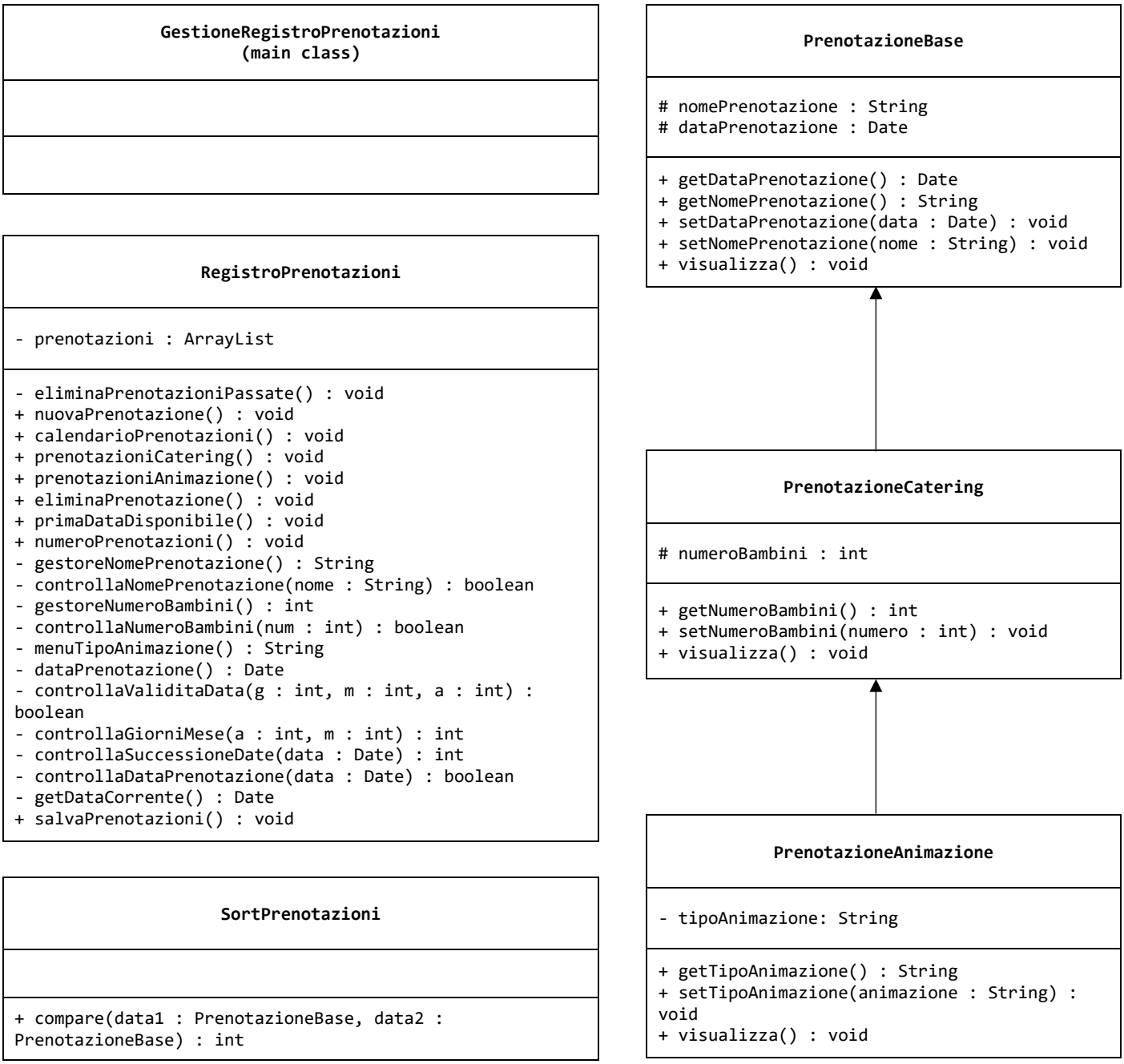
# Descrizione della struttura del progetto

Righetti Valentina (matr. 604013)

Programmazione e Analisi di Dati
Modulo A – Laboratorio di Programmazione Java
a.a. 2019/2020

## Struttura del progetto:



## Descrizione del progetto

Il programma è stato pensato e realizzato per essere utilizzato dal proprietario/gestore del locale, per tenere traccia delle prenotazioni (ognuna con le sue caratteristiche) che vengono effettuate. Per questo motivo, ad esempio, le prenotazioni vengono visualizzate con tutti i dati che, altrimenti, da lato client, dovrebbero essere nascoste (come il nome della prenotazione).

La classe `GestioneRegistroPrenotazioni` (che contiene il main) inizializza un nuovo oggetto `registro` della classe `RegistroPrenotazioni.` Manda in output l'interfaccia testuale (menu) e permette all'utente di scegliere quale opzione preferisce.

La classe `RegistroPrenotazioni` funge da *terminale* del programma, definendo i metodi che permettono a quest'ultimo di svolgere le diverse opzioni implementate. Le prenotazioni vengono inserite in un ArrayList di tipo `<PrenotazioneBase>` come oggetti e poi, una volta che il programma viene terminato (opzione "Salva ed esci" del menu), vengono scritte su un file binario. Se il file esiste già, al momento dell'esecuzione del programma gli oggetti già scritti sul file (se presenti) vengono letti e inseriti nell'ArrayList. Una scelta implementativa è stata quella di eliminare le prenotazioni in data precedente alla data corrente, se presenti nel file binario.
Oltre ai metodi che permettono di implementare le opzioni del menu sono presenti i metodi di controllo sugli input dell'utente (`controllaNomePrenotazione`, `controllaNumeroBambini`, `controllaValiditaData`, `controllaGiorniMese` e `controllaDataPrenotazione`) e il metodo per la scrittura delle prenotazioni sul file binario (`salvaPrenotazioni`).

La classe `PrenotazioneBase` definisce la prenotazione "base" del solo locale. È la superclasse delle sottoclassi `PrenotazioneCatering` (che definisce una prenotazione che prevede il catering) e `PrenotazioneAnimazione` (quest'ultima a sua volta sottoclasse di `PrenotazioneCatering;` definisce una prenotazione che prevede catering e animazione) che ne ereditano i metodi oltre a ridefinire il metodo `visualizza()` e a definirne di nuovi (relativi al numero di bambini presenti alla festa e al tipo di animazione). Le tre classi implementano l'interfaccia `Serializable` che permette la scrittura/lettura su file.

La classe `SortPrenotazioni` implementa l'interfaccia `Comparator` che permette di ordinare l'ArrayList secondo una delle variabili degli oggetti che contiene; nel caso del programma la variabile è quella della data della prenotazione. Le prenotazioni, insomma, vengono ordinate in ordine cronologico, per poter essere visualizzate correttamente quando si richiamano i metodi `calendarioPrenotazioni`, `PrenotazioniCatering`, `PrenotazioniAnimazione` e `PrimaDataDisponibile`.

## Il metodo `getDataCorrente`

Il metodo `getDataCorrente` restituisce la data corrente; viene invocato quando devono essere effettuati dei confronti fra la data di una prenotazione e la data corrente stessa. Il metodo sfrutta `Calendar` per poter impostare i valori del `time` della data corrente a zero, perché il solo istanziare un nuovo oggetto `Date` con `Date data = new Date()` crea un oggetto con la data corrente e il `time` impostato al momento in cui l'oggetto è stato istanziato. Di conseguenza, se l'utente inserisse, come data di prenotazione del locale, la data corrente (che, automaticamente, ha il `time` a zero) e si facesse un confronto con un oggetto `Date data = new Date()` le due date risulterebbero diverse.