

Pascal: kiss it goodbye?

Agostino Lorenzi
Università di Bergamo, Via Caniana 2, 24127 Bergamo

agostino.lorenzi@unibg.it

Sommario

I docenti di matematica che curano l'approccio all'informatica nelle classi iniziali della scuola superiore si sono disaffezionati al linguaggio Pascal: i nuovi orientamenti vanno verso l'uso del foglio elettronico e dei software per la matematica. È possibile recuperare gli aspetti formativi che erano presenti nell'uso del Pascal con altri ambienti software e attraverso una nuova visione della programmazione che tenga conto delle tecnologie ormai consolidate? La proposta qui presentata si basa sull'uso del foglio elettronico anche come ambiente di programmazione, con uno sguardo al Web.

1. Introduzione

In modo del tutto indolore e silenzioso, l'uso del linguaggio Pascal come approccio alla programmazione e all'informatica è stato abbandonato nelle classi iniziali della scuola superiore (nel biennio, come si dice comunemente) nelle attività di progettazione e di laboratorio che negli ultimi anni erano state affidate ai docenti di matematica.

La cosa può anche essere valutata positivamente, se considerata come scelta di rinnovamento nella didattica e di aggiornamento rispetto alle nuove tecnologie. È noto d'altra parte come in generale il linguaggio Pascal sia diventato obsoleto di fronte all'emergere di nuovi linguaggi, ma soprattutto inadeguato per i nuovi stili di progettazione e di sviluppo del software, oltre che per le applicazioni per il Web che hanno oggi un'importanza centrale.

Si tratta allora di discutere se e come gli aspetti formativi presenti nell'attività didattica, che usava il linguaggio Pascal come strumento, possano essere recuperati utilizzando altri ambienti o altri linguaggi, per sostenere comunque alcuni obiettivi di fondo irrinunciabili per i processi di apprendimento negli studenti dei primi anni della scuola superiore.

2. Cosa c'era di bello nel Pascal

Nella seconda metà degli anni '80 il linguaggio Pascal entra nella scuola come erede del linguaggio Basic, anche in questo caso in modo indolore e silenzioso, nel senso che, pur in assenza di indicazioni precise da parte del Ministero, l'orientamento prevalente negli insegnanti "pionieri" dell'informatica e negli organizzatori di corsi di aggiornamento era per un linguaggio fortemente strutturato che potesse servire a tradurre in modo facile, ma anche rigoroso, i procedimenti risolutivi come algoritmi formalizzati.

Credo che nel corso degli anni moltissimi docenti di informatica e matematica abbiano fatto un'esperienza di insegnamento della programmazione basata sul linguaggio Pascal.

Del linguaggio Pascal abbiamo sicuramente apprezzato la facilità e la semplicità dell'ambiente di sviluppo, e nello stesso tempo la richiesta di rigore nella stesura del codice, che poteva servire a creare negli studenti l'abitudine al rispetto della sintassi e al formalismo

informatico. Il linguaggio risultava particolarmente adatto alla codifica delle strutture di controllo (sequenza, selezione e ripetizione nelle loro diverse forme) e all'organizzazione del programma in sottomoduli indipendenti attraverso le procedure e le funzioni. Non poteva essere altrimenti, dal momento che il linguaggio Pascal rappresentava l'implementazione dei principi della programmazione strutturata e della metodologia di sviluppo top-down.

Altro aspetto formativo di non poco conto era rappresentato dall'obbligo di dichiarare i dati e le strutture dati, specificandone nome e tipo, rendendo così quasi inevitabile almeno un minimo di analisi del problema e di progettazione iniziale da parte degli studenti, prima di mettersi davanti alla tastiera per editare il codice del programma.

A ciò si aggiunga poi il fatto che l'uso di un linguaggio compilato introduceva nell'attività di laboratorio la fase forse più interessante della programmazione, rappresentata dalla ricerca e dalla correzione degli errori di sintassi e ancor più degli errori logici.

3. Nuovi stili di programmazione

La domanda che ci si pone è la seguente: è possibile recuperare gli aspetti formativi, che erano presenti nell'uso del Pascal, con altri ambienti software? La risposta è sicuramente positiva.

Per quanto riguarda il tipo di attività informatica di cui stiamo parlando, il passaggio dal linguaggio Pascal ad altre tecnologie per l'approccio all'informatica, può essere paragonato al momento che ha visto il passaggio graduale dall'uso del C:>, cioè la riga comandi del DOS, all'uso dei sistemi operativi con interfaccia grafica.

Nel linguaggio Pascal, e in generale in tutti i linguaggi di programmazione evoluti, la programmazione si basava sull'equazione fondamentale (enunciata dal padre del Pascal, N. Wirth): programma = dati + istruzioni, o meglio,

$$\text{programma} = \text{algoritmo} + \text{strutture dati}$$

Nell'informatica moderna questo non basta più. L'equazione deve forse essere modificata in (Figura 1):

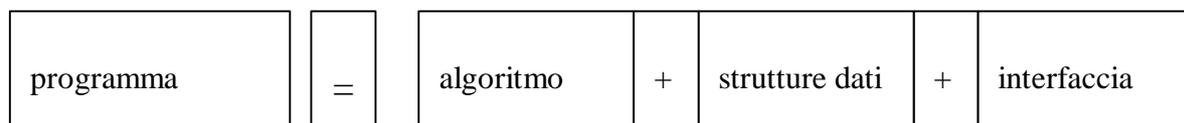


Figura 1 – Elementi della programmazione

L'elemento nuovo è la presenza dell'interfaccia come componente fondamentale del software. Questo è dovuto a molti motivi, ma soprattutto a tre fondamentali: (1) l'uso del computer non è più riservato a specialisti, ma è diffuso a livello di massa; (2) l'interazione con l'utente è sempre più amichevole e basata sull'uso di oggetti grafici; (3) le tecnologie software hanno avuto una forte accelerazione con lo sviluppo delle applicazioni per il Web.

Si aggiunga inoltre che l'uso esteso delle interfacce grafiche ha introdotto nei programmi, oltre alle variabili e alle istruzioni, altri elementi: gli *oggetti* che hanno attributi e metodi e gli *eventi*, generati dall'azione dell'utente o del computer, che il programmatore deve gestire.

Tutto questo aggiunge nuovi aspetti formativi nell'attività didattica: gli studenti non devono costruire programmi che sono soltanto la traduzione di un procedimento risolutivo, ma devono fare attenzione a tutto quello che può succedere durante la fase di esecuzione del programma, prevedendone gli eventi, oltre che considerare con attenzione il problema della

comunicazione e delle modalità di interazione con l'utente del programma nella costruzione delle interfacce.

4. Un percorso nuovo basato sul foglio elettronico

La scelta di introdurre il foglio elettronico nelle classi iniziali della scuola superiore, come ambiente software di base per le attività di laboratorio informatico, è sicuramente corretta: è un ambiente di facile approccio, che consente anche applicazioni matematiche in modo esteso. Allora l'idea può essere quella di mantenere lo stesso ambiente per introdurre i principi della programmazione, usando il linguaggio VBA presente all'interno del programma Excel.

In sostanza si possono ipotizzare tre fasi di un nuovo percorso di formazione informatica:

1) una prima fase nella quale il docente insegna l'uso del foglio elettronico, con i concetti di base di ricalcolo, riferimenti di cella, formule e funzioni di base, rappresentazioni grafiche;

2) una seconda fase nella quale si usano le funzioni del foglio elettronico per fare applicazioni matematiche, evidenziando i due livelli fondamentali del foglio elettronico, il foglio dati e il foglio formule; si possono anche aggiungere funzioni statistiche e funzioni di ricerca (*Conta.Se*, *Cerca.Vert*, ecc.)

3) una terza fase nella quale si usa l'ambiente di programmazione VBA di Excel per costruire piccole procedure, evidenziando il terzo livello del foglio elettronico, rappresentato dal codice.

La figura 2 rappresenta i tre livelli ai quali l'utente del foglio elettronico si può collocare:

- il livello *dati* è proprio dell'utilizzatore finale che usa il foglio per i calcoli e per organizzare i dati in modo tabellare;
- il livello *formule* riguarda l'attività di progetto e di formalizzazione di procedimenti di calcolo normalmente svolti in modo non automatizzato;
- il livello *codice* corrisponde alla personalizzazione del foglio elettronico con l'aggiunta di nuove procedure.

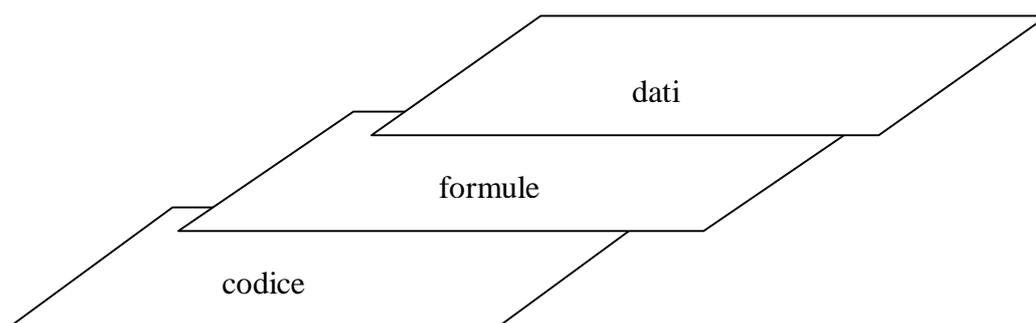


Figura 2 – I tre livelli del foglio elettronico

5. Dalle formule al codice

La necessità della programmazione dovrebbe nascere in modo naturale, come risposta alla richiesta di trattare problemi complessi o comunque non risolvibili con le sole formule e funzioni del foglio elettronico.

Per esempio si può iniziare con un primo semplice problema che può essere risolto solo con la programmazione: come si può fare per raddoppiare il contenuto della cella B3?

La soluzione, presentata dalla figura seguente, mostra la riga di codice che consente di aggiornare il contenuto della cella con il doppio del suo valore.

	A	B	C	D
1				
2				
3		200		
4				Raddoppia
5				
6				
7				
8				

```
Private Sub Raddoppia_Click()
Range("B3") = Range("B3") * 2
End Sub
```

Figura 3 – Un esempio semplice di programmazione

Inoltre l’approccio al codice può scaturire dalla presentazione del concetto di macro e dall’analisi del codice creato automaticamente dal programma Excel durante la creazione delle macro. Lo studente prima registra una nuova macro, facendo compiere al programma alcune azioni, quali inserimenti di valori, copia o spostamento, oppure modifica dei colori dei caratteri e degli sfondi di cella. Successivamente visualizza il codice della macro, evidenziando le modalità con cui vengono rappresentate le celle del foglio e la sintassi dei comandi di più immediata comprensione. In questa fase non è importante riconoscere il significato completo delle istruzioni della macro, quanto comprendere la sequenza delle azioni e come ciascuna di esse corrisponda a una o più righe di codice.

Ma l’aspetto formativo più interessante è rappresentato dalla costruzione di piccole procedure nel linguaggio VBA che riproducono lo stesso effetto delle funzioni predefinite di Excel, come nell’esempio della figura 4 che illustra una procedura per simulare la funzione predefinita *Conta.Se* di Excel.

	A	B	C	D	E
1	12				
2	34		Numeri > 20		
3	19				
4	2		3		
5	6				
6	45				
7	21			Conta	
8					
9					
10					

```
Private Sub Conta_Click()
Dim Quanti As Integer
Quanti = 0
For Each numero In Range("Valori")
If numero.Value > 20 Then
Quanti = Quanti + 1
End If
Next
Conteggio.Text = Quanti
End Sub
```

Figura 4 – Simulazione della funzione Conta.Se con codice VBA

Questo serve a far capire allo studente quello che “sta sotto” il foglio dei dati e sotto il foglio delle formule, fornendo una visione più precisa del funzionamento del sistema di elaborazione come insieme di risorse hardware e software.

La programmazione può servire anche ad estendere le funzionalità del foglio elettronico con la costruzione di procedure che possono rendere più efficiente l’elaborazione, con riferimento a particolari esigenze di calcolo o di personalizzazione del foglio di lavoro. Per

esempio la figura 5 illustra una semplice procedura che consente di calcolare la somma dei numeri contenuti in una colonna visualizzando il valore calcolato nella prima cella libera della colonna stessa. L'esercizio serve anche a dimostrare l'uso della struttura di ripetizione con controllo in coda *Do...Loop Until*.

	A	B	C	D
1	20			
2	30			
3	40			
4	50			
5	140	totale		
6				
7				
8				

CalcolaSomma

```

Private Sub CalcolaSomma_Click()
  s = 0
  i = 0
  Do
    i = i + 1
    valore = Cells(i, 1)
    s = s + valore
  Loop Until valore = ""
  Cells(i, 1) = s
  Cells(i, 2) = "totale"
End Sub

```

Figura 5 – Aggiunta di una nuova funzionalità al foglio elettronico

Resta comunque intatta l'importanza della programmazione e del computer come strumenti per la validazione degli algoritmi, come già accadeva per il linguaggio Pascal.

Uno strumento molto efficace è costituito, anche in questo ambiente di programmazione, dal debugger, con la possibilità di eseguire passo passo il programma e di visualizzare il valore delle variabili durante l'esecuzione.

Conviene anche aggiungere che l'editing del codice in VBA presenta alcune caratteristiche interessanti per il lavoro del programmatore: la formattazione automatica delle righe di codice, l'uso dei colori per evidenziare le parole chiave del linguaggio e i commenti, la segnalazione degli errori di lessico e sintassi del linguaggio di programmazione durante la stesura del codice. Sono tutti elementi di supporto al lavoro del programmatore.

Di seguito è presentata una breve descrizione schematica dei contenuti di un possibile modulo didattico dedicato alla programmazione:

Formalizzazione dei problemi, algoritmi, principi di programmazione

Formalizzare i problemi

L'algoritmo

Dati e istruzioni

Rappresentazione degli algoritmi

Tracing

Le strutture di controllo

Il passo successivo: l'ambiente di programmazione

Il codice sotto le macro

La codifica delle strutture di controllo fondamentali: sequenza, selezione, ripetizione

Dati e strutture dati

Le strutture derivate di selezione e ripetizione

Gli oggetti dell'interfaccia grafica

Algoritmi standard: contare, sommare, calcolo del massimo, ricerca sequenziale.

6. Il foglio elettronico e il Web

L'introduzione di nuovi percorsi formativi per l'informatica può portare beneficio anche come conseguimento, da parte degli studenti, di competenze che sono propedeutiche alla capacità di realizzare semplici applicazioni per il Web. Le tecnologie Web moderne, infatti, utilizzano in modo esteso il concetto di oggetto grafico di interfaccia, oltre che le proprietà degli oggetti, i metodi applicati agli oggetti e la gestione degli eventi (si pensi per esempio al linguaggio JavaScript), in modo del tutto simile al linguaggio VBA.

Inoltre è tipico delle tecnologie per il Web la realizzazione di applicazioni, non come lunghi programmi, ma come insieme di moduli software di piccole dimensioni e riutilizzabili, oltre che associabili all'interfaccia offerta all'utente.

In sostanza lo stile di programmazione si deve adeguare alle modalità tipiche, considerate ormai standard, di scrittura del software con i linguaggi legati alle moderne tecnologie Web.

Non trascurerei inoltre la possibilità offerta dal programma Excel di esportare un foglio di lavoro in formato Web, mantenendo anche l'interattività nel momento in cui il foglio viene aperto dal browser. Il foglio di Excel può essere infatti salvato come pagina Web, mantenendo gli elementi fondamentali dell'interazione con l'utente. Quando la pagina Web contenente il foglio viene aperta nel browser, è possibile operare su di essa inserendo valori e formule di calcolo nelle celle. Eventualmente il foglio può essere salvato sul computer locale dell'utente in formato Excel. A questo punto, dopo aver apportato le modifiche, l'utente può a sua volta pubblicare la nuova versione del foglio in formato Web con mantenimento dell'interattività.

Molto interessante è la possibilità di esportare in formato Web anche un grafico, mantenendo l'interattività: in questo caso nella pagina Web vengono inclusi automaticamente anche i dati di origine del grafico. Operando dal browser, l'utente può modificare i valori di origine e ottenere sul video la conseguente modifica del grafico.

Si crea di fatto uno strumento di lavoro collaborativo tra docente e studente e tra studente e studente, attraverso l'utilizzo di fogli di calcolo dinamici e non statici nel Web.

7. Alcune esperienze già condotte

Il percorso proposto è stato sperimentato da me personalmente in tre contesti diversi, con risultati ugualmente positivi per l'interesse che l'impostazione didattica ha suscitato e per la maggiore facilità riscontrata negli studenti per l'apprendimento dei principi della programmazione:

1) breve corso introduttivo alla programmazione del computer per gli studenti di una classe seconda dell'ITC "B. Belotti" di Bergamo.

L'esperienza riguarda una serie di incontri di orientamento per gli studenti che avevano scelto di frequentare l'indirizzo programmatori (Mercurio) nel triennio. Il background degli studenti era costituito dalle conoscenze del sistema operativo Windows e dall'uso operativo del foglio elettronico Excel nelle ore di laboratorio di matematica. L'aspetto di maggiore interesse per gli studenti è stata la progettazione dell'interfaccia grafica, con la scoperta della possibilità di ricreare con la propria applicazione gli oggetti grafici (finestre, bottoni, caselle di testo, ecc.) già visti e usati ampiamente in Windows. Dal punto di vista formativo, per il docente è risultato più semplice far cogliere l'importanza della scelta dei dati e dei tipi di dato da associare agli oggetti grafici e la necessità di prevedere il controllo degli eventi causati dall'utente durante l'esecuzione del programma. Infatti in questo tipo di attività lo studente riesce più facilmente a distinguere tra i dati e le funzioni all'interno di un'applicazione software.

2) corso di Informatica al primo anno della Facoltà di Economia dell'Università di Bergamo [Gnudi e Lorenzi, 2002].

Questo contesto appare un po' anomalo rispetto a quanto detto sopra; in realtà, a parte la differenza di età, il livello delle competenze informatiche degli studenti del primo anno dei corsi universitari non è necessariamente molto elevato, né ovviamente omogeneo, per la diversità degli indirizzi di studi di provenienza. Molti studenti arrivano alle Facoltà universitarie da Istituti tecnici e professionali con un buon bagaglio di conoscenze e abilità di tipo informatico, altri, provenienti dai Licei, hanno competenze in questo settore quasi inesistenti o comunque meno consolidate. Il corso di Informatica tratta quindi gli aspetti dell'informatica di base e contiene anche un modulo sugli elementi minimi ed elementari della programmazione.

Le motivazioni di questo piccolo modulo stanno nell'importanza che gli studenti acquisiscano un'abitudine alla formalizzazione dei problemi e dei procedimenti risolutivi. Anziché introdurre un linguaggio di programmazione apposito, si è deciso più opportunamente di usare l'ambiente Excel, in modo da operare in un ambiente già conosciuto e con un'uguale interfaccia.

Riguardo all'esperienza svolta, si può rilevare un aumento consistente del numero di studenti che hanno sostenuto e superato l'esame al termine del corso e i buoni risultati anche nelle domande del test di esame riguardanti la programmazione, rispetto ai pessimi risultati che mediamente venivano riscontrati quando la programmazione era presentata in ambiente Pascal.

3) corso di Elementi di Informatica nella Scuola di Specializzazione per l'Insegnamento Secondario (SSIS) dell'Università di Bergamo [Cavalli e Lorenzi, 2002].

In questo contesto il percorso formativo è stato proposto ai docenti, specializzandi di matematica, come nuova tecnologia didattica a supporto dell'insegnamento della matematica nella scuola superiore

Gli argomenti fondamentali del corso di "Elementi di Informatica" sono stati: uso del foglio elettronico, algoritmi e fondamenti della programmazione in ambiente VBA, uso dei prodotti software per la matematica (Derive, Cabri).

Con questa proposta è stata offerta ai corsisti la possibilità di sperimentare personalmente una modalità di didattica attiva che può arricchire la loro professionalità docente e che essi stessi potranno riproporre in futuro per i loro studenti.

Al secondo anno della scuola i corsisti hanno potuto provare da subito i materiali prodotti nel corso con i propri allievi, in quanto già docenti nella scuola superiore oppure come tirocinanti durante i periodi di stage previsti nel programma della scuola di specializzazione.

8. Riferimenti bibliografici

Cavalli E., Lorenzi A. (2002), *Teacher Training Through E-Learning Technology Online Educa, 8th International Conference on Technology Supported Learning & Training*, Berlin

Gnudi A, Lorenzi A. (2002), *Competenze informatiche per gli studenti al primo anno degli studi universitari: standard di base e skill aggiuntivi Atti del XL Congresso annuale AICA*, Conversano, Bari, pp.81-87.