# Modulo 7

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Martina |
|  |  | Cognome | Libreri |
|  |  | 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 a runtime viene gestito con il blocco catch nel quale viene utilizzata un’istanza di tipo Exception o una sua derivata (systemException, applicationException), nel caso di più controlli bisogna verificare l’ordine con il quale vengono inseriti. Questa classe ha diverse proprietà come message che una volta che viene eseguito il runtime mostrerà a console la descrizione dell’errore, proprietà targetSite, che indica il metodo che ha sollevato l’eccezione ...*

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

*Le possibili cause di un eccezione sono: un’autorizzazione fallita, problemi dell’hardware e delle risorse non disponibili oltre ad eventuali errori da parte dello sviluppatore.*

*Errori da arte dello sviluppatore possono essere errori nel software, possono esserci errori di input con l’interimento di dati sbagliati, gli errori hardware sono errori per esempio per una non connessione, mentre le autorizzazioi fallite sono errori dovuti alla violazione della sicurezza di una determinata risorsa come un file.*

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

*La gestione di un eccezione viene fatta attraverso un costrutto try – catch.*

*Vengono definite una serie di operazioni che il programma deve svolgere nel blocco try e nel catch (il blocco Gestore) vengono inserite delle parti di codice per gestire una o più eccezioni. Nel momento in cui si genera un’eccezione quindi il programma si blocca e ripercorrendo lo stack va a ricercare il blocco gestore per vedere come gestire quel tipo di errore; nel caso in cui venga trovato quel tipo di errore viene gestito, altrimenti si chiude l’applicazione.*

*Viene inserit eventualmente un altro blocco, successivo al catch, il blocco finally, in cui si inseriscono eventuali comanti da svolgere anche nel caso in cui sia stata generata un’eccezione, come la chiusura di una connessione ad un database in uso.*

1. *Quali sono i possibili casi in cui è necessario lanciare un eccezione da codice? Fornire un esempio.*

*Il lancio di un’eccezione da codice può essere eseguito attraverso la parola throw e viene fatto quando un metodo non può svolgere una richiesta, non può essere eseguita una chiamata verso un oggetto, o quando un metodo genera un’eccezione. Vengono creati oggetti che descrivono un errore e vengono generati con throw.*

*Esempio:*

*public static void Testo( string testo)*

*{*

*\_ = testo ?? ArgumentException(“il testo non può essere nullo”, nameof(testo));*

*}*

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

*Esempi di eccezioni più comuni possono essere:*

1. *InvalidCastException: eccezione generata quando si vuole eseguire un cast, una conversione, ad un altro tipo con esito negativo.*
2. *IndexOutOfRange: eccezione generata quando l’indice inserito supera l’indice del range del vettore.*
3. *OutOfMemoryException: quando si vuole inserire un oggetto in memoria ma è piena.*
4. *StackOverflowException: quando vengono eseguite diverse chiamate ad un metodo e rimangono in attesa e quindi viene esaurito lo stack di esecuzione.*
5. *DivideByZeroException: quendo si cerca di eseguire una divisione ma il valore per il quale si vuole dividere è zero.*
6. *Descrivere i ruoli che esistono all’interno di un progetto*

*I ruoli che esistono all’interno di un progetto sono sei:*

1. *Product Owner: è il cliente, colui che assegna il progetto da svolgere e che definisce le caratteristiche che vuole nel progetto. Quindi decide e comunica come deve essere il progetto finale ed è colui che lo utilizzerà.*
2. *Business Analysts: è colui che ha competenze relative al tipo di progetto, quindi è una figura intermediaria tra il cliente (Product Owner) e lo sviluppatore. Deve capire i bisogni del business.*
3. *Software architect: colui che progetta il sistema che gli sviluppatori dovranno utilizzare per creare il codice del progetto. Creano uno schema architetturale concettuale che è necessario per creare un progetto specifico.*
4. *Sviluppatori: coloro a cui viene affidato il compito di creare il lavoro richiesto dal cliente e faranno riferimento al progetto architetturale svolto dal software architect.*
5. *Team QA: è un team che testa il software sviluppato asicurandosi che rispetti i requisiti richiesti. Testano sia le caratteristiche funzionali sia la performance e sicurezza.*
6. *Team Operation: è un team che prende il software finito e testato per svolgere le ultime fasi del progetto come la configurazione di requisiti tecnici.*
7. *Dare una definizione di UML e spiegare perché si utilizza.*

*L’UML (Universal Modelling Language) è un tipo di linguaggio standard universalmente accettato che viene utilizzato per la modellazione orientata agli oggetti. Viene utilizzata per avere una linea generale dell’applicazione, una comunicazione efficente ed efficace del progetto in modo da ridurre il gap sematico tra utente e tecnico che deve realizzare il progetto e per avere un’astrazione utile, ossia per avere un’overview del progetto.*

*È un linguaggio basato su diagrammi e permette di specificare, visualizzare, realizzare e documentare un artefatto di un sistema. L’obiettivo è quello di costruire diverse viste del progetto correlandole tra loro. L’insieme di questi modelli è detto Visual Modeling.*

*É possibile effettuare diverse modifiche in modo più efficace permettendo di avere un’idea chiara di cosa andare a modificare con le varie relazioni.*

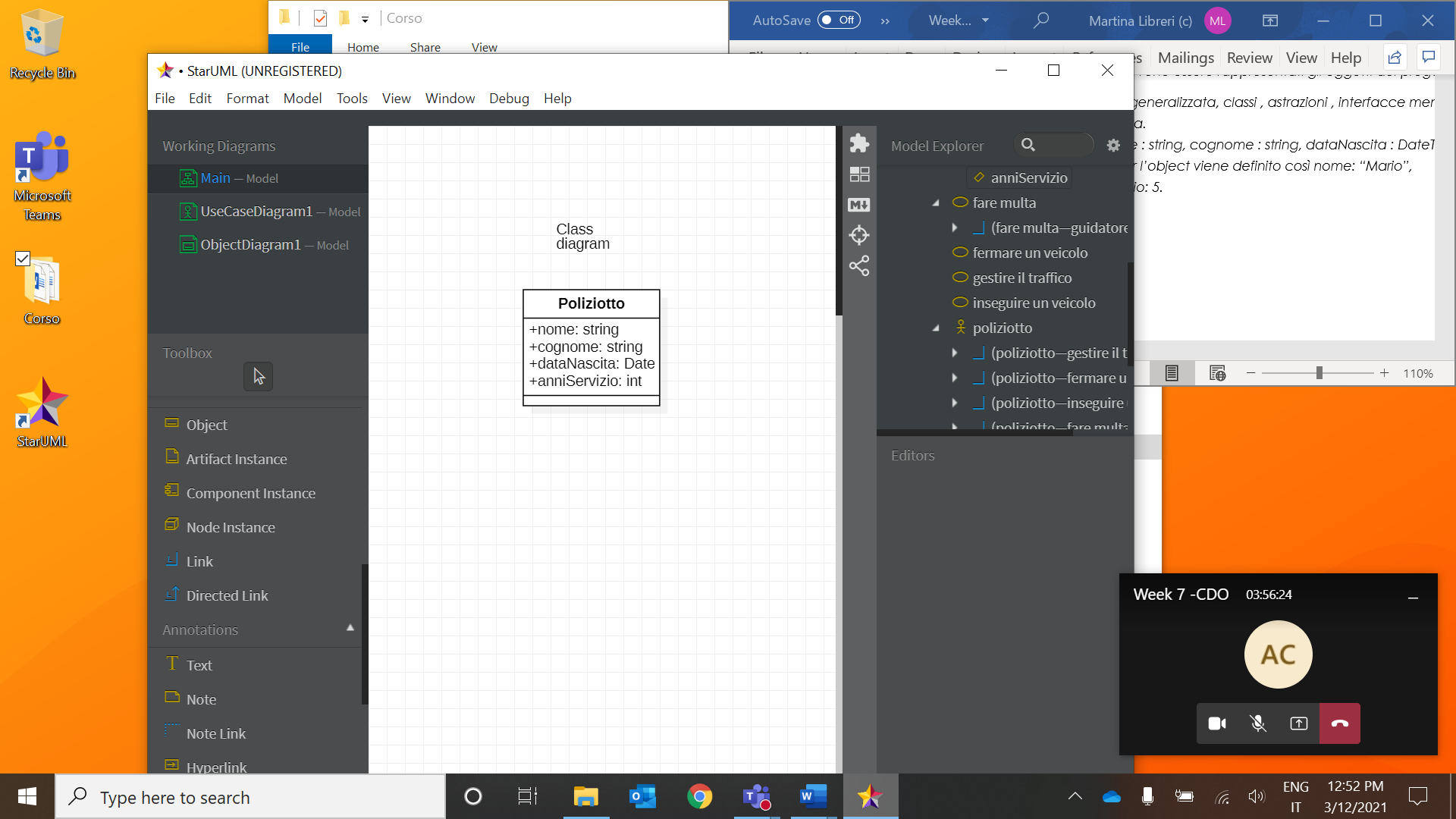
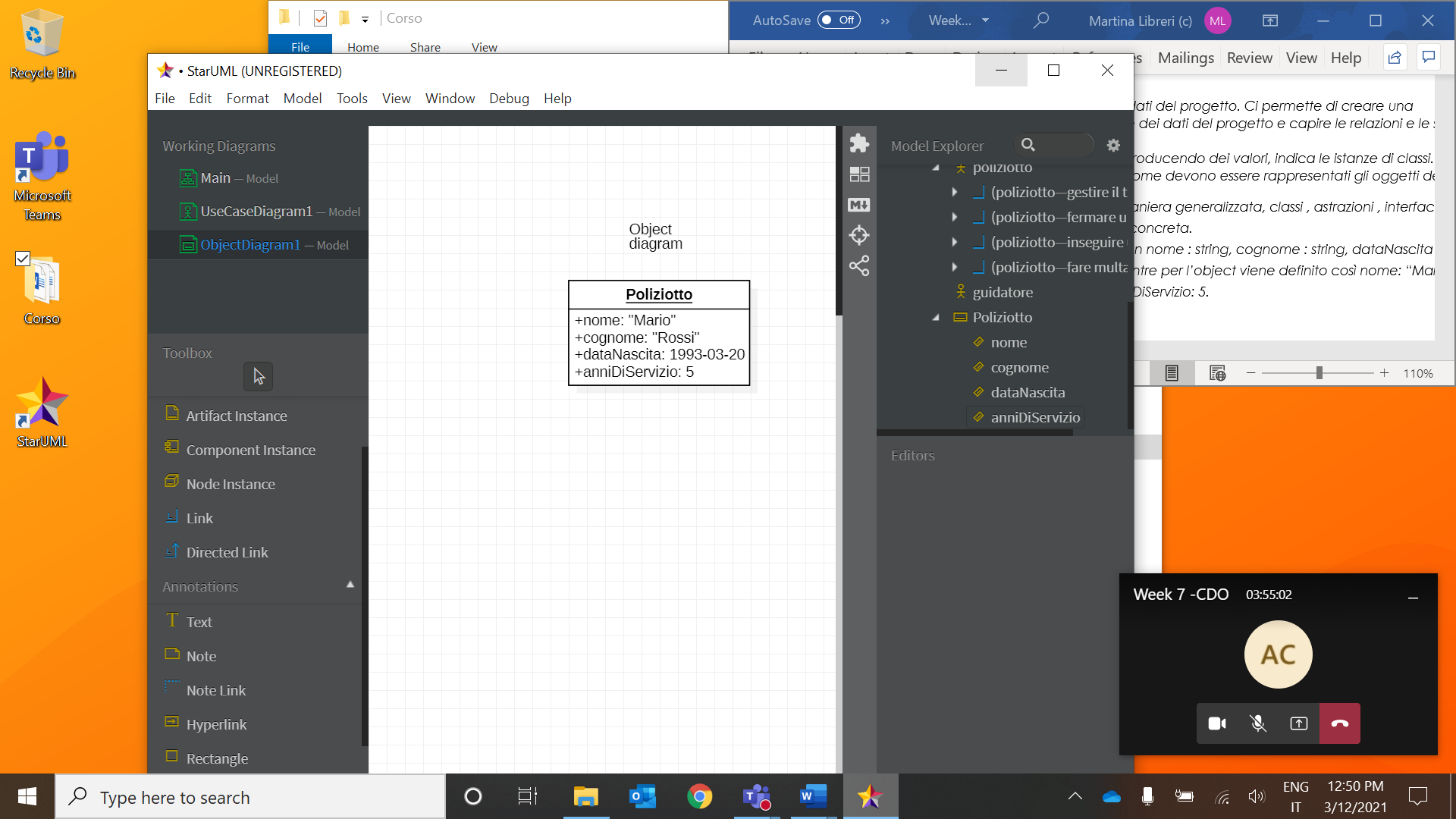
1. *Descrivere la differenza tra class diagram e object diagram descrivendo gli elementi di cui possono essere composti. Fornire un esempio.*

*Sono presenti diversi structural diagrams tra cui:*

1. *Class diagram: utilizzato per il dominio dati del progetto. Ci permette di creare una rappresentazione schematica generale dei dati del progetto e capire le relazioni e le specifiche richieste dal product owner.*
2. *Object diagram: descrive gli oggetti introducendo dei valori, indica le istanze di classi. Viene utilizzato per mostrare degli esempi di come devono essere rappresentati gli oggetti del progetto.*

*Il diagramma di classi connette gli elementi in maniera generalizzata, classi , astrazioni , interfacce mentre l’object diagram mostra un’oggetto in maniera concreta.*

*Per esempio: si definisce un oggetto poliziotto con nome : string, cognome : string, dataNascita : DateTime, anniDiServizio : int con il diagramma di classi, mentre per l’object viene definito così nome: “Mario”, cognome: “Rossi”, dataNascita: 1993-03-20, anniDiServizio: 5.*

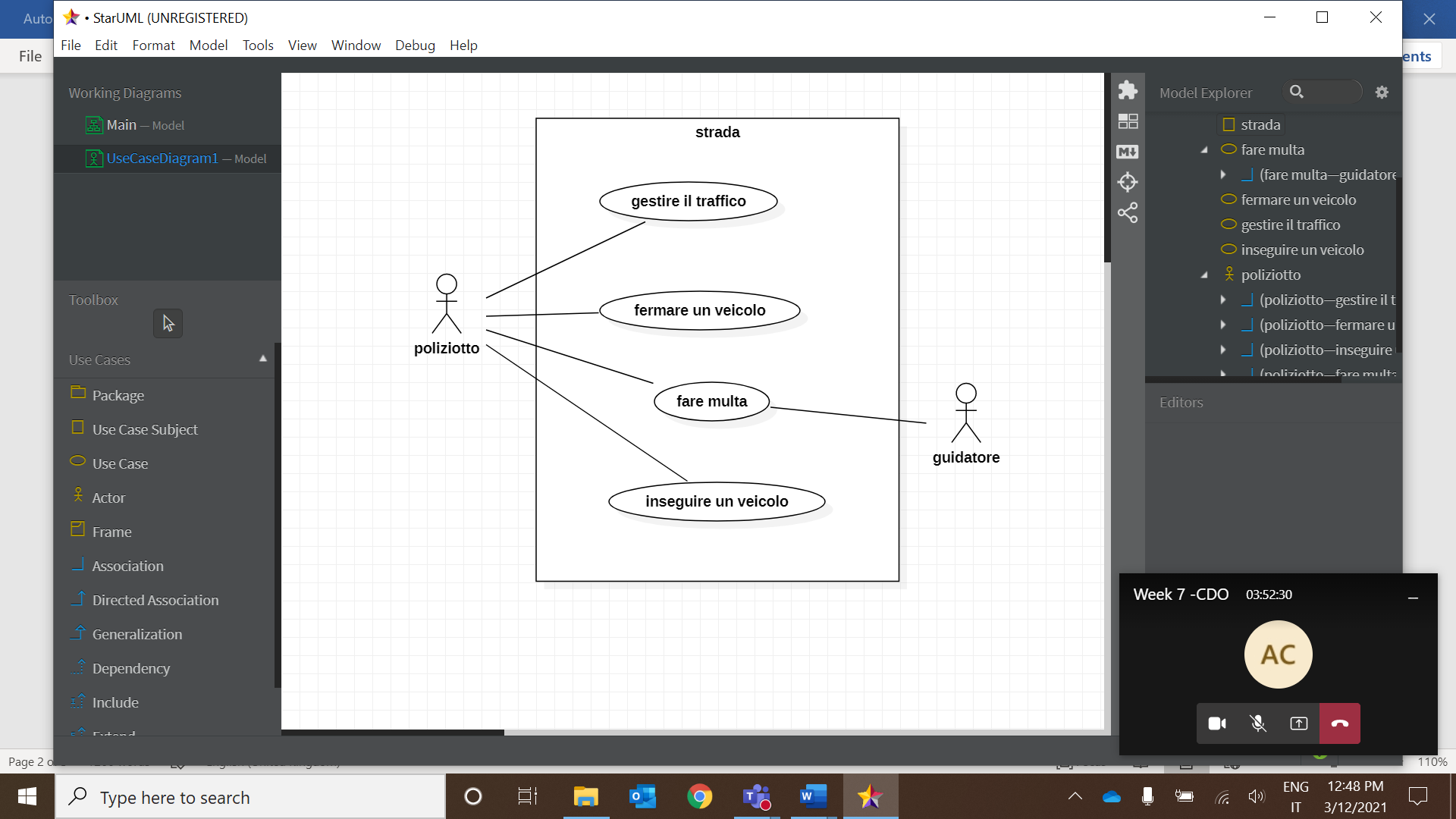


1. *Fornire un esempio di use case diagram. Quali elementi sono in gioco e qual è l’obiettivo del diagramma?*

*(Non usare gli esempi presentati in classe)*

*Nello use case diagram vengono inseriti gli attori, colui che interagisce nel progetto, lo use case ossia le azioni che possono essere svolte.*

*Ad esempio un actor poliziotto che inizia l’azione quindi viene posizionato a sinistra nel diagramma può fesmare un veicolo, gestire il traffico, inseguire un veicolo, fare una multa ad un altro attore posizionato a destra che subisce l’azione.*



*.*

*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.*

