

Test di fine settimana – Week 3

Nome Valentina

Cognome Commerci

Data 4/6/21

Leggete attentamente ogni domanda e argomentare quanto più possibile fornendo anche degli esempi. ATTENZIONE: Le domande a risposta multipla possono contenere più risposte corrette.

1. Elencare le principali differenze tra strutture e classi.

Entrambi classi e strutture possono definire data member, proprietà e metodi

ma:

- Le classi supportano costruttori e inizializzazione membri mentre le strutture no.
- Le classi supportano il metodo Finalize e l'ereditarietà, mentre le strutture no.
-
- Le classi sono reference type che vengono inizializzate nell'Heap, mentre le strutture sono dei Value Type inizializzati nello stack.

2. Nell'istruzione `Persona p = new Dipendente();` la variabile `p` si comporta in maniera polimorfa. Perché? Cosa si intende per comportamento polimorfico?

- Il polimorfismo ci concede di trattare l'istanza di un tipo come se fosse l'istanza di un altro tipo
- di conseguenza un oggetto può avere più comportamenti
- se ho una classe di base come `Persona` (con proprietà nome e cognome) e una sottoclasse `Dipendente` (con proprietà stipendio)
- la classe `Dipendente`, e quindi l'istanza creata e "stored" nella variabile `p` può essere sia trattata come tipo `Dipendente` ma anche come `Persona` generica
- ha quindi un comportamento duplice grazie al quale potrà essergli attribuito sia un nome che un cognome ed in aggiunta anche uno stipendio.

3. In che modo è possibile utilizzare la combinazione di keyword del linguaggio virtual e override? Fornire un esempio.

- quando una classe derivata deve poter provvedere una nuova implementazione di un metodo presente nella classe base
 - 1. questo metodo deve essere contrassegnato come virtual nella classe base
 - 2. e la classe derivata che voglia o debba fornire una implementazione sostitutiva della classe base deve marcare il metodo con override

(con virtual intendo che se una classe avesse una sottoclasse il metodo potrebbe essere sovrascritto da essa)
(se non virtual non viene specificato e tolgo anche override, verrà usato il metodo della classe base)

```
public abstract class strumento
{
    public abstract void Accorda();

    public virtual void PausaTempo(){
        //...
    }
}
```

Public class Piano : Strumento

```
{
    public override void Accorda (){
        //...
    }
    public override void PausaTempo (){
        //...
    }
}
```

4. Spiegare la differenza tra Array, Collection e Collezioni Generiche.

ARRAY

- raggruppa dati di un tipo stabilito (fortemente tipizzata)
- ha una misura definita
- non prevede nessun processo di boxing e unboxing

COLLECTION

- può raggruppare dati di tipo diverso
- non fortemente tipizzate

GENERICS

- Consente di scrivere classi e metodi indipendenti dal tipo
- permette di migliorare le performance
- evitiamo il problema di avere una lista di oggetti di tipo object per il quale avviene il processo di boxing e unboxing rischiando di causare problemi in fase esecuzione.