Week6 - Prova Settimanale - Parte Teorica

1. Le eccezioni sono errori dati da una situazione imprevista avvenuta durante l’esecuzione di un programma. Possono essere causate sia da errori dello sviluppatore, sia da altri problemi indipendenti da lui, come un’ autorizzazione fallita o un problema di hardware. Durante l’esecuzione di un codice è possibile che si verifichino diverse tipologie di errore: errori di utilizzo, errori del programma, errori di sistema. Ci sono eccezioni date dal compilatore che possono verificarsi durante le operazioni aritmetiche (come quando si divide per 0 un numero intero), oppure quando una conversione esplicita ha esito negativo, quando si cerca di indicizzare un array ma l’indice è negativo o supera i limiti imposti dalla dimensione, oppure quando il tipo di un elemento che si cerca di archiviare in un array non è compatibile con il tipo dell’array.
2. Quando viene generata un’eccezione, l’esecuzione del programma viene interrotta. Viene associata al contesto di runtime il tipo di eccezione generata e, ripercorrendo lo stack delle chiamate fino al Main, si cerca un gestore dell’errore. Se quest’ultimo esiste, prende in carico l’ eccezione, se invece non esiste un gestore nello stack l’eccezione comporta la chiusura dell’applicazione. Il gestore si mette in pratica tramite il costrutto TRY-CATCH: nel blocco try si mette il codice che potrebbe causare l’eccezione e nel blocco catch si mette il gestore dell’eccezione. Si può aggiungere un blocco finally dove inserire del codice da inserire sia nel caso che vada in errore sia nel caso che ci vada.

Esempio:

try

{

StreamReader sr = new StreamReader(@ “fileProva.txt”);

string contenuto = sr.ReadToEnd();

}

catch (FileNotFoundException ex)

{

Console.WriteLine(“Errore: {0}”, ex.message);

}

finally

{

if(sr != null)

sr.Close();

}

In questo esempio cerchiamo di leggere il contenuto di fileProva.txt, se il file non viene trovato si stampa il messaggio dell’eccezione FileNotFoundException, in ogni caso alla fine se lo StreamReader non è nullo, viene chiuso.

1. Le figure che compongono il team che si occupa di un progetto normalmente sono 6:
2. Product Owner: è il proprietario del prodotto finale.
3. Business Analyst: svolge il ruolo di intermediario tra il product owner e il team tecnico.
4. Software/solution Architect: è colui che progetta i sistemi, identifica il modo in cui gli sviluppatori dovrebbero costruire il sistema e si pone come guida per tutti gli aspetti tecnici relativi al progetto.
5. Sviluppatori: ereditano i disegni funzionali e tecnici redatti dai business analyst e dall’ architect.
6. Team QA (Quality Assurance) o Team di testing: si assicura che il software scritto dagli sviluppatori corrisponda ai requisiti definiti in fase di analisi, e che il sistema risponda sia in termini di logica funzionale, sia in termini di performance e di sicurezza.
7. Team Operations: prende il software finito e testato, predispone gli ambienti di esecuzione temporanea (Staging) e finali (Production) occupandosi della configurazione di quelli che sono i requisiti tecnici di esecuzione.

4. Un class diagram ci permette di produrre una rappresentazione schematica delle entità presenti nel modello dati, comprendente tutte le specifiche funzionali che sono state espresse dal committente. Un object diagram è un grafo che include oggetti e valori di stato. Un diagramma di oggetti statici rappresenta le istanze delle classi descritte nel diagramma di classe.