



Università degli studi di Bergamo

---

SCUOLA DI INGEGNERIA

Corso di Laurea Magistrale in Ingegneria Informatica

# RELAZIONE PROGETTO C++

## CORSO DI PROGRAMMAZIONE AVANZATA

*Giulia Allievi*  
*Matricola: 1058231*

---

Anno Accademico 2021-2022



## Introduzione

L'applicazione progettata permette di gestire le informazioni relative a delle visite mediche, mediante l'inserimento di alcune informazioni riguardanti l'esame, il paziente sul quale questo viene effettuato e il medico che lo esegue.

## Funzionamento

L'applicazione è formata da 8 classi: Persona, Dottore, Paziente, Esame, Pet, Mr, PetMr e Metodi (una classe di gestione). Una persona può essere o un paziente o un dottore, entrambi ereditano dalla classe persona in modo pubblico. Entrambe le classi sono caratterizzate da un ID, la classe Dottore possiede un campo specializzazione mentre la classe Paziente un campo per memorizzare la categoria alla quale appartiene. Ereditano i campi di nome, cognome, codice fiscale e anno di nascita dalla classe Persona. I campi nome e cognome sono protected, mentre i campi relativi al codice fiscale e all'anno di nascita sono public. La gerarchia delle classi che descrivono gli esami formano un diamante, perché sia la classe Pet che la classe Mr hanno come classe base Esame, la classe PetMr invece rappresenta un esame ibrido fra i due precedenti, questa classe eredita sia dalla classe Pet che dalla classe Mr (mediante il meccanismo dell'ereditarietà multipla del C++). La classe Metodi raccoglie invece tutte le funzionalità che vengono offerte dal programma e che sono poi richiamate nel main del programma. L'utente digita la lettera corrispondente all'operazione che intende eseguire, sarà quindi richiamato il metodo corrispondente a quella determinata azione.

## Scelte implementative

Stl, ered multi, (guarda sito così metti tutto). I metodi che richiamano i costruttori di una classe sono private così utente non può creare a caso e così come metodi di controllo

## Diagramma delle classi

Di seguito si riportano i diagrammi delle classi dell'applicazione.

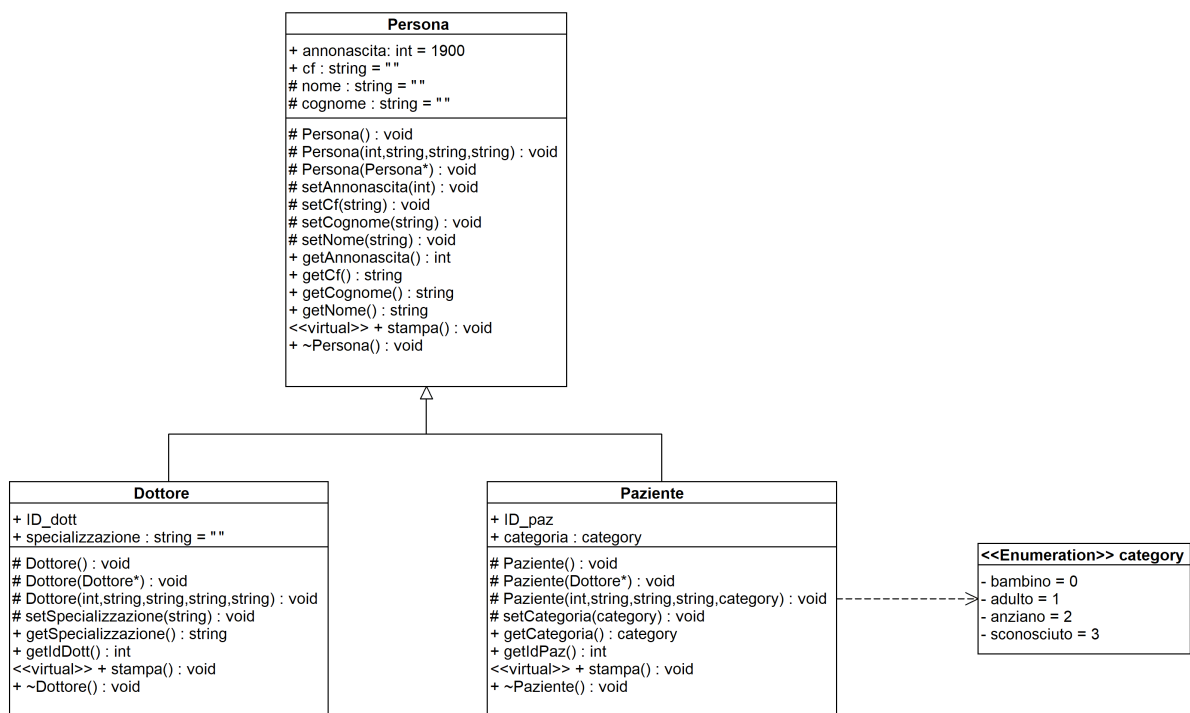


Figura 1: Diagramma delle classi - Gerarchia utenti.

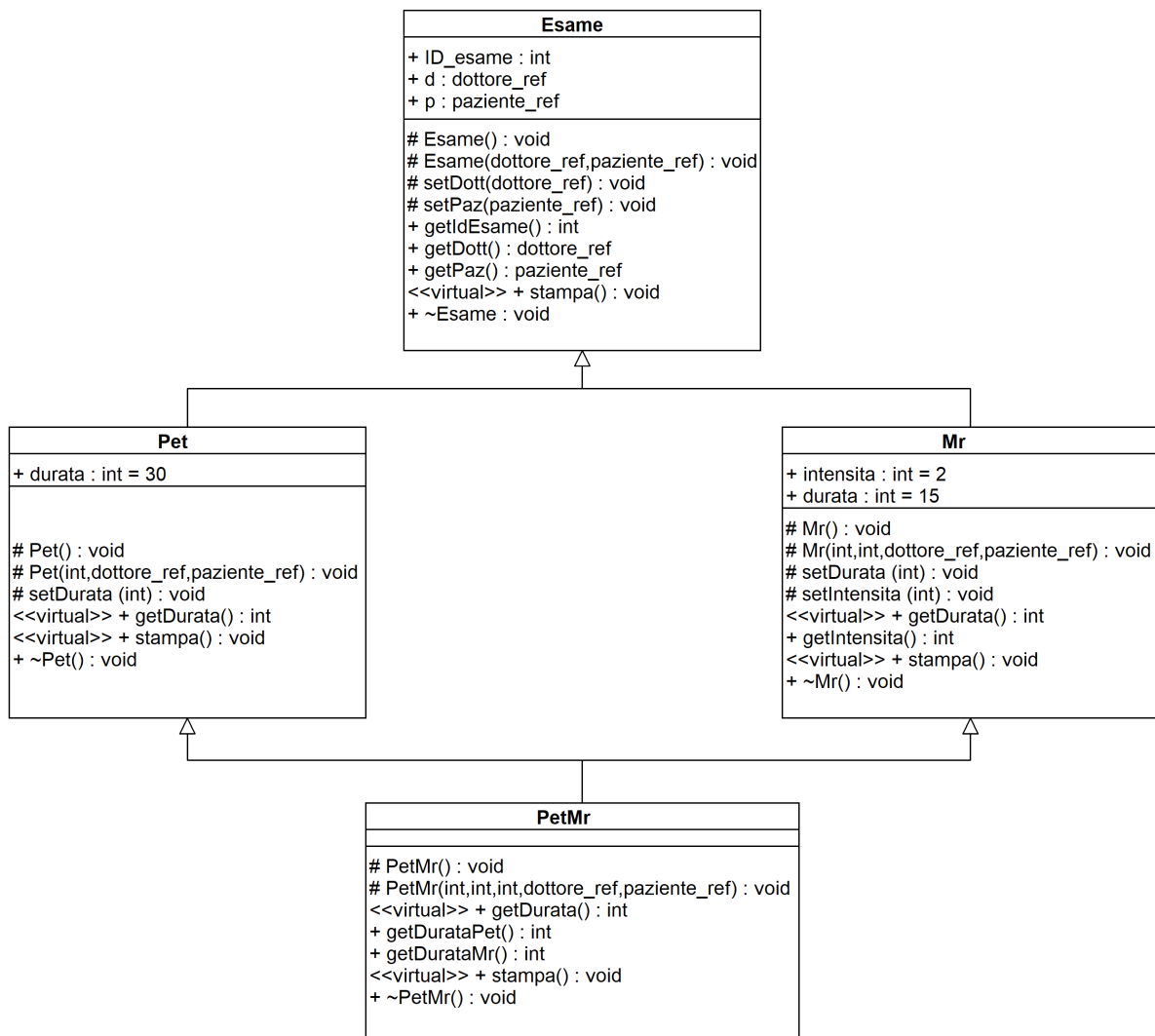


Figura 2: Diagramma delle classi - Gerarchia esami.

| Metodi  |
|---|
| <ul style="list-style-type: none"> <li>- data : time_t</li> <li>- now : tm*</li> <li>- annocorrente : int</li> <li>- lista_pazienti : vector&lt;paziente_ref&gt;</li> <li>- lista_dottori : vector&lt;dottore_ref&gt;</li> <li>- lista_esami : vector&lt;esame_ref&gt;</li> <li>- lista_esami_pet : vector&lt;pet_ref&gt;</li> <li>- lista_esami_mr : vector&lt;mr_ref&gt;</li> <li>- lista_esami_petmr : vector&lt;petmr_ref&gt;</li> </ul>  |
| <ul style="list-style-type: none"> <li>- creaPaziente() : paziente_ref</li> <li>- creaPaziente(Paziente*) : paziente_ref</li> <li>- creaPaziente(int,string,string,string,category) : paziente_ref</li> <li>- creaDottore() : dottore_ref</li> <li>- creaDottore(Dottore*) : dottore_ref</li> <li>- creaDottore(int,string,string,string,string) : dottore_ref</li> <li>- creaEsame() : esame_ref</li> <li>- creaEsame(dottore_ref,paziente_ref) : esame_ref</li> <li>- creaEsamePet() : pet_ref</li> <li>- creaEsamePet(int,dottore_ref,paziente_ref) : pet_ref</li> <li>- creaEsameMr() : mr_ref</li> <li>- creaEsameMr(int,int,dottore_ref,paziente_ref) : mr_ref</li> <li>- creaEsamePetMr() : esame_ref</li> <li>- creaEsamePetMr(int,int,int,dottore_ref,paziente_ref) : esame_ref</li> <li>- caricadati() : void</li> <li>- ctrl_paz(string) : bool</li> <li>- ctrl_dott(string) : bool</li> <li>- ctrl_paz(int) : int</li> <li>- ctrl_dott(int) : int</li> <li>- cerca_paz(int) : void</li> <li>- cerca_dott(int) : void</li> <li>+ Metodi() : void</li> <li>+ inserisciPaziente() : void</li> <li>+ stampaElenco_paz() : void</li> <li>+ ordinaElenco_ID_paz() : void</li> <li>+ ordinaElenco_Categoria_paz() : void</li> <li>+ inserisciDottore();</li> <li>+ stampaElenco_dott() : void</li> <li>+ ordinaElenco_ID_dott() : void</li> <li>+ ordinaElenco_Specializzazione_dott() : void</li> <li>+ inserisciEsame() : void</li> <li>&lt;&lt;virtual&gt;&gt; + stampaElenco_esami() : void</li> <li>+ stampaEsami_ID_dott() : void</li> <li>+ stampaEsami_ID_paz() : void</li> <li>+ inserisciEsamePet() : void</li> <li>&lt;&lt;virtual&gt;&gt; + stampaElenco_esami_pet() : void</li> <li>+ stampaEsami_PET_corti() : void</li> <li>+ inserisciEsameMr() : void</li> <li>&lt;&lt;virtual&gt;&gt; + stampaElenco_esami_mr() : void</li> <li>+ stampaEsami_MR_forti() : void</li> <li>+ inserisciEsamePetMr() : void</li> <li>&lt;&lt;virtual&gt;&gt; + stampaElenco_esami_pet_mr() : void</li> <li>+ stampaEsami_PETMR_corti_forti() : void</li> <li>+ getYear() : int</li> <li>+ ~ Metodi : void</li> </ul> |

Figura 3: Diagramma delle classi - Metodi resi disponibili nel programma principale.