

Corso Programmazione ad Oggetti
Corso di Laurea in Informatica
Esercitazione di Laboratorio del 19/11/2018

Esercizio 1. Utilizzare l'ereditarietà per realizzare la classe **ListaOrdinataStringhe** che implementa una lista di stringhe in cui gli elementi sono rigorosamente ordinati in ordine lessicografico. NB: evitare l'uso della funzione *sort*.

Esercizio 2. Al pronto soccorso, i pazienti vengono registrati e gli viene fornito un codice di gravità. La classe **Paziente** è definita nel seguente modo:

```
#ifndef PAZIENTE_H_
#define PAZIENTE_H_

#include <iostream>
#include <string>
using namespace std;

enum CodiceUrgenza {
    BIANCO = 0, VERDE, GIALLO, ROSSO
};

class Paziente {
private:
    string nome;
    CodiceUrgenza codice;
public:
    Paziente() : nome(""), codice(BIANCO) {}
    Paziente(string n, CodiceUrgenza c) : nome(n), codice(c) {}

    CodiceUrgenza getCodice() const { return codice; }
    void setCodice(CodiceUrgenza codice) { this->codice = codice; }

    const string& getNome() const { return nome; }
    void setNome(const string& nome) { this->nome = nome; }

    bool operator==(const Paziente& p) const { return codice == p.codice; }
    bool operator<(const Paziente& p) const { return codice < p.codice; }
    bool operator<=(const Paziente& p) const { return codice <= p.codice; }
    bool operator>(const Paziente& p) const { return codice > p.codice; }
    bool operator>=(const Paziente& p) const { return codice >= p.codice; }

};

#endif
```

Utilizzare l'ereditarietà per realizzare la classe **CodaPazienti** che implementi una coda con priorità di pazienti, in cui i pazienti che arrivano sono serviti in base al codice di urgenza (da rosso a bianco) e, a parità di urgenza, nel consueto ordine FIFO (First In First Out).

Esercizio 3. Utilizzare l'ereditarietà per realizzare la classe **CodaConGentleman** che implementa una coda di persone in cui, ad ogni estrazione, se il prossimo da servire è un uomo, questo "fa il gentleman" e lascia passare tutte le donne in coda tra lui e l'uomo successivo. Si progetti quindi la classe **Persona**, in modo tale che abbia un metodo che consenta di distinguere il sesso.

Esercizio 4. Utilizzare l'ereditarietà per realizzare la classe **CodaMedicoDiFamiglia** che implementa una speciale coda di persone in cui gli informatori del farmaco hanno diritto di precedenza sui pazienti. Un informatore può passare quando davanti a lui ci siano meno di 10 persone, altrimenti deve attendere che questa condizione si verifichi.

Corso Programmazione ad Oggetti
Corso di Laurea in Informatica
Esercitazione di Laboratorio del 19/11/2018

Esercizio 5. Realizzare una classe **Computer**. Ogni oggetto di tale classe deve essere caratterizzato almeno dalle seguenti informazioni:

- nomeComputer: il nome di un computer (string);
- connessoA: il computer con cui è connesso in rete, stringa vuota se non esiste (string).

Realizzare una classe **GestoreComputers** contenente una lista di computers. Dotare tale classe di un metodo che data la seguente definizione:

Un computer C1 è connesso con un Computer C2 se:

- *C1 è connesso con C2, oppure*
- *C1 è connesso con C3 e C3 è connesso con C2*

restituisca il numero massimo di computer connessi tra loro. Nel caso la lista sia vuota, restituire -1.

Alcuni suggerimenti per lo svolgimento:

- Non utilizzare caratteri accentati per nomi di campi, variabili o metodi.
- Per il carattere tilde (~) occorre digitare: ALT + 126 su sistemi Windows, ALT gr + ` su sistemi Linux, ALT + 5 su sistemi Mac OS X.
- Per ogni esercizio realizzare una cartella contenente tutti i file creati.
- Si consiglia di utilizzare un semplice editor di testo (ad esempio, gedit), e compilare da linea di comando. Per compilare occorre passare tutti i file sorgente (.cpp) al compilatore.
- Minimizzare le inclusioni.