

Programmazione di sistema

Anno accademico 2014-2015

Esercitazione 1

Si progetti e si realizzi la classe Elenco che rappresenta un contenitore dinamico di Persone. Tale classe offre i seguenti metodi pubblici:

- `int size()` – restituisce il numero di persone contenute nell'elenco;
- `void add(Persona p)` – inserisce il parametro passato al fondo dell'elenco;
- `Persona get(int pos)` – restituisce l'elemento alla posizione `pos`;
- `Persona remove(int pos)` – elimina l'elemento alla posizione `pos` e lo restituisce;
- `void clear()` – rimuove tutte le persone dall'elenco.

La classe `Persona` incapsula un nome ed un cognome.

Entrambe le classi devono essere copiabili ed assegnabili. Si faccia attenzione a rilasciare correttamente tutta la memoria allocata.

Per testare il corretto funzionamento si utilizzi il seguente main:

```
int _tmain(int argc, _TCHAR* argv[]){
    Elenco e1;
    e1.add(Persona("a", "b"));
    e1.add(Persona("c", "d"));
    e1.add(Persona("e", "f"));
    Elenco e2=e1;
    e1.remove(1);
    e2.remove(2);
    for(int i=0;i<e1.size();i++)
        cout<<i<<": "<<e1.get(i).getNome()<<" "<<e1.get(i).getCognome()<<endl;
    cout<<endl;
    for(int i=0;i<e2.size();i++)
        cout<<i<<": "<<e2.get(i).getNome()<<" "<<e2.get(i).getCognome()<<endl;
    cout<<endl;
    e1.clear();
    cout<<e1.size()<<endl;
    return 0;
}
```

Il cui output su video è

0: a b
1: e f

0: a b
1: c d

0

Competenze da acquisire

- Gestione della memoria
- Incapsulamento
- Copia in profondità