

1.1 PICO

Linguaggio didattico avanzato 1995-2004



Pico è un linguaggio sviluppato alla Vrije Universiteit di Brussel (1995), per permettere a studenti non informatici di programmare con un linguaggio semplice e potente.

Pico si ispira al dialetto LISP Scheme ed al libro di Abelson e Sussman "Structure and Interpretation of Computer Programs".

Di Pico esistono due versioni, la 1.x e la 2.0 relativamente differenti.

1.1.1 Pico 1.0

Pico è dinamicamente tipizzato, i tipi di dato sono numeri, interi e reali, e stringhe. Inoltre la costante stringa `eoln` quale ritorno a capo e `void`.

Un programma Pico si basa sul paradigma delle funzioni: il valore assegnato ad una variabile è il risultato di una funzione; ciò significa che oltre alle funzioni native (matematiche, di trattamento delle stringhe di caratteri, introspezzive, ecc...) sono funzioni anche gli operatori (aritmetici, relazionali e logici), `if`, `while`, `until` e `begin`, quest'ultima è una funzione con un numero variabile di parametri, di fatto è un blocco di istruzioni:

```
begin(a:10, fac(n):if(=(n,0),1,*(n,fac(-(n,1))))), fac(a))
```

Pico tuttavia accetta una notazione sintattica alternativa, a blocchi ed operatori infissi, in particolare `begin(...)` è sostituibile da `{...}` con le istruzioni separate da `;` (tranne l'ultima), per cui il frammento di programma precedente può essere scritto come:

```
{a:10; fac(n):if(n=0,1,n*fac(n-1)); fac(a)}
```

L'operatore di assegnazione è `:` per creare una variabile o una funzione e `:=` per modificare una variabile o una funzione.

Sono accettati nomi di funzioni composte da caratteri speciali, ad esempio:

```
begin(++(i):i+1, display(++(++37))) ` produce 39
```

Si noti che la funzione è stata utilizzata prima come funzione, poi come operatore.

Le funzioni di controllo del flusso sono limitate ad `if` e `while`, `if` deve prevedere entrambe le alternative.

L'input output è, dato lo scopo del linguaggio, essenziale: `accept` per acquisire stringhe di caratteri da tastiera e `display` per inviare a console una lista di dati.

L'esempio che segue verifica che dalla quantità di numeri primi fra di loro in un certo intervallo si può ricavare il valore di Π^1 .

```
`WinPico Professional 1.03
`Pico Engine Version 1.1
{a#b: {
  while(not(a = b),if(a>b,a:=a-b,b:=b-a))
};
xGCD(n) : {
  ck: 0;          ` 1 is coprime with all numbers
  i: 1;
  while (i<n, {
    i:=i+1;
    j:i;
    while(j<n, {
      j:=j+1;
      if((i#j) = 1, ck:=ck+1,ck)}
    });
  ck := 2*ck+2*n-1;
```

¹ La probabilità che due numeri a caso siano primi fra di loro è $6/\Pi^2$ (Eulero)

```

display("Coprime numbers from 1 to ",n,' : ',ck,eoln);
display("pi Greek from GCD : ",sqrt((6*n^2)/ck),eoln)
};
PI: *(4,arctan(1));
display("pi Greek : ",PI,eoln);
xGCD(100);
xGCD(500)
}

```

I numeri coprimi fra 1 e 100 danno un valore di $\pi = 3,1396$, fra 1 e 500 $\pi = 3,13902$.

1.1.2 Pico 2

Pico 2 estende Pico, da semplice linguaggio didattico a strumento per esplorare paradigmi di programmazione quali programmazione funzionale ed a oggetti.

Innanzitutto gli operatori di assegnazione sono :

- `:` assegna un valore o una funzione ad variabile,
- `:=` modifica una variabile esistente,
- `::` assegna un valore o una funzione ad un identificatore, in modo immutabile².

Sono introdotti i simboli (*quoted symbols* come `'labelx`) e strutture dati complesse denominate *tables*. Queste possono essere dichiarate e contemporaneamente inizializzate; ogni elemento può, a sua volta contenere una *table*, l'accesso al singolo elemento avviene con la notazione `table[indice1][indice2]....` (oppure `table[indice1,indice2, ...]`).

1.1.3 Varianti

Pic% è una estensione ad oggetti basata su prototipi.

dPico: una variante ad oggetti distribuiti e concorrenti

Sic%: è una implementazione di Pic% con Smalltalk.

TextPico for Linux

sPico for DrScheme

E' prevista una implementazione in Java.

² Cio non vale per la versione per Windows, versione peraltro piuttosto instabile.