COESIONE ITALIA 21-27 PIEMONTE

Cofinanziato dall'Unione europea

REGIONE PIEMONTE

WEB DEVELOPER
Fondamenti di Programmazione

Massimo PAPA

ESEMPI PASSAGGI DI PARAMETRI CON STRUCT

Materiale WEB estratto da:

http://www.iet.unipi.it/g.nardini/ifts/Slides_03.pdf

G.Nardini - Università di Pisa

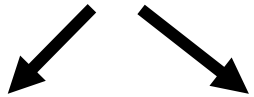
Strutture come parametri di funzioni

```
int main() {

  struct PolReg {
    int numLati;
    float lunghezzaLato;
  };
  PolReg p1;

  p1.numLati = 3;
  p1.lunghezzaLato = 10.0;

  float perimetro = p1.numLati * p1.lunghezzaLato;
  cout << "Il perimetro di p1 e' " << perimetro << " cm" << endl;

  return 0;
}
```

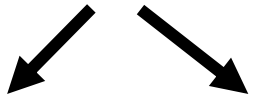
Strutture come parametri di funzioni

```
struct PolReg {
  int numLati;
  float lunghezzaLato;
};

float calcolaPerimetro(int numero, float lun) {
  float perimetro = numero * lun;
  return perimetro;
}

int main() {
  PolReg p1;
  p1.numLati = 3;
  p1.lunghezzaLato = 10.0;
  float perimetro = calcolaPerimetro(p1.numLati, p1.lunghezzaLato);
  cout << "Il perimetro di p1 e' " << perimetro << " cm" << endl;
  return 0;
}
```

Parametri formali



I singoli campi della struttura sono i parametri attuali della funzione



Strutture come parametri di funzioni

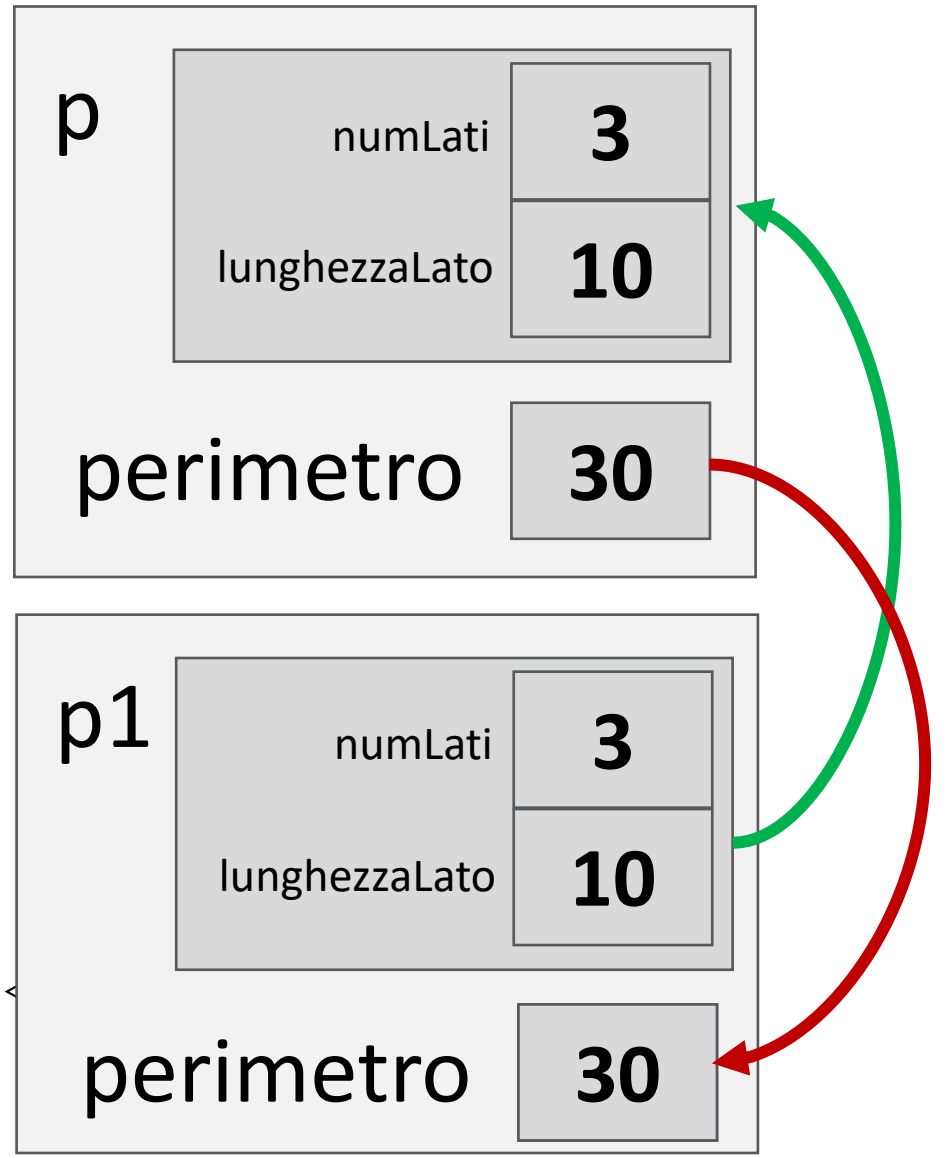
```
struct PolReg {
  int numLati;
  float lunghezzaLato;
};

float calcolaPerimetro(PolReg p) {
  float perimetro = p.numLati * p.lunghezzaLato;
  return perimetro;
}

int main() {
  PolReg p1;
  p1.numLati = 3;
  p1.lunghezzaLato = 10.0;
  float perimetro = calcolaPerimetro(p1);
  cout << "Il perimetro di p1 e' " << perimetro << " cm" << endl;
  return 0;
}
```

Parametro formale di tipo struct

Parametro attuale di tipo struct

Strutture come parametri di funzioni

```
struct PolReg {
  int numLati;
  float lunghezzaLato;
};

float calcolaPerimetro(PolReg p) {
  float perimetro = p.numLati * p.lunghezzaLato;
  return perimetro;
}

int main() {
  PolReg p1;
  p1.numLati = 3;
  p1.lunghezzaLato = 10.0;
  float perimetro = calcolaPerimetro(p1);
  cout << "Il perimetro di p1 e' " << perimetro <<
  return 0;
}
```



Strutture come parametri di funzioni

```
struct PolReg {
  int numLati;
  float lunghezzaLato;
};

float calcolaPerimetro(PolReg p) {
  float perimetro = p.numLati * p.lunghezzaLato;
  return perimetro;
}

int main() {
  PolReg p1;
  p1.numLati = 3;
  p1.lunghezzaLato = 10.0;
  float perimetro = calcolaPerimetro(p1);
  cout << "Il perimetro di p1 e' " << perimetro << " cm" << endl;
  return 0;
}
```

Strutture come parametri di funzioni

```
struct PolReg {
  int numLati;
  float lunghezzaLato;
};

void inizializza(PolReg p, int n, float l) {
  p.numLati = n;
  p.lunghezzaLato = l;
}

int main() {
  PolReg p1;
  inizializza(p1, 3, 10.0);
  float perimetro = calcolaPerimetro(p1);
  cout << "Il perimetro di p1 e' " << perimetro << " cm" << endl;
  return 0;
}
```

ATTENZIONE!

La funzione modifica l'oggetto di nome *p*. Al termine della funzione, l'oggetto *p* viene distrutto**.** Le modifiche **non** si riflettono sull'oggetto *p1* nel main

Strutture come parametri di funzioni

```
struct PolReg {
  int numLati;
  float lunghezzaLato;
};

void inizializza(PolReg& p, int n, float l) {
  p.numLati = n;
  p.lunghezzaLato = l;
}

int main() {
  PolReg p1;
  inizializza(p1, 3, 10.0);
  float perimetro = calcolaPerimetro(p1);
  cout << "Il perimetro di p1 e' " << perimetro << " cm" << endl;
  return 0;
}
```

Strutture come risultato di funzioni

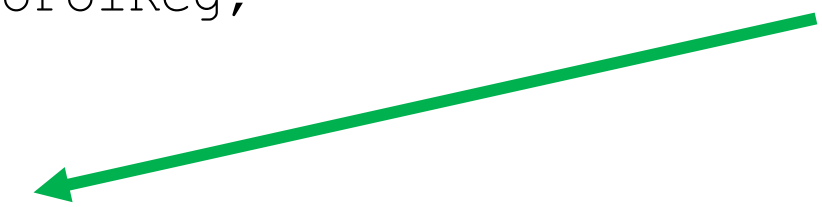
```
struct PolReg {
  int numLati;
  float lunghezzaLato;
};

PolReg inizializza(int n, float l) {
  PolReg nuovoPolReg;
  nuovoPolReg.numLati = n;
  nuovoPolReg.lunghezzaLato = l;
  return nuovoPolReg;
}
int main() {
  PolReg p1;
  p1 = inizializza(3, 10.0);
  float perimetro = calcolaPerimetro(p1);
  cout << "Il perimetro di p1 e' " << perimetro << " cm" << endl;
  return 0;
}
```

Restituisce un tipo struct PolReg

La funzione crea un nuovo oggetto di nome *nuovoPolReg* e lo restituisce al chiamante

Strutture come risultato di funzioni

```
struct PolReg {
  int numLati;
  float lunghezzaLato;
};

PolReg inizializza(int n, float l) {
  PolReg nuovoPolReg;
  nuovoPolReg.numLati = n;
  nuovoPolReg.lunghezzaLato = l;
  return nuovoPolReg;
}
int main() {
  PolReg p1;
  p1 = inizializza(3, 10.0);
  float perimetro = calcolaPerimetro(p1);
  cout << "Il perimetro di p1 e' " << perimetro << " cm" << endl;
  return 0;
}
```



Assegna a p1 l'oggetto restituito dalla funzione

Funzioni che operano su oggetti struttura

```
int main() {
  PolReg p1, p2;

  p1 = inizializza(3,10.0);
  p2 = inizializza(4,20.5);

  float perimetro1 = calcolaPerimetro(p1);
  cout << "Il perimetro di p1 e' " << perimetro1 << " cm" <<endl;
  float perimetro2 = calcolaPerimetro(p2);
  cout << "Il perimetro di p2 e' " << perimetro2 << " cm" <<endl;

  return 0;
}
```