Programmazione Avanzata

uso auto

Pierluigi Roberti

pierluigi.roberti@unitn.it

Stampa lista - auto

```
vector<int>s;
vector<int>::iterator it;

s.push_back(11);
s.push_back(22);
s.push_back(33);
s.push_back(55);

cout << "stampa con iterator:" << endl;;
for (it = s.begin(); it!=s.end(); it++) {
       cout << "["<< *it << "] ";
}
cout << endl << endl;

//utilizzo keyword auto
cout << "utilizzo keyword auto" << endl;
for (auto ita = s.begin(); ita != s.end(); ita++) {
       cout << "["<< *ita << "] ";
}
```

Stampa lista rovesciata - auto

```
vector<int>s;
vector<int>::reverse_iterator rit;

s.push_back(11);
s.push_back(22);
s.push_back(33);
s.push_back(55);

cout << "stampa con reverse iterator:" << endl;;
for (rit = s.rbegin(); rit!=s.rend(); rit++) {
       cout << "["<< *rit << "] ";
}
cout << endl << endl;

//utilizzo keyword auto
cout << "utilizzo keyword auto" << endl;
for (auto ita = s.rbegin(); ita != s.rend(); ita++) {
       cout << "["<< *ita << "] ";
}
cout << endl << endl;
```

Uso di auto

Quanto è sicuro in questo caso l'uso della parola chiave auto?
E se il tipo di vettore fosse float? string?
Un compilatore C++ sa qual è il tipo di un'espressione (di inizializzazione)!

```
vector<int>s;

for (int it=s.begin();it!=s.end();it++){
       cout<<*it<<endl;
}
```

[**Error**] cannot convert 'std::vector<int>::iterator {aka __gnu_cxx::__normal_iterator<int*, std::vector<int> >}' to 'int' in initialization

Uso di auto

In C++11 si può scrivere

```
#include <vector>
#include <algorithm>    // std::reverse

cout << "utilizzo keyword auto for c++11" << endl;
for (auto& it : s) {
       cout << "["<< it << "] ";
}
cout << endl;

reverse( s.begin(),s.end() );     //serve libreria algorithm

for (auto& it : s ) {
       cout << "["<< it << "] ";
}
cout << endl;
```

Uso di auto

```
cout << "utilizzo keyword auto for c++11" << endl;
//scorro per copia
for (auto item : s) {
	cout << "["<< item << "] ";
	item++; //modifica del contenuto non ha effetto
}
cout << endl;

//scorro per riferimento
for (auto& item : s) {
	cout << "["<< item << "] ";
	item++; //modifica del contenuto
}
cout << endl;

//scorro per riferimento const
for (const auto& item : s) {
	cout << "["<< item << "] ";
	//item++; //no modifica del contenuto
}
cout << endl;
```

Uso di auto

La parola chiave auto ottiene il tipo dall'espressione a destra dell' =.
Pertanto funzionerà con qualsiasi tipo, l'unico requisito è inizializzare la variabile auto quando la dichiara in modo che il compilatore possa dedurre il tipo.

```
auto a = 0.0f;  // a is float
auto b = std::vector<int>();
// b is std::vector<int>()

//con le funzioni
//invocazione funzione
auto c = foo();
// c is TipoDato
```

```
//definizione tipo dato
class TipoDato{
private:
        int datoI;
        float datoF;
public:
        TipoDato(){datoI=1;datoF=0.0;}
};
//definizione funzione
TipoDato foo()  {
        return TipoDato();
}
```