Corso Intermedio - Lezione 1

http://www.vbsimple.net/intermed/int_01.htm

- Prima di cominciare....
- L'analisi di un progetto.

Il corso intermedio si apre con un po' di sana teoria. Gli argomenti trattati in questa lezione sono validi sempre e indipendentemente dal linguaggio di sviluppo utilizzato.

Imparare a memoria la sintassi di un linguaggio è perfettamente inutile se non si conosce e comprende esattamente il problema da affrontare. Quello che si intende insegnare è la mentalità del programmatore/analista, lo spirito con cui si deve affrontare un problema, **prima ancora di decidere se Visual Basic è il linguaggio più idoneo**.

Un giorno qualcuno disse *"Analisti si nasce, programmatori si diventa"*.
Nei progetti di piccole dimensioni le due figure sono svolte dalla medesima persona ma nessuno dei due ruoli può essere scartato nello sviluppo di un progetto.

Un **analista** si occupa di studiare il problema in questione scomponendolo in piccoli problemi semplici e di trovare le soluzioni migliori per risolverli. Al contrario, un **programmatore** si dovrà occupare soltanto di tradurre, nel linguaggio deciso per la risoluzione del problema, le decisioni prese dall'analista, scegliendo i mezzi più idonei per svolgere questo lavoro. Si tratta comunque di uno stretto rapporto di collaborazione.

Analisti si nasce per il senso di osservazione e per il rapporto verso l'ambiente esterno.
Il senso di osservazione porta alla scoperta di ciò che sta dietro le cose, alla loro scomposizione in elementi semplici, come un arancio in spicchi, uno spicchio in polpa e succo ma senza dimenticare la presenza di eventuali noccioli. Soltanto così potrà essere affrontato qualsiasi problema indipendentemente dalla sua grandezza, dei mezzi a disposizione per la sua risoluzione e dei costi che essa ne comporterà.

Programmatori si diventa in funzione delle necessità.
Lo studio di un linguaggio, infatti, richiede un tempo davvero breve (variabile da persona a persona) se accompagnato da uno studio del problema. Il cambio di linguaggio di sviluppo non dovrebbe spaventare affatto: quello che è necessario è imparare la nuova sintassi, acquisendo i pregi ed i limiti del linguaggio stesso.

Il bravo analista si limiterà alla costruzione di un modello generico, indipendente dal linguaggio che il programmatore vorrà utilizzare senza entrare nel merito delle decisioni che lo sviluppatore vorrà prendere.

Il bravo programmatore, invece, si manterrà scrupolosamente alle decisioni prese dall'analista. In caso di difficoltà nello sviluppo sarà opportuno discutere con l'analista e trovare una soluzione alternativa, evitando assolutamente di prendersi la briga di

modificare le decisioni dell'analista a sua insaputa. Questo infatti potrebbe pregiudicare lo sviluppo futuro del prodotto in esame.

Nei frequentissimi casi in cui la figura dell'analista coincide con quella dello sviluppatore viene adottata una tecnica di sviluppo del tutto simile all'addobbo di un ***albero di Natale***. Viene costruito uno scheletro di base (l'albero) e poi arricchito con i particolari che svolgeranno l'elaborazione vera e propria (le decorazioni dell'albero).

Questo genere di tecnica, però, è soggetta a grossi ed a volte insormontabili rischi. Una decisione presa inizialmente, infatti, può condurre ad un vicolo cieco ovvero all'impossibilità di continuare il proprio lavoro a meno di tornare indietro e modificare le decisioni iniziali e quindi il programma finora scritto.

Una soluzione pratica per evitare problemi di questo genere consiste nella scomposizione del codice in più parti, ognuna delle quali **completamente indipendente** dal resto del programma. Questo è infatti uno dei punti di forza della programmazione ad oggetti che trova il suo cardine principale nell'*Information hiding*, ovvero l'atto del nascondere alle parti del codice che accedono ad una classe quei dati fondamentali per il corretto funzionamento della stessa. Ciò farà si che una modifica strutturale o di svolgimento del ruolo della classe non influenzerà il comportamento del resto del programma, esterno alla classe. La classe infatti si limiterà a svolgere una determinata funzione, in maniera del tutto ignorata dal resto del programma.

Prendiamo in esame di analisi un caso abbastanza semplice ovvero lo sviluppo di un software che archivi i brani dei CD musicali. Il programma fondamentalmente richiede all'utente l'inserimento di certi dati che saranno poi archiviati da qualche parte per poi restituirli al momento della consultazione.

Le possibilità di sviluppo sono infinite. Quale sarà la soluzione migliore?
L'analisi si fonda sul alcune domande semplicissime:

- **Qual'è il problema?**
  - *Possiedo molti CD ed a volte trovare un brano specifico tra i tanti è difficile e comporta una perdita di tempo.*
  - *Possiedo delle compilation di autori vari ed il rintracciamento di un autore rende difficoltosa e lunga la ricerca.*

- **Come intendo risolverlo?**
  - *Sviluppando un software che archivi i titoli e gli autori di ogni brano dei CD.*
  - *Il medesimo software permetterà le ricerche sia per autore che per brano.*
  - *Consentirà anche la consultazione casuale, ovvero la vista dell'elenco dei vari brani; mi accade ogni tanto di voler ascoltare qualche brano ma non so quale scegliere.*

- **Esistono altre soluzioni, anche peggiori?**
  - *Avevo pensato di scrivere i titoli dei brani di ogni CD in un documento di testo ed utilizzare un documento per ogni CD.*
  - *Scannerizzare tutte le copertine dei CD.*

- **Perché ho scelto questa soluzione?**

- *Perché tutte le altre soluzioni (ovviamente) comportavano difficoltà e difetti.*
  - **Le difficoltà sono legate ai miei limiti oppure sono intrinseche alla soluzione stessa?**
    - *Sono legate al tipo di soluzione e quindi non permetterebbero la risoluzione del mio problema.*

- **Che costi (in termine di risorse e lavoro) questa soluzione mi comporta?**
  - *Ho già qualche conoscenza del linguaggio e penso di potermela cavare in un certo tot di tempo e ciò mi sta bene.*
  - *Non ho idea però per organizzare le operazioni di consultazione dei brani.*
    - **Ho provato a leggere la guida in linea fornita con VB?**
      - *Non l'ho installata.*
        - **Lo farò immediatamente.**
      - *No.*
        - **Lo farò immediatamente.**
      - *Si, ma non contiene nulla di idoneo al mio problema.*
        - **Ho dato uno sguardo agli esempi presenti sui CD di Visual Basic?**
          - *Wow! Non sapevo ci fossero esempi di VB nei CD!*
    - **Ho provato a cercare soluzioni sul Web?**
      - *Si, ma non ho trovato nulla.*
      - *Ho trovato alcune cose ma non sono idonee al mio caso.*
    - **Sono alla conoscenza di alcuni siti Web che trattano VB?**
      - *No.*
        - **Allora che sto leggendo giusto ora?** 😉
          - *Proverò a guardare prima in questo sito e poi ai siti riportati nella sezione Links.*
      - *Si ma non ho trovato nulla al loro interno.*
        - **Forse sarà meglio utilizzare un motore di ricerca.**
          - *Proverò con http://www.google.com oppure con http://www.yahoo.it*
    - **Esistono altre risorse oltre il Web?**
      - *Si! Esistono newsgroup, chat e mailing list.*
        - **Prima di chiedere leggerò i messaggi vecchi.**
        - **Esiste un sito Web legato al newsgroup o alla mailing list?**
          - Si.
            Utilizzerò il motore di ricerca del sito del newsgroup o della mailing list.
        - **Esiste una sezione FAQ (Frequently Asked Questions)?**
          - Si e la consulterò!
            Quasi sempre si trova la soluzione.
        - **Chiederò aiuto al newsgroup o alla mailing list!**

- **Esistono componenti esterni riutilizzabili?**
  - *Ho trovato una soluzione utilizzando un OCX sviluppato da terzi.*
    - **È gratuito? Posso utilizzarlo nel mio progetto?**
      - *No! Lo compro perché è molto utile.*
      - *Si.*
    - **Sono disposto ad appesantire il mio progetto includendo un**

componente di terze parti?
- *No.*
  Potrei provare a farne a meno.
- **Che rischi/problemi mi comporta utlizzare tale componente esterno?**
  - *Pagarlo.*
  - *Dovrò sacrificare alcune caratteristiche del mio programma perché incompatibili con l'utilizzo di questo componente esterno.*
  - *Mi tocca distribuirlo assieme al resto del progetto e quindi registrarlo.*
    Il pacchetto di installazione potrebbe crescere di dimensioni e richiedere altri componenti.
- *Ho trovato un controllo utente non compilato o del codice sorgente gratuito.*

- **A quale target è rivolto il mio programma?**
  - *A me e ai miei amici.*
  - *A tutti quanti lo desiderino.*
    - **Sarà necessario scrivere un minimo di documentazione per utilizzarlo.**
  - *A tutto il mondo.*
    - **Sarà necessario scrivere la documentazione del programma.**
    - **Sarà necessario anche un file di risorse o un meccanismo di traduzione del testo all'interno del progetto.**
      - *Sarà meglio utilizzare un file di risorse meglio ancora se una DLL separata.*

- **Il progetto sarà distribuito con codice sorgente allegato?**
  - *Si.*
    - **Sarà necessario indentare bene il codice scritto.**
    - **Dovrò commentarlo per bene e seguire delle regole di base nella scrittura del codice e nella denominazione di variabili e degli oggetti.**

- **Il programma sarà eseguito da un solo utente contemporaneamente?**
  - *No. Sarà eseguito anche in un ambiente multiutenza (LAN o Internet).*
    - **Sarà pertanto obbligatorio utilizzare un sistema di archiviazione multiutente.**
    - **Quanti brani può arrivare a contenere l'archivio?**
      - *Meno di 1.000.*
      - *Tra 1.000 e 30.0000.*
      - *Un numero difficile da stimare.*
        - **Perché?**
          - *Perché la LAN conta molti computer e molti utenti.*
          - *Perché l'archivio sarà reso pubblico e verrà utilizzato tramite Internet da tutti gli utenti del programma.*

L'analisi potrebbe proseguire per molti punti ancora ed è esattamente ciò che avviene in progetti di grosse dimensioni e con tante persone legate al progetto.

Poiché la stragrande maggioranza degli utenti di questo corso sviluppa in privato ci limiteremo ad un progetto di piccole dimensioni con un archivio locale monoutente. Affronteremo questo progettino nelle prossime lezioni.

Fibia FBI
21 Dicembre 2001


Torna all'introduzione

Vai alla seconda lezione