi, vengono posti in due aree di memoria diversi: questo perché vengono visti come due entità diverse.

Es:

|  |  |
| --- | --- |
| auto1 | auto2 |
| Marca: Fiat | Marca: Fiat |
| Modello: Punto | Modello: Punto |

1. *Come è possibile accedere ad un oggetto?*

Per accedere ad un’oggetto bisogna prima di tutto creare una sua istanza, tramite la keyword *new*. Una volta faccio ciò dipende dai modificatori di accesso con cui è stato progettato: se per esempio ai suoi attributi è stato applicato il modificatore di accesso *private*, per accedere a questi è necessario usare i metodi getter/setter. Per accedere poi ai vari metodi si usa l’operatore “.”.

1. *Il seguente codice permette di accedere ad una funzione della classe Albero?*

Albero albero;

albero. setNomeAlbero;

* Si
* No

1. *Che cos’ è un costruttore?*

In una classe il costruttore è il metodo che permette l’istanziazione di essa. Viene usato quando si definisce una nuova istanza dell’oggetto e viene affiancato alla keyword *new*. Il suo compito è quello di associare valori agli attributi dell’istanza, Ci possono essere diversi tipi, come il costruttore di default che non fa altro che associare dei valori di default agli attributi e non ha bisogno di essere definito, oppure costruttori costruiti ad hoc dal programmatore che hanno la possibilità di accettare dei parametri che gli vengono passati in fase di costruzione.

|  |
| --- |
|  |

1. *Data l’istruzione di seguito, distinguere fra classe, variabile, costruttore*

**Fiore rosa = new Fiore ();**

Classe Fiore

Variabile rosa

Costruttore Fiore()

1. *Due o più variabili possono puntare allo stesso oggetto?*

* Si
* No

1. *Che cosa è il garbage collector e come si invoca?*

Si tratta di un thread che ha il compito di liberare la memoria heap distruggendo oggetti che non vengono più utilizzati (cioè quelli a cui non fa più riferimento nessun programma). Viene invocato automaticamente dal compilatore.

1. *Quali classi possano accedere ad un package?*

public XXX class.

SI

NO

public final XXX class.

NO

SI

private XXX class.

NO

SI

protected XXX class.

NO

SI

1. *Se bisogna invocare uno o più metodi o funzioni di una classe in un’altra classe di cui anche il package è diverso, si può accedere alle classi senza importare il package relativo alla classe stessa?*
2. *Descrivere cosa vuole dire il codice seguente*

import package.class1 viene importato solo la class1 del package

import package.\* viene importato tutto il package

1. *è possibile estendere una classe final?*

* Si
* No

1. *Quale e la risposta corretta?*

Public static final String SYNCLAB = SYNCLAB;

* Attibuto
* Variabile
* Classe
* Costante
* Classe final

1. *Cos’è una classe interna? Ne possiamo modificare la visibilità?*

Una classe interna è una classe che viene definita all’interno di un’altra classe. La sua visibilità può essere modificata tramite la keyword static che la rende di fatto una classe regolare