# Modulo 7

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Tiziana |
|  |  | Cognome | Camboni |
|  |  | Data | 12/03/2021 |

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. *Come avviene il lancio di un’eccezione a runtime ?*

*Il lancio di un eccezione viene fatta quando, identificata un’istruzione che non può essere eseguita, viene identificata come eccezione. Dopo esere stata identifica, se l’eccezione non è contenuta in un blocco try-catch, determina il terminamento dell’esecuzione del programma.*

1. *Quali sono le possibili cause di un’eccezione?*

*Le possibile cause di un eccezione possono essere:*

* *Operazioni artimetiche non ammesse (es.Divisioni per 0)*
* *Lettura di un file non trovato*
* *Quando si utilizza una matrice/array e l’indice è oltre la dimensione*
* *Lettura di caratteri non ideonei in un file*
* *Memoria terminata per eseguire una nuova scrittura*
* *Conversione esplicita tra tipi non compatibili*
* *Riferimento ad un valore nullo*

1. *Descrivere il costrutto formale per la gestione di una eccezione fornendo un esempio.*

*Costrutto try- catch:*

* + - 1. *Viene eseguito il codice del blocco try: il blocco può essere eseguito e terminato(-> punto 3) e generare un’eccezione( -> punto 2)*
      2. *Nel caso venga generata un’eccezione, viene cercato nei blocchi catch relativi al try di sopra, se uno di questi è adatto a gestire l’eccezzione.*
      3. *Infine si può avere un ulteriore blocco finally in cui è inserito un blocco di codice che viene sempre eseguito, sia che si sia terminato il blocco try, sia che si sia entrati nel catch.*

*Try*

*{*

*Addition(A, B);*

*}*

*Catch(exception e)*

*{*

*Console.writeline(“Non è stato possibile fare l’addizione”);*

*}*

*Public static int Addition(int a, int b)*

*{ c = a +b;*

*Return c; }*

1. *Quali sono i possibili casi in cui è necessario lanciare un eccezione da codice? Fornire un esempio.*
   * + 1. *Quando un metodo non può essere svolto (es. Incompatibilità dell’input)*
       2. *Chiamata inapropriata ad un oggetto(settare una proprietà impostata in sola lettura)*
       3. *Quando un argomento in input di un metodo può generare eccezione.*

*public static void PositiveAddition(int A, int B)*

*{*

*int c = A + B;*

*if (c<0)*

*{*

*throw new ArgumentException(" \nLa somma deve essere fatta di numeri posivi"); //se items è nullo allora fai nuovo argument*

*}*

*}*

*public static void Run()*

*{*

*try*

*{*

*PositiveAddition(-9, 3);*

*}*

*catch (Exception e)*

*{*

*Console.WriteLine("Si è verificato un errore" + e.Message);*

*}*

*}*

1. *Fornire una panoramica delle eccezioni più comuni e definire a cosa sono dovute.*

* *ArithmentAexception: errori durante calcoli aritmetici*
* *devideByZeroException: quando viene fatta una divisione per 0*
* *InvalidCastExeption: quindo non è possibile fare espicitamente una conversione tra tipi*
* *NullreferenceException: quando si fa riferimento ad un oggetto null*
* *OutOfMemoryException: quando non è possibile allocare memoria quando di vuole creare una nuova istanza di un oggetto.*

1. *Descrivere i ruoli che esistono all’interno di un progetto*

*Principalemnte all’interno di un progetto si possono individuare 6 ruoli:*

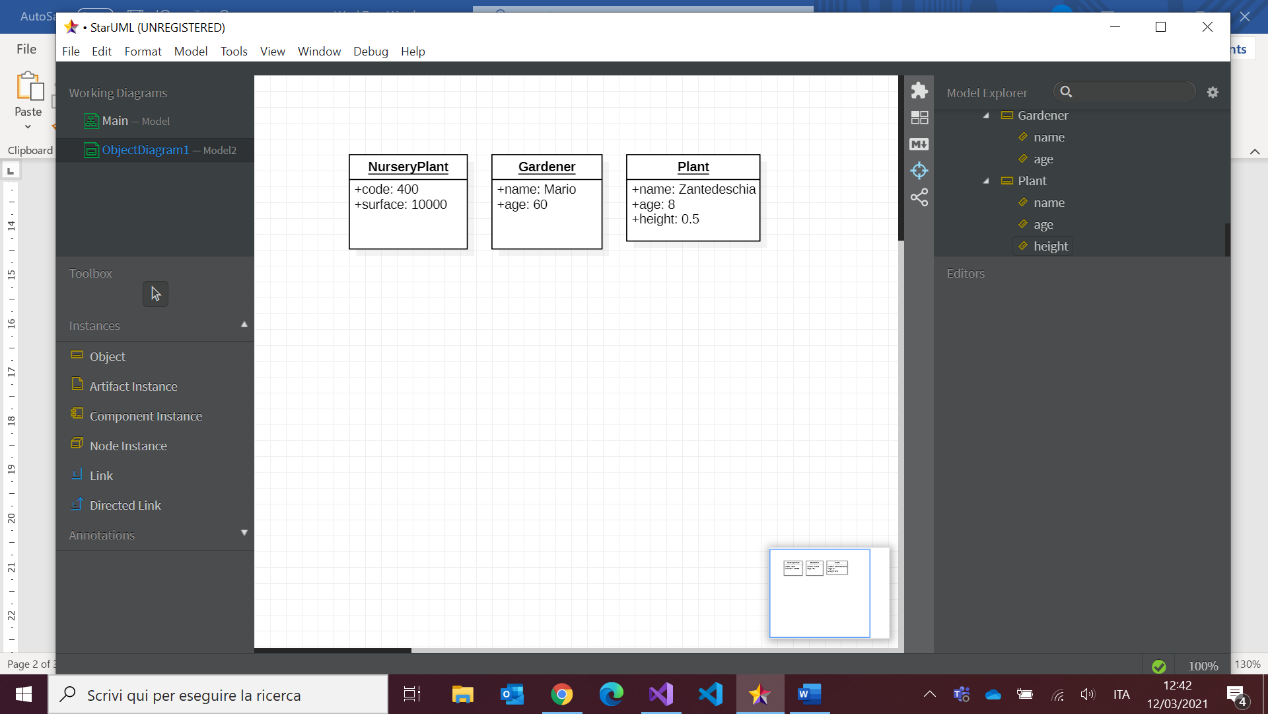
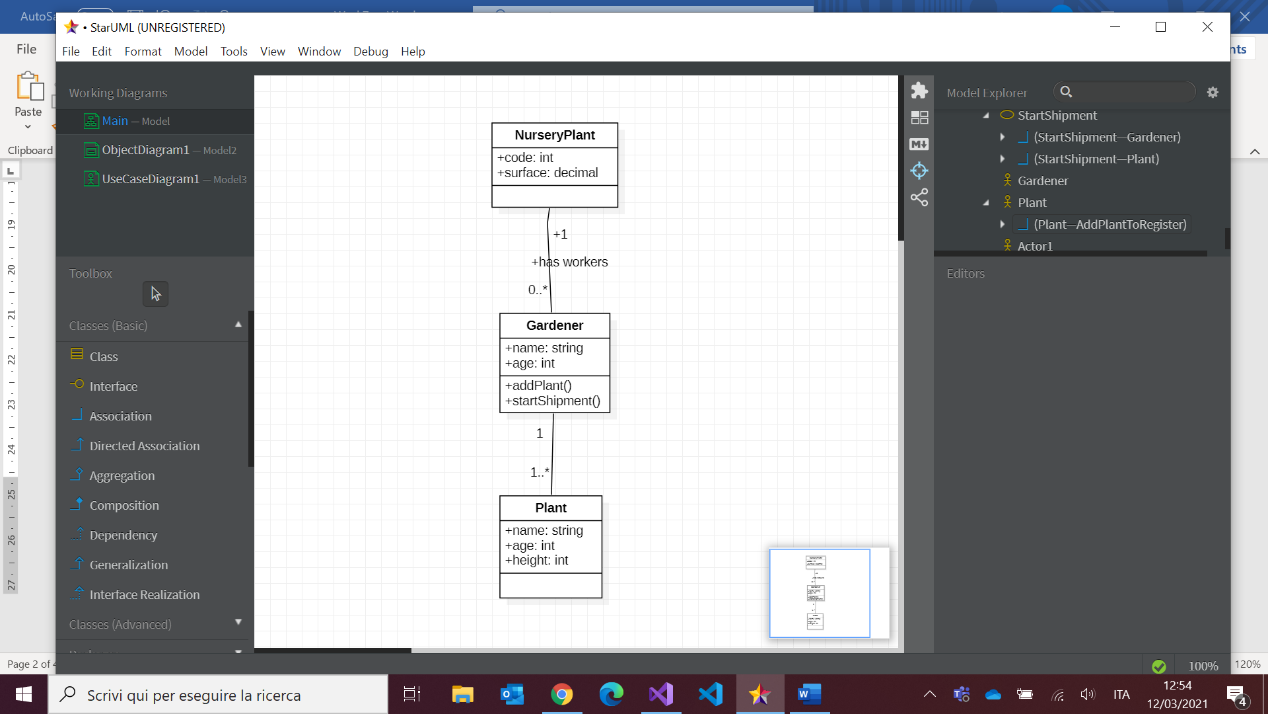
* + - 1. *Product owner: colui che ha la necessitrà /desidero di un prodotto con deteminate caratteristiche*
      2. *Business analyst: fiigura intermediaria che grazie a conoscenze sulla possibile realizzazione interpreta in modo piu tecnico le caratteristiche proposte dal product owner.*
      3. *Software Architet: identifica delle possibili soluzioni da base per gli sviluppatori*
      4. *Sviluppatori: sulla base dei progetti dell’architet, implementano e realizzano il sofware di progetto*
      5. *Team Quality Assurance: testano il prodotto per determinare se ha le funzionalità e la logca predisposta nella fase analisi*
      6. *Team operation: testano il prodotto finale per determinare che siano soddifatti i requisiti tecnici di esecuzione e/ lo testa in diversi ambienti*

1. *Dare una definizione di UML e spiegare perché si utilizza.*

*UML può essere definito com un linguaggio di modellazione orietato ad oggetti, in cui è possibile strutturare 13 diversi digrammi, ciascuno determinato da comportamento del prodotto(interazione tra classi, tra oggetti, dinamica temporale ecc.); quindi grazie all’impostazione di differenti diagrammi è possibile osservare diverse prospettive di un unico prodotto.*

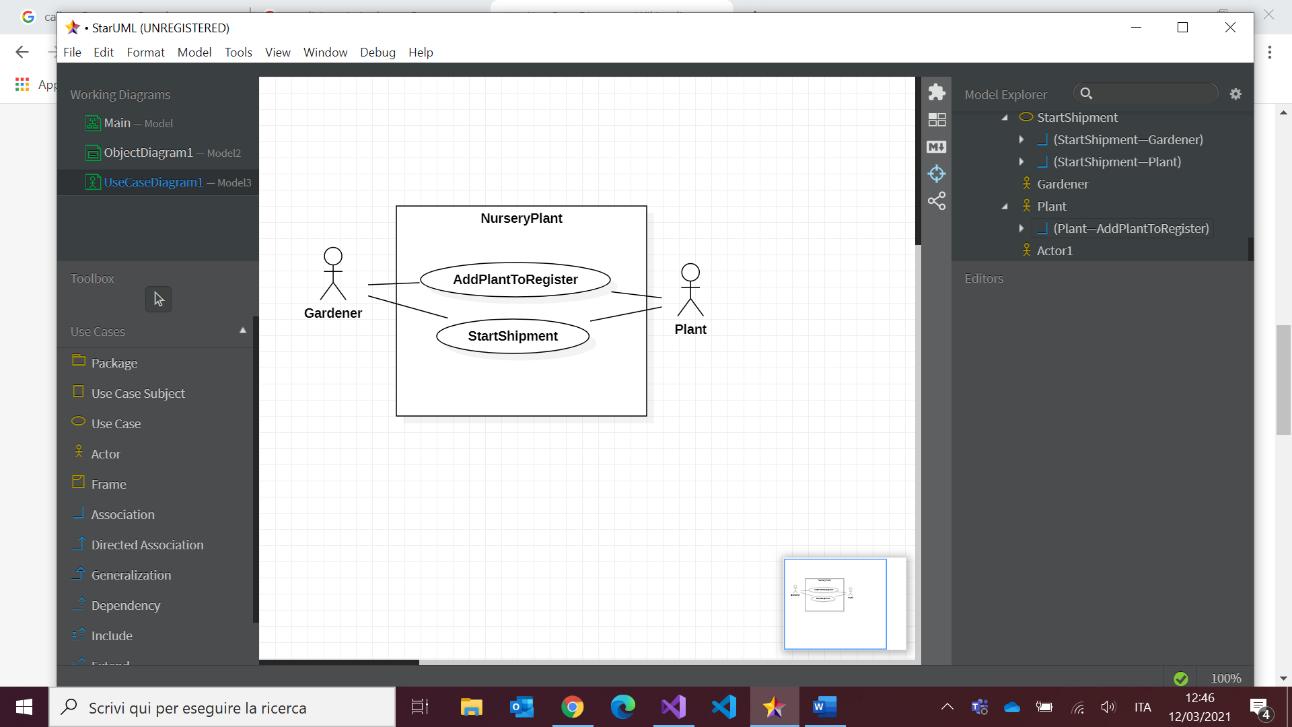
1. *Descrivere la differenza tra class diagram e object diagram descrivendo gli elementi di cui possono essere composti. Fornire un esempio.*

*Il class diagram e object diagram differiscono in quanto il primo esplicita le relazioni tra le classi individuate, i tipi delle propietà di queste e i metodi per ciascuna classe; nel secondo vengono rappresentati gli oggeti modo esemplificativo: vengono create quindi una o piu istanze per ogni oggetto, esplicitando a titolo di esempio il contenuto delle propeirtà.*



1. *Fornire un esempio di use case diagram. Quali elementi sono in gioco e qual è l’obiettivo del diagramma?*

*(Non usare gli esempi presentati in classe)*

*Lo use case diagram ha l’obiettivo mostrare le funzioni di un sistema mettendo in gioco gli “attori”, ovvero le persone fisicamente che interagisocno con il sistema proposto, e le possibili situazioni possibili.*

*Esercitazione pratica*

*UML*

*Data un’applicazione che recuperi da un database i dati relativi alla gestione di una scuola, fare un object diagram, un class diagram e un use case diagram tenendo conto che:*

*Ogni classe presente nella scuola è composta da studenti. Un insegnante può impartire lezioni di più materie ed essere assegnato a più classi.*

*Uno studente inoltre può essere valutato in più occasioni per materia.*

*Uno studente è definito da Nome, cognome e data di nascita.*

*Una classe è definita da un codice e un livello.*

*Un insegnante ha un nome, cognome e una data di nascita. Può inserire una votazione a sistema indicando lo studente, la materia e il giorno della prova.*

*EXCEPTION HANDLING*

1. *Dimostrare come gestire un’eccezione data dalla mancata connessione a un database filtrando il tipo per errore fatale.*
2. *Creare una custom Exception relativa a un utente non trovato. Dimostrarne l’utilizzo.*