

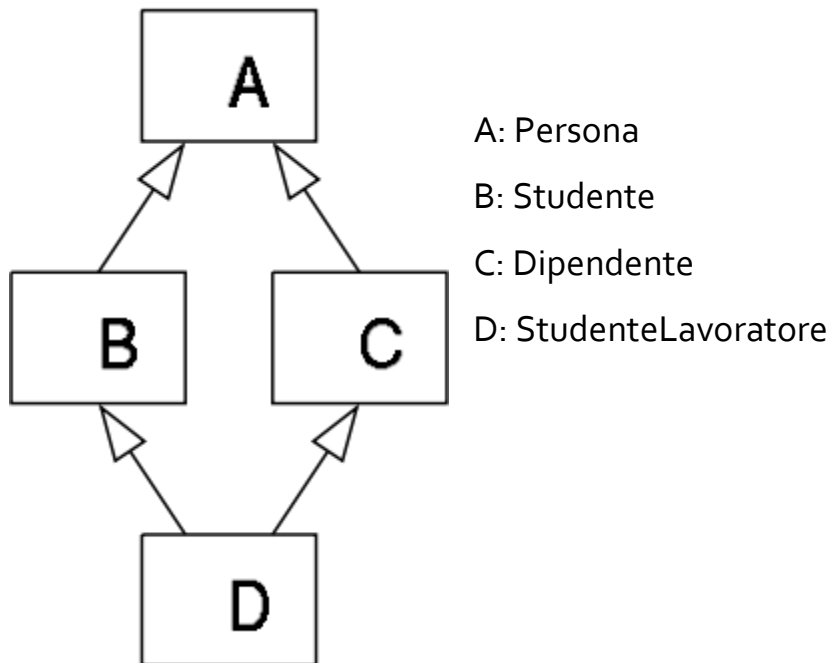
Marco Milesi – 1030184 – C++

Realizzazione di un progetto C++
con l'applicazione di diversi costrutti e ereditarietà multipla

INFORMATICA 3A
Università degli Studi di Bergamo

Introduzione

Il progetto realizzato in C++ descrive la struttura organica di una generica università in cui il capitale umano può essere dipendente, studente oppure entrambe le cose (esempio nell'ambito di progetti lavorativi interni all'istituto). Ogni entità è realizzata attraverso una classe separata, secondo lo schema seguente:



La gestione di ogni classe è realizzata in file separati (header e corpo definizione cpp), in aggiunta al file contenente il main di avvio.

La memorizzazione di ogni entità passa attraverso una classe ulteriore, definita "Lista", creata per agevolare la gestione delle chiamate in modo da rendere il codice più pulito, specialmente all'interno del <main>. La classe Lista, infatti, ha un metodo aggiungi(Persona p) che consente di effettuare un push_back di p su un oggetto <list>. Avere a disposizione questa classe ausiliaria consente inoltre di scalare meglio le funzionalità, per esempio aggiungendo nuovi metodi per filtrarne i dati.

La classe Lista calcola inoltre l'età media degli studenti utilizzando una funzione ausiliaria di **tipo T-generico** (template).

Ogni classe ha un proprio distruttore e un distruttore (sfruttato in caso di delete sull'oggetto). Inoltre i vari campi e metodi sono dichiarati **private, public o protected** a seconda dei casi (per esempio, i metodi sono pubblici e i campi protected o private a seconda del grado di "protezione" desiderato).

Molti metodi sono dichiarati **virtual** per consentirne un override nelle classi figlie.

Ogni classe contiene un metodo toString() di ritorno void che effettua un cout diretto dell'elemento stesso, secondo un pattern hard-coded per ogni oggetto. Lo stesso metodo è presente inoltre nella classe Lista, e si appoggia ad un iteratore per effettuare la lettura dei vari elementi che contiene. Ogni elemento chiamato con toString() riporta a schermo il tipo dell'oggetto: S (studente), P (persona), D (dipendente), SD (studente dipendente).

Per restrizioni di MINGW su Windows la stampa a schermo di interi è demandata ad una prima fase di conversione utilizzando la libreria **sstream**.

Il main dell'applicazione effettua diverse prove, tra cui l'inserimento dei vari oggetti nella lista e l'invocazione di alcuni metodi sulla lista stessa:

- stampa: per la stampa di tutti gli elementi
- stampa(1994): per la stampa di tutti gli elementi nati nel 1994
- etaMedia: per la stampa dell'età media (template generico T)

Esempio di Output

(Nomi e matricole di fantasia)

```
# STAMPA generica #
[P: Marco, Milesi, 1994]
[P: Luca, ABC, 1995]
[S: Marco, Milesi, 1030184, Informatica]
[S: Mario, Rossi, 1030185, Informatica]
[D: Giovanni, Rossi, 1965, 2005@27, Salvecchio]
[D: Ruggero, Silvani, 1955, 1997@30, Dalmine]
[P: Chiara, Agazzi, 1995]
[SD: Chiara, Morenti, 1975, 1995@30, Agostino, SCADE: ottobre 2017]
# STAMPA 1994 #
[P: Marco, Milesi, 1994]
[S: Marco, Milesi, 1030184, Informatica]
# ETA' MEDIA TEMPLATE GENERICO #
Età media: 1983
```