

Sommario: 9/04/2019

- Trasmissione per Reference: Errori
- Trasmissione per Reference: Trasm. del Valore Calcolato
- Trasmissione per Reference vs. Uso di Non Locali

# Trasmissione per Reference: Errori

## Esercizio

*Si consideri il seguente programma di un Linguaggio C-like a blocchi, scope statico, procedure ricorsive con trasmissione per valore, reference, e semantica standard per i vari costrutti utilizzati.*

```
{
  void fact(val n,ref r){
    if (n > 0) {
      r = (r * n);
      fact((n - 1),r);
    }
    else r = 1;
  }
  var out;
  fact(5,out);
}
```

*Si dica cosa calcola e perchè, mostrando anche la sequenza degli stati generati dalla computazione.*

## Soluzione

## Esercizio

*Si corregga il programma dell'esercizio precedente in modo tale che ogni invocazione di fact, nel blocco inLine principale, calcoli il fattoriale del primo argomento assegnandolo al valore della variabile trasmessa come secondo argomento.*

```
{
  void fact(val n,ref r){
    if (n > 0) {
      r = (r * n);
      fact((n - 1),r);
    }
    else r = 1;
  }
  var out;
  fact(5,out);
}
```

*Si mostri anche la sequenza degli stati generati dalla computazione allorchè il valore del primo argomento sia 2.*

## Soluzione

## Esercizio

*Si commenti il programma sotto al variare dell'inizializzazione della variabile out. In particolare si dica cosa calcola l'invocazione allorchè out sia inizializzata ad un intero arbitrario k (ignora l'annotazione traceAll)*

```
{
@traceAll
void fact(val n,ref r){
    if (n > 0) {
        r = (r * n);
        fact((n - 1),r);
    }
}
var out = 1;
fact(5,out);
}
```

## Soluzione

# T. per Reference: Trasmissione del Valore Calcolato

## Esercizio

*Il programma sotto richiede che la variabile out, usata per il risultato, sia inizializzata. Come potremmo ridefinire fact in modo che non richieda l'inizializzazione di out e associ sempre al secondo parametro il fattoriale del primo argomento?.*

```
{
  @traceAll
  void fact(val n,ref r){
    if (n > 0) {
      r = (r * n);
      fact((n - 1),r);
    }
  }
  var out = 1;
  fact(5,out);
}
```

*Emulare la trasmissione per Result.*

## Soluzione

# T. per Reference e Non Locali

## Esercizio

Il programma sotto utilizza `out`, come non locale per il risultato dell'invocazione di `fact`.

```
{
  @traceAll
  void fact(val n){
    if (n > 0) {
      out = (out * n);
      fact(n - 1);
    }
  }
  var out = 1;
  fact(5);
}
```

Quali sono i limiti di questa soluzione e perchè deve essere scoraggiata quando il Linguaggio ha trasmissione per reference?

## Soluzione

Almeno 2 ragioni per scoraggiarne l'uso:

- I Non Locali contestualizzano le definizioni. Le astrazioni in particolare, non risultano più indipendenti dal contesto.
- In molti casi l'uso cambia il comportamento: Sostituiamo il parametro `r` con il non locale, nella definizione sotto

```
{
  @traceAll
  void fact(val n,ref r){
    if (n > 0) {
      fact((n - 1),r);
      r = (r * n);
    }
    else r = 1;
  }
  var out;
  fact(3,out);
}
```