

## Gestione Preventivi: Il Listino Prezzi

Vorrei ora presentarvi la costruzione di un'applicazione di **Gestione preventivi** molto semplice e intuitiva nell'uso quotidiano, che dispone di un archivio per poter archiviare e visionare i preventivi emessi, di un'anagrafica clienti, di un listino prezzi e altri piccoli accessori che possono essere utili senza dover ricorrere a software costosi e altrettanti onerosi canoni di manutenzione. Ritengo la fase di progettazione di un software molto importante in quanto deve essere prevista la possibilità di eseguire future implementazioni e si deve operare in modo da poter essere in grado di approntarle in un prossimo futuro. Pertanto è opportuno dotare il foglio preventivi delle opportune "ruote di scorta" che possono esserci utili nel proseguo della programmazione. Parte fondamentale di questa breve presentazione è il foglio **Listino** e il foglio **Archivio**, in quanto si deve prevedere la possibilità di poter aggiungere dei campi a seconda delle necessità.

Ho pensato di utilizzare un listino prezzi separando gli articoli in categorie merceologiche in modo da poter individuare facilmente gli articoli durante la stesura del preventivo stesso, e facilitando anche le operazioni di stampa per categoria per i normali usi di gestione giacenze. Pertanto la struttura del foglio *Listino*, avrà questo aspetto:

|   | A                  | B                           | C           | D            | E             | F           | G           | H             | I           | J           | K          | L          | M          | N          | O |
|---|--------------------|-----------------------------|-------------|--------------|---------------|-------------|-------------|---------------|-------------|-------------|------------|------------|------------|------------|---|
| 1 | <b>Categoria 1</b> |                             |             |              |               |             |             |               |             |             |            |            |            |            |   |
| 2 | <b>Codice</b>      | <b>Descrizione Articoli</b> | <b>Note</b> | <b>Forn.</b> | <b>CodFor</b> | <b>U.M.</b> | <b>Data</b> | <b>Prezzo</b> | <b>Util</b> | <b>Tot.</b> | <b>RDS</b> | <b>RDS</b> | <b>RDS</b> | <b>RDS</b> |   |
| 3 | cod1.1             | Articolo 1 categoria 1      |             |              |               |             |             | 3,10          | 30%         | 5,28        |            |            |            |            |   |
| 4 | cod2.1             | Articolo 2 categoria 1      |             |              |               |             |             | 6,10          | 30%         | 9,92        |            |            |            |            |   |
| 5 | cod3.1             | Articolo 3 categoria 1      |             |              |               |             |             | 4,10          | 30%         | 6,24        |            |            |            |            |   |
| 6 | cod4.1             | Articolo 4 categoria 1      |             |              |               |             |             | 7,10          | 30%         | 4,64        |            |            |            |            |   |
| 7 | cod5.1             | Articolo 5 categoria 1      |             |              |               |             |             | 8,10          | 30%         | 5,28        |            |            |            |            |   |
| 8 |                    |                             |             |              |               |             |             |               |             |             |            |            |            |            |   |

**Fig. 1**

**Nota:** La larghezza delle colonne è stata volutamente ristretta per poterla rappresentare in questa guida in modo graficamente visibile, potete dare le dimensioni che volete alle varie colonne.

Come vedete ho strutturato il Listino Prezzi sviluppando le categorie sulle colonne e dedicando ad ogni categoria 14 colonne, separandole da una colonna vuota. La *riga 1* ospita il titolo della categoria (rif. A1), la *riga 2* le etichette (rif. A2) per le varie colonne e dalla *riga 3* inseriamo tutti gli articoli con i relativi dati. Essendo ogni categoria composta da 14 colonne, divise da una colonna vuota, e considerando che un foglio di Excel ha 256 colonne disponibili (ver. 2003), facendo un rapido calcolo, dividendo le 256 colonne totali per le 15 utilizzate (14 + la colonna vuota) danno come risultato 16 categorie, che a mio avviso è un numero più che soddisfacente di categorie.

Ogni categoria ha una colonna in cui inserire il codice articolo, poi la descrizione dell'articolo e infine una colonna note, a cui si può dedicare la locazione dell'articolo. Per esempio se parliamo di un componente del motore di una vettura sarà alloggiato nella categoria Motore, e nel campo note si può identificare ulteriormente inserendo Distribuzione oppure Iniezione. Può ritornare utile questo "sottogruppo" per poter estrarre tutti gli articoli che appartengono alla categoria Motore suddividendoli per sottocategorie, come per esempio Iniezione o Distribuzione. Dopo il campo note troviamo il campo Fornitore e di seguito il Codice Fornitore, che sarebbe il codice dell'articolo assegnato dal fornitore nel suo archivio, molto utile da riportare in eventuali ordini a fornitore.

Nel campo U.M. va inserita l'unità di misura dell'articolo e nel campo successivo la data cui viene inserito il prezzo, che si trova nel campo successivo e che si riferisce al prezzo di acquisto dell'articolo. Il campo successivo rappresenta la percentuale di ricarico sul prezzo di acquisto dell'articolo e il prodotto viene inserito nel campo successivo che rappresenta il prezzo finale di vendita. In seguito ci sono altre 4 colonne vuote, che simpaticamente le chiamiamo RDS cioè Ruota di Scorta, utili per ricevere eventuali implementazioni in un prossimo futuro

Fatta questa premessa per spiegare la struttura del foglio Listino possiamo ora passare all'interrogazione dello stesso ipotizzando che in questa fase gli articoli di prova siano stati inseriti manualmente agendo direttamente dal foglio, vedremo anche l'inserimento e la modifica degli articoli nella Form che segue. Come abbiamo già spiegato nel corso base inseriamo una Userform e la strutturiamo come la seguente inserendo i vari controlli.

Fig. 2

Come prima operazione da eseguire, dobbiamo popolare il ListBox1 (quello delle categorie), e per farlo sfruttiamo l'evento **Attivate** della form che ospita i controlli inserendo questo codice.

Codice:

```
Private Sub UserForm_Activate()
1) Dim j As Integer
2) j = 1
3) Do Until Sheets("Listino").Cells(1, j) = Empty
4) ListBox1.AddItem (Sheets("Listino").Cells(1, j).Value)
5) j = j + 15
6) Loop
End Sub
```

**Nota:** Il codice sorgente esposto **NON** include un numero davanti ad ogni linea, La numerazione delle linee dei listati rappresentati è stata usata solo per facilitare l'identificazione e la discussione di particolari linee nei vari listati

Commentiamo questo codice: Alla *riga 1* dell'enunciato viene dichiarata la variabile *j* come Integer e alla *riga 2* viene posta uguale a 1. Alla *riga 3* con un ciclo Do Loop si scorre tutto il foglio *Listino* con le coordinate di *riga 1* e di *colonna j* che in questo momento ha il valore di 1, pertanto le coordinate diventano *riga 1, colonna 1*. Alla *riga 4* carichiamo il ListBox1 con i valori reperiti nel foglio Listino alle coordinate di *riga 1* e *colonna j* ma alla *riga 5* la variabile *j*, che vale 1 viene incrementata di 15 unità e sostituendo nell'operazione matematica il nome della variabile e inserendo il suo valore ci riporta  $j=1+15$  che prende il valore di 16. Alla *riga 6* ritorniamo ad inizio del ciclo tramite la parola chiave Loop con la variabile *j* che adesso vale 16. Si continua nel ciclo Do effettuando un "salto" ogni 16 colonne fino a trovare nelle coordinate di riga e colonna nessun valore e interrompere il ciclo.

Una volta che abbiamo riempito il ListBox1 delle categorie, facendo clic sullo stesso dobbiamo riempire il ListBox2 con l'elenco degli articoli presenti per la categoria selezionata. Per eseguire l'operazione richiesta sfruttiamo l'evento **Click** del ListBox2 e inseriamo questo codice

Codice:

```
Private Sub ListBox1_Click()
1) posiz = (ListBox1.ListIndex + 1) * 15 - 14
2) If ListBox2.ListCount >= 1 Then ListBox2.Clear
3) i = 3
4) Do Until Sheets("Listino").Cells(i, posiz) = Empty
5) With Sheets("Listino")
6) elemen = .Cells(i, posiz).Value & Space(5) & .Cells(i, posiz + 1).Value
```

```

7)      End With
8)      ListBox2.AddItem elemen
9)      i = i + 1
10)     Loop
11)     cancella
End Sub

```

Nella prima istruzione usiamo una variabile (*posiz*) che abbiamo precedentemente dichiarato pubblica ad inizio del modulo, per cui condivisibile in tutte le routine dello stesso.

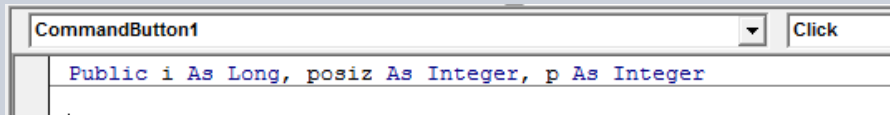


Fig. 3

Nella *riga 1* usiamo la variabile *posiz* per calcolare le coordinate della colonna iniziale delle categorie usando una espressione matematica articolata in questo modo. L'istruzione `ListBox.ListIndex + 1` riporta la posizione della voce che andiamo a selezionare nella casella a discesa (ListBox1), per esempio se clicchiamo sulla Categoria 2, la funzione riporta come risultato il numero 2. Per verificarlo modifichiamo il listato in questo modo e mandiamolo in esecuzione

Codice:

```

Private Sub ListBox1_Click()
posiz = (ListBox1.ListIndex + 1)
MsgBox posiz
End Sub

```

E ci riporta un messaggio come questo

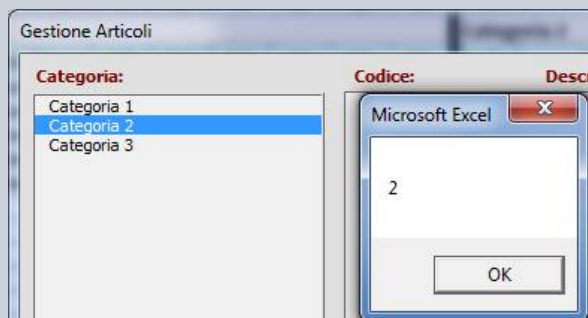


Fig. 4

Pertanto selezionata la categoria nel ListBox1 abbiamo il valore di `ListBox.ListIndex` che andremo a moltiplicare per **15** (che rappresenta il numero delle colonne presenti in una categoria) e a cui sottraiamo il numero **14** (che rappresenta il numero delle colonne meno la colonna vuota nera di divisione). Facendo un rapido calcolo le 3 categorie si trovano alla *colonna 1* (Rif. A1) poi alla *colonna 16* (Rif. P1) e l'ultima categoria si trova alla *colonna 31* (Rif. AE1)

Infatti cliccando sulla prima voce del ListBox1 ci viene rimandato come valore di `ListBox.ListIndex` il numero 1 che moltiplicandolo per 15 e sottraendo 14 riporta come risultato finale **1**, che rappresenta la colonna 1. Utilizzando la stessa logica matematica per le altre 2 categorie, avremmo che per la seconda categoria `ListBox.ListIndex` assume il valore di 2 e per la terza categoria il valore di 3. Eseguendo l'operazione matematica per la seconda categoria moltiplichiamo il valore di `ListBox.ListIndex` che è 2 per 15 e sottraendo 14 otteniamo come risultato finale il numero 16. Che corrisponde all'esatto numero di colonna in cui inizia la seconda categoria. Anche per la terza categoria, seguendo sempre il calcolo matematico, moltiplicando il valore di `ListBox.ListIndex` che è 3 per 15 e sottraendo 14 riporta come risultato finale il numero 31.

Ritornando all'enunciato del ListBox1 alla *riga 2* si verifica se il numero di voci presenti nel ListBox2 sia maggiore di zero o uguale a 1, se il risultato della verifica è True (cioè Vero) allora si svuota tutto l'elenco di voci presenti nel ListBox2. Proseguendo alla *riga 3* poniamo la *variabile i* uguale a 3 e nella *riga 4*, scorriamo tutte le colonne del foglio Listino alle coordinate di *riga i* (che vale 3) e di *colonna posiz* (che vale 1 per la prima voce, 16 per la seconda e 31 per la terza) e il ciclo si ferma quando alle coordinate di colonna NON trova niente scritto niente (Empty = vuoto) nella cella. Successivamente alla *riga 5* iniziamo un ciclo With che alla *riga 6* usiamo per caricare la matrice *elemen* con i valori presenti alle coordinate di *riga i* e *colonna posiz*, inseriamo poi uno spazio di 5 caratteri e memorizziamo in matrice anche i valori che si trovano alle coordinate di *riga i* e *colonna posiz+1*. Alla *riga 7* dichiariamo la fine del ciclo With e alla `[color="Navy"] riga 8[/color]` popoliamo il listBox2 con la matrice *elemen*. Alla *riga 9* incrementiamo la variabile *i* di una unità e alla *riga 10* ritorniamo all'inizio del ciclo Do con la parola chiave *Loop* e ripetiamo tutte le istruzioni con il valore di riferimento di riga incrementato di 1 per passare così alle righe successive. Il ciclo si arresta quando si trova una riga vuota. Alla *riga 11* viene richiamata la routine Cancella che si occupa di svuotare tutti i valori

presenti nei vari TextBox della form tramite questo codice

Codice:

```
Private Sub cancella()
z = 1
For z = 1 To 10
    Controls("TextBox" & z) = ""
    Next z
End Sub
```

Con questo codice scorriamo tutti i controlli TextBox assegnando tramite il ciclo For l'indice z che corrisponde a TextBox(z). al primo ciclo z vale 1 per cui l'enunciato diventa TextBox1 e via di seguito fino alla fine del ciclo. A questo punto abbiamo il ListBox 2 popolato che ci presenta tutti gli articoli presenti nella categoria selezionata, ora cliccando su un articolo presente nel listBox2 dobbiamo estrarre i valori dell'articolo presenti nel foglio e inserirli nelle varie caselle di testo (TextBox) utilizzando il seguente codice.

Codice:

```
Private Sub ListBox2_Click()
1)      p = ListBox2.ListIndex + 3
2)      cancella
3)      vedi
End Sub
```

L'istruzione alla *riga 1* calcola il numero di riga corrispondente alla voce selezionata, come abbiamo visto per il ListBox1, l'istruzione *ListBox.ListIndex* riporta il numero di voce che viene selezionata, per cui se selezioniamo la 3° voce ci riporta il numero 3 a cui sommiamo ancora il valore 3 per ottenere come risultato finale e nostro riferimento di riga il valore 6. Se andiamo a controllare in **figura 1** noteremo che la 3° voce dell'elenco degli articoli è posizionato sulla riga 6. Ritornando al codice alla *riga 2* viene richiamata la routine Cancellata che abbiamo appena visto per il ListBox1, mentre alla riga 3 dell'enunciato viene richiamata la routine *vedi*, che si occupa di recuperare i dati forniti dalle coordinate di riga e colonna fornite dal programma. Il codice usato nella routine *vedi* è il seguente

Codice:

```
Private Sub vedi()
On Error Resume Next
With Sheets("Listino")
    TextBox1.Text = .Cells(p, posiz)           'codice articolo
    TextBox2.Text = .Cells(p, posiz + 4)       'codice articolo fornitore
    TextBox3.Text = .Cells(p, posiz + 1)       'descrizione articolo
    TextBox4.Text = .Cells(p, posiz + 2)       'note articolo
    TextBox5.Text = .Cells(p, posiz + 3)       'fornitore
    TextBox6.Text = .Cells(p, posiz + 5)       'unità di misura
    TextBox7.Text = .Cells(p, posiz + 6)       'data inserimento costo
    TextBox8.Text = Format(.Cells(p, posiz + 7), "#0.00") 'costo articolo
    TextBox9.Text = Format(.Cells(p, posiz + 8), "%") '% ricarico
    TextBox10.Text = .Cells(p, posiz + 9)      'prezzo vendita
    calcola
End With
End Sub
```

Come si vede nel codice tramite un ciclo **With** andiamo ad estrarre dal foglio Listino i valori alle coordinate di *riga p*, che abbiamo determinato con il valore 3 e alla *colonna posiz* che abbiamo già stabilito il suo valore in base alla categoria scelta e inseriamo nei vari TextBox. Da notare che determinato il valore di *posiz*, che corrisponde alla prima colonna della categoria scelta, per identificare le altre colonne usiamo la notazione *posiz+1*, *posiz+2* etc.

Essendo *posiz* un valore variabile tramite la scelta della categoria, in questo modo possiamo spostarci nelle varie colonne con una notazione "variabile" nel codice, cioè non usando delle coordinate fisse che andrebbero bene per una categoria ma non per le altre che andremo a scegliere. Nel proseguo del listato incontriamo poi la routine *calcola*, che si occupa di verificare il prodotto tra il campo "costo articolo" e "percentuale di ricarico" la routine è:

Codice:

```
Private Sub calcola() 'calcolo prezzo articolo con ricarico
TextBox10.Text = Format(CDbl(TextBox10) + CDbl(TextBox8) * Val(TextBox9) / 100, "#0.00")
End Sub
```

In questo enunciato dobbiamo moltiplicare 2 valori, il prezzo di acquisto dell'articolo per la percentuale di ricarico, li andiamo ad inserire nel TextBox del prezzo finale di vendita usando la formattazione numerica con 2 decimali. Per usare un valore decimale si usa la funzione **Cdbl** che converte un valore integer in valore *Double*. Per il TextBox9 (percentuale di ricarico) usiamo invece la funzione **Val** che converte un valore stringa in valore numerico. Ho pensato inoltre che in fase di consultazione del listino si potrebbe decidere di cambiare la percentuale di ricarico, oppure il fornitore ci comunica che il prezzo di acquisto di un determinato articolo è stato variato, invece di dover editare possiamo semplicemente cambiare i valori presenti nei vari TextBox inserendo quelli nuovi e tramite l'evento **Change** possiamo ricalcolare gli importi tramite questo codice

Codice:

```
Private Sub TextBox8_Change() ' costo di acquisto
If TextBox8 <> "" Then
TextBox10.Text = Format(CDbl(TextBox8) + CDbl(TextBox8) * Val(TextBox9) / 100, "#0.00")
Else
End If
End Sub

Private Sub TextBox9_Change() '% ricarico
If TextBox9 <> "" Then
TextBox10.Text = Format(CDbl(TextBox8) + CDbl(TextBox8) * Val(TextBox9) / 100, "#0.00")
Else
End If
End Sub
```

Con le due routine sopra esposte possiamo modificare i valori del TextBox8 (prezzo di acquisto) e del TextBox9 (percentuale di ricarico) per aggiornare automaticamente il prezzo finale di vendita. Da notare che abbiamo inserito l'espressione matematica in un ciclo IF in quanto se dovessimo cancellare tutte le cifre in uno dei 2 TextBox ci verrebbe rimandato un errore, in questo modo evitiamo l'errore in quanto la procedura prosegue finchè il TextBox è diverso da stringa vuota. Possiamo inoltre modificare qualsiasi altro TextBox e correggere eventuali errori oppure effettuare degli aggiornamenti a determinati articoli, però per poter salvare nel foglio Listino i nuovi valori si deve cliccare sul pulsante "*Salva*" che riporta questo codice

Codice:

```
Private Sub CommandButton1_Click()
If p = 0 Then
MsgBox "Devi scegliere un articolo", vbExclamation
Exit Sub
End If
scrivi
End Sub
```

Inseriamo anche una condizione con il ciclo IF per essere certi che sia selezionato un articolo quando si clicca sul pulsante di salvataggio dei dati, ricordo che la variabile **p** viene tratta *dall'evento click del ListBox2* dove calcoliamo il numero di riga dell'articolo selezionato, per cui se p è uguale a zero non abbiamo selezionato nessun articolo e rimandiamo un avviso all'utente che deve selezionare un articolo prima di procedere con il salvataggio e forziamo l'uscita dalla routine. Successivamente viene richiamata la routine scrivi che presenta questo codice

Codice:

```
Private Sub scrivi()
On Error Resume Next
With Sheets("Listino")
.Cells(p, posiz) = TextBox1.Text 'codice articolo
.Cells(p, posiz + 1) = TextBox3.Text 'descrizione articolo
.Cells(p, posiz + 2) = TextBox4.Text 'note articolo
.Cells(p, posiz + 3) = TextBox5.Text 'fornitore
.Cells(p, posiz + 4) = TextBox2.Text 'codice articolo fornitore
.Cells(p, posiz + 5) = TextBox6.Text 'unità di misura
.Cells(p, posiz + 6) = TextBox7.Text 'data inserimento costo
.Cells(p, posiz + 7) = TextBox8.Text 'costo articolo
.Cells(p, posiz + 8) = TextBox9.Text '% ricarico
.Cells(p, posiz + 9) = TextBox10.Text 'Prezzo vendita
End With
End Sub
```

Come si può notare è molto simile alla routine *vedi*, sono solo state invertite le posizioni dei TextBox rispetto ai riferimenti del foglio nel ciclo With. In questo caso il valore ubicato alle coordinate di *riga p e colonna posiz* vengono sovrascritte dai valori

presenti nei vari TextBox. Proseguiamo assegnando al pulsante **Uscita** la routine

Codice:

```
Private Sub CommandButton2_Click()
    Unload Me
End Sub
```

Passiamo ora alla routine **Stampa Categoria**, ricordando che *si deve prima selezionare una categoria nel ListBox1 e solo successivamente premere il pulsante Stampa categoria* a cui viene assegnato il seguente codice

Codice:

```
Private Sub CommandButton3_Click()
    stampa
End Sub
```

La routine sopra descritta viene assegnata al tasto **Salva** presente nella Userform che richiama la routine *stampa* che si occupa della stampa vera e propria degli elementi selezionati con questo codice

Codice:

```
Sub stampa()
1) Application.ScreenUpdating = False
2) Sheets("listino").Select
3) Cells(1, posiz).Select
4) ultima = Cells(Rows.Count, posiz).End(xlUp).Row
5) Range(Cells(1, posiz), Cells(ultima, posiz + 9)).Select
6) Selection.Copy
7) Workbooks.Add
8) Range("A1").Select
9) ActiveSheet.Paste
10) Range("a1").Select
11) Range("C:C,I:I").Delete
12) Columns(1).ColumnWidth = 8 'codice
13) Columns(2).ColumnWidth = 30 'descrizione
14) Columns(3).ColumnWidth = 15 'fornitore
15) Columns(4).ColumnWidth = 8 'codice fornitore
16) Columns(5).ColumnWidth = 8 'U.M.
17) Columns(6).ColumnWidth = 10 'data
18) Columns(7).ColumnWidth = 10 'vendita
19) With ActiveSheet.PageSetup
20) .PrintTitleRows = "$1:$2"
21) .CenterHeader = "Listino Articoli - Pagina &P di &N"
22) .RightHeader = "&""Arial,Grassetto"&12Stampa del &D"
23) .LeftMargin = Application.InchesToPoints(0)
24) .RightMargin = Application.InchesToPoints(0)
25) .CenterHorizontally = True
26) .CenterFooter = "Data stampa :&D"
27) .Zoom = 90
28) End With
29) ActiveSheet.PrintOut Copies:=1
30) ActiveWorkbook.Saved = True
31) ActiveWorkbook.Close
32) Windows("Preventivi.xls").Activate
33) Range("Q10").Select
34) Application.CutCopyMode = False
35) Application.ScreenUpdating = True
36) Unload Me
End Sub
```

Questa routine di stampa è stata strutturata per avere un riepilogo scritto degli articoli presenti in una categoria relativo ai prezzi di acquisto, sono stati volutamente esclusi i campi in cui abbiamo inserito la percentuale di ricarico e il prezzo di vendita. Può essere utile in forma scritta se si vuole controllare i prezzi di acquisto dei fornitori per singolo articolo, oppure affidare il controllo a qualche subalterno senza dover divulgare notizie sensibili sulle percentuali di utile.

Commentiamo ora il codice della routine. Con l'istruzione presente alla *riga 1* si evita di vedere a video le operazioni che compie

la macro in esecuzione, alla *riga 2* attiviamo il foglio “Listino” e selezioniamo la cella alle coordinate di riga 1 e colonna posiz (che conosciamo già) alla *riga 4* troviamo l’ultima cella scritta della colonna posiz e alla *riga 5* selezioniamo l’intervallo di celle partendo dalla riga 1 e colonna posiz fino alla riga ultima e colonna posiz + 9.

Alla *riga 6* copiamo l’intervallo di celle, alla *riga 7* aggiungiamo una nuova cartella di lavoro, selezioniamo la cella A1 alla *riga 8* e alla *riga 9* incolliamo i dati copiati precedentemente, poi selezioniamo la cella A1 alla *riga 10* e cancelliamo due colonne alla *riga 11*. Dalla *riga 12* alla *riga 18* modifichiamo la larghezza delle prime 7 colonne per adattare al testo e dalla *riga 19* alla *riga 28* impostiamo il setup della stampante, per poi mandare il documento in stampa alla *riga 29*. Alla *riga 30* salviamo il file creato, (magari utile per future consultazioni) e alla *riga 31* lo chiudiamo. Passiamo poi il controllo al file origine, cioè “Preventivi.xls” alla *riga 32*, selezioniamo la cella Q10 alla *riga 33* e alla *riga 34* disabilitiamo la selezione delle celle copiate nel foglio Listino, per poi riabilitare lo schermo e chiudere la routine alla *riga 36*

---