# Modulo 7

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Valentyna |
|  |  | Cognome | Bogachova |
|  |  | 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 ?*
2. *Quali sono le possibili cause di un’eccezione?*
3. *Descrivere il costrutto formale per la gestione di una eccezione fornendo un esempio.*
4. *Quali sono i possibili casi in cui è necessario lanciare un eccezione da codice? Fornire un esempio.*
5. *Fornire una panoramica delle eccezioni più comuni e definire a cosa sono dovute.*
6. *Descrivere i ruoli che esistono all’interno di un progetto*
7. *Dare una definizione di UML e spiegare perché si utilizza.*
8. *Descrivere la differenza tra class diagram e object diagram descrivendo gli elementi di cui possono essere composti. Fornire un esempio.*
9. *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)*

Risposte

1. Le eccezioni vengono usate per indicare che si è verificato un errore durante l'esecuzione del programma. Vengono creati oggetti eccezione che descrivono un errore e quindi generati con la parola chiave [throw](https://docs.microsoft.com/it-it/dotnet/csharp/language-reference/keywords/throw). Il runtime cerca quindi il gestore di eccezioni più compatibile.
2. Quando il metodo non può completare la funzione, per esempio un parametro non valido. Quando si esegue una chiamata non appropriata a un oggetto, in base allo stato dell’oggetto(scrivere su un file di sola lettura). Quando un argomento verso un metodo genere un’eccezione. In questo cosa bisogna riprendere l’eccezione originale e creare un’istanza di ArgumentException.
3. usano un blocco [try](https://docs.microsoft.com/it-it/dotnet/csharp/language-reference/keywords/try-catch) per creare partizioni di codice in cui potrebbe essere rilevata un'eccezione. I blocchi [catch](https://docs.microsoft.com/it-it/dotnet/csharp/language-reference/keywords/try-catch) associati vengono usati per gestire tutte le eccezioni risultanti. Un blocco [finally](https://docs.microsoft.com/it-it/dotnet/csharp/language-reference/keywords/try-finally) contiene codice che viene eseguito indipendentemente dal fatto che venga generata un'eccezione nel try blocco, ad esempio il rilascio di risorse allocate nel try blocco. Un blocco try richiede uno o più blocchi catch associati, un blocco finally o entrambi.

try

{

}

catch (SomeSpecificException ex)

{

}

1. Abbiamo tre casi in cui dobbiamo lanciare un eccezione1. Quando un argomento verso un metodo genera un’eccezione: ad esempio viene passato un indice di un array non valido.  
   2. Quando la chiamata non appropriata ad un oggetto: ad esempio quando si prova a scrivere su un file di sola lettura  
   3. Quando un metodo non può completare la sua funzionalità perchè viene passato un parametro non valido.  
   ad esempio: una funzione prende come parametrouna stringa su cui deve fare dei calcoli  
   public static void CalcolaStringa(string nome){  
   try {  
   if(nome == nul) { throw new ArgumentException(“Il nome è voto”); }  
   catch (ArgumentException exc) {//la gestisco}
2. Fornire una panoramica delle eccezioni più comuni e definire a cosa sono dovute
   1. InvalidCastException: quando fallisce un cast tra due tipi diversi
   2. NullReferencesException : quando si cerca di fare dei calcoli su un oggetto nullo  
      - IOException: quando ci sono degli errori nel leggere/scrivere i dati   
      - DivideByZeroException: è un’eccezione matematica
   3. IndexOutOfRange : quando si esce dal range di un array  
        
      StackOverFlowException : quando lo stack delle chiamate è pieno

6.1.Product owner il proprietario del prodotto finale.2.Buisness Analysts è la figura che sta in mezzo tra owner del prodotto e il team tecnico.3.Software/solution Architect progetta i sistemi , decide il modo in cui gòi sviluppatori dovrebbero realizzare il sistema.4.Sviluppatori – riprendono i disegni funzionali e tecnici redatti dagli analisi e dall’arcitect.5.TeamQA(Quality Assurance) team di Testing , si assicura che il software scrtitto dal team di sviluppo corrisponda ai requisti definite in fase di analisi6.TeamOperation – prende il sofrware finite e testato.

7. UML, Unified Modeling Language, è un linguaggio semiformale e grafico (basato su diagrammi) per I specificare I visualizzare I realizzare I modificare I documentare gli artefatti di un sistema (solitamente un sistema software). Un artefatto è un qualunque prodotto tangibile del progetto: sorgenti, eseguibili, documentazione, file di configurazione, tabelle di database, benchmark

8.Un diagramma delle classi è un grafo compost da classi e relazioni.

Un diagramma degli ogetti è un grafo composto di classi(oggetti) e relazioni; esso è una istanza del diagramma delle classi. Gli elementi in cui possono essere composti sono i classi, interfacce, associazioni, relazione, invece alto ogetti, artifact instance, component instance.

9. I diagrammi dei casi d'uso descrivono le relazioni e le dipendenze tra i gruppi di casi d'uso e gli attori coinvolti nel processo. i diagrammi dei casi d'uso non intendono rappresentare un progetto e non possono descrivere le parti interne di un sistema. I diagrammi dei casi d'uso hanno lo scopo di facilitare l'interazione con i futuri utenti del sistema, con i clienti e sono particolarmente utili per determinare le caratteristiche richieste del sistema. In altre parole, i diagrammi dei casi d'uso indicano cosa dovrebbe fare il sistema senza specificare i metodi effettivi da utilizzare.

*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.*