

Lezione 9

26–27 ottobre 1999

Argomenti trattati

– Esempi ed esercizi.

9.1 Esempio: calcolatrice tascabile

Vogliamo costruire un programma che simuli una calcolatrice tascabile estremamente rudimentale. La calcolatrice legge due interi non negativi e il simbolo dell'operazione desiderata, calcola e scrive il risultato ottenuto. Le operazioni sono la somma, la sottrazione, il prodotto, la divisione e il resto della divisione, e vengono selezionate utilizzando, rispettivamente, i simboli `+`, `-`, `*`, `/`, `%`.

Ad alto livello, il programma sarà costituito dalle seguenti fasi:

1. lettura degli operandi;
2. lettura dell'operatore;
3. calcolo del risultato;
4. scrittura del risultato.

Le variabili principali di questo programma saranno i due operandi, che chiameremo `x` e `y`, di tipo `integer`, l'operatore, che chiameremo `op`, di tipo `char`, e il risultato, che chiameremo `risultato` di tipo `integer`. Osserviamo che durante l'esecuzione si potrebbero verificare delle condizioni d'errore (ad esempio a causa di una divisione per zero), pertanto introduciamo una variabile booleana di nome `errore` il cui scopo sia quello di segnalare il verificarsi di eventuali condizioni di errore.

Le due fasi di lettura sono strutturate in maniera simile a quella del programma precedente, richiedendo l'inserimento dei dati quando questi non siano nel formato desiderato:

```
{lettura operandi}  
write('Inserisci due numeri interi non negativi ');  
readln(x, y);  
{controllo dati}  
WHILE (x < 0) OR (y < 0) DO  
  BEGIN
```

©1999 Giovanni Pighizzini

Il contenuto di queste pagine è protetto dalle leggi sul copyright e dalle disposizioni dei trattati internazionali. Il titolo ed i copyright relativi alle pagine sono di proprietà dell'autore. Le pagine possono essere riprodotte ed utilizzate liberamente dagli studenti, dagli istituti di ricerca, scolastici ed universitari afferenti ai Ministeri della Pubblica Istruzione e dell'Università e della Ricerca Scientifica e Tecnologica per scopi istituzionali, non a fine di lucro. Ogni altro utilizzo o riproduzione (ivi incluse, ma non limitatamente a, le riproduzioni a mezzo stampa, su supporti magnetici o su reti di calcolatori) in toto o in parte è vietata, se non esplicitamente autorizzata per iscritto, a priori, da parte dell'autore.

L'informazione contenuta in queste pagine è ritenuta essere accurata alla data della pubblicazione. Essa è fornita per scopi meramente didattici e non per essere utilizzata in progetti di impianti, prodotti, ecc.

L'informazione contenuta in queste pagine è soggetta a cambiamenti senza preavviso. L'autore non si assume alcuna responsabilità per il contenuto di queste pagine (ivi incluse, ma non limitatamente a, la correttezza, completezza, applicabilità ed aggiornamento dell'informazione). In ogni caso non può essere dichiarata conformità all'informazione contenuta in queste pagine. In ogni caso questa nota di copyright non deve mai essere rimossa e deve essere riportata anche in utilizzi parziali.

```

        writeln('Errore: i dati inseriti devono essere non negativi');
        write('Ripeti l''inserimento ');
        readln(x, y)
    END; {while x<...}

{lettura operatore}
write('Inserisci il simbolo d''operazione ( + , - , * , / , % ) ');
readln(op);
WHILE (op <> '+') AND (op <> '-') AND (op <> '*')
        AND (op <> '/') AND (op <> '%') DO
    BEGIN
        writeln('Errore: gli operatori previsti sono + , - , * , / , %');
        write('Ripeti l''inserimento ');
        readln(op)
    END; {while op <>...}

```

La fase di scrittura, tenendo conto di quanto detto, sarà:

```

{scrittura del risultato}
IF errore THEN
    writeln('L''operazione ', x : 1, op, y : 1, ' e'' impossibile')
ELSE
    writeln('Il risultato di ', x : 1, op, y : 1, ' e'' ', risultato : 1)

```

Il calcolo del risultato avviene in maniera differente in base all'operatore selezionato. Le istruzioni da eseguire devono essere quindi selezionate in base all'operatore scelto. Questa è una tipica situazione in cui risulta utile effettuare una selezione mediante un'istruzione **CASE**. Nel caso specifico l'istruzione avrà la seguente struttura:

```

CASE op OF
    '+':
        ...
    '-':
        ...
    '*':
        ...
    '/':
        ...
    '%':
        ...
END; {case}

```

Nel caso della somma, della differenza e del prodotto, sarà sufficiente assegnare alla variabile **risultato**, il risultato della corrispondente operazione sui valori contenuti in **x** e **y**. Non si potrà invece procedere al calcolo del quoziente o del resto, se la variabile **y** contiene 0. In tal caso occorrerà segnalare la condizione d'errore, assegnando **true** alla variabile **errore** (che va inizializzata a **false**).

Ecco il listato completo del programma:

```

PROGRAM calcolatrice (input, output);

{simula il comportamento di una calcolatrice tascabile, con operazioni di
somma, sottrazione, moltiplicazione, divisione, resto}

VAR

```

```

x, y: integer;          {operandi}
op: char;               {operatore}
risultato: integer;    {risultato}
errore: boolean;

BEGIN {calcolatrice}

  {lettura operandi}
  write('Inserisci due numeri interi non negativi ');
  readln(x, y);
  {controllo dati}
  WHILE (x < 0) OR (y < 0) DO
    BEGIN
      writeln('Errore: i dati inseriti devono essere non negativi');
      write('Ripeti l'inserimento ');
      readln(x, y)
    END; {while x<...}

  {lettura operatore}
  write('Inserisci il simbolo d'operazione ( + , - , * , / , % ) ');
  readln(op);
  WHILE (op <> '+') AND (op <> '-') AND (op <> '*')
    AND (op <> '/') AND (op <> '%') DO
    BEGIN
      writeln('Errore: gli operatori previsti sono + , - , * , / , %');
      write('Ripeti l'inserimento ');
      readln(op)
    END; {while op <>...}

  {calcolo del risultato}
  errore := false;
  CASE op OF
    '+':
      risultato := x + y;
    '-':
      risultato := x - y;
    '*':
      risultato := x * y;
    '/':
      IF y = 0 THEN
        errore := true
      ELSE
        risultato := x DIV y;
    '%':
      IF y = 0 THEN
        errore := true
      ELSE
        risultato := x MOD y
  END; {case}

  {scrittura del risultato}
  IF errore THEN

```

```

        writeln('L'operazione ', x : 1, op, y : 1, ' e' impossibile')
    ELSE
        writeln('Il risultato di ', x : 1, op, y : 1, ' e' ', risultato : 1)
END. {calcolatrice}

```

9.2 Esempio: di nuovo la calcolatrice tascabile

Vogliamo ora ripetere l'esempio precedente, supponendo che il nostro computer sia in grado di eseguire solo somme e sottrazioni. Si tratta dunque di riscrivere il codice relativo alle operazioni di prodotto, divisione e resto.

Per quanto riguarda il prodotto, è sufficiente osservare che il valore di $x * y$ può essere ottenuto sommando y volte x , a partire da zero. Pertanto il prodotto può essere calcolato come segue:

```

risultato := 0;
FOR i := 1 TO y DO
    risultato := risultato + x

```

Per la divisione utilizziamo un procedimento inverso, determiniamo cioè il numero massimo di sottrazioni del valore di y dal valore di x , con le quali si riesce ad ottenere un numero non negativo. Più precisamente, sottraiamo ripetutamente il valore di y dal valore di x sino ad ottenere un numero minore del valore di y . Il quoziente è dato dal numero di sottrazioni effettuate. Per memorizzare via via il risultato di ogni sottrazione, si utilizza una variabile `diff` di tipo `integer`:

```

risultato := 0;
diff := x;
WHILE diff >= y DO
    BEGIN
        diff := diff - y;
        risultato := risultato + 1
    END {while diff...}

```

Per il calcolo del resto della divisione si può utilizzare una strategia del tutto simile: si sottrae ripetutamente il valore di y dal valore di x sino ad ottenere un valore inferiore a quello di y :

```

risultato := x;
WHILE risultato >= y DO
    risultato := risultato - y

```

Ecco il listato completo del programma:

```

PROGRAM calcolatrice2 (input, output);

{simula il comportamento di una calcolatrice tascabile, con operazioni di
 somma, sottrazione, moltiplicazione, divisione, resto}
{In questa versione, i calcoli vengono effettuati utilizzando esclusivamente
 operazioni di somma e sottrazione}

VAR
    x, y: integer;           {operandi}
    op: char;                {operatore}
    risultato: integer;      {risultato}
    errore: boolean;

```

```
    i, diff: integer;

BEGIN {calcolatrice2}

    {lettura operandi}
    write('Inserisci due numeri interi non negativi ');
    readln(x, y);
    {controllo dati}
    WHILE (x < 0) OR (y < 0) DO
        BEGIN
            writeln('Errore: i dati inseriti devono essere non negativi');
            write('Ripeti l''inserimento ');
            readln(x, y)
        END; {while x<...}

    {lettura operatore}
    write('Inserisci il simbolo d''operazione ( + , - , * , / , % ) ');
    readln(op);
    WHILE (op <> '+') AND (op <> '-') AND (op <> '*')
        AND (op <> '/') AND (op <> '%') DO
        BEGIN
            writeln('Errore: gli operatori previsti sono + , - , * , / , %');
            write('Ripeti l''inserimento ');
            readln(op)
        END; {while op <>...}

    {calcolo del risultato}
    errore := false;
    CASE op OF
        '+':
            risultato := x + y;
        '-':
            risultato := x - y;
        '*':
            BEGIN
                risultato := 0;
                FOR i := 1 TO y DO
                    risultato := risultato + x
                END;
        '/':
            IF y = 0 THEN
                errore := true
            ELSE
                BEGIN
                    risultato := 0;
                    diff := x;
                    WHILE diff >= y DO
                        BEGIN
                            diff := diff - y;
                            risultato := risultato + 1
                        END {while diff...}
                END; {else}
    END;
```

```
'%':
  IF y = 0 THEN
    errore := true
  ELSE
    BEGIN
      risultato := x;
      WHILE risultato >= y DO
        risultato := risultato - y
      END {else}
    END; {case}

  {scrittura del risultato}
  IF errore THEN
    writeln('L''operazione ', x : 1, op, y : 1, ' e'' impossibile')
  ELSE
    writeln('Il risultato di ', x : 1, op, y : 1, ' e'' ', risultato : 1)
  END. {calcolatrice2}
```

Esercizi

1. Modificare il programma che simula la calcolatrice in modo che, dopo avere ottenuto il risultato di un calcolo, l'utente possa inserire iterativamente nuovi dati e operatori, per effettuare altri calcoli, senza essere costretto a rilanciare il programma.
2. Estendere il programma che simula la calcolatrice, nelle due versioni viste, ampliando l'insieme di operazioni possibili. Ad esempio, si potrebbero aggiungere le operazioni di elevamento a potenza, il calcolo del massimo comun divisore e del minimo comun multiplo.