Corso di Programmazione ad Oggetti – Anno 2014/2015
Corso di laurea in Ingegneria e Scienze Informatiche
Università di Bologna - Sede di Cesena

Relazione relativa allo sviluppo di un sistema per la gestione di una galleria d'arte.

Elisa Casadio
Sofia Rosetti

Indice

Capitolo 1: Analisi

1.1 Requisiti

L'applicativo da realizzare dovrà essere in grado di permettere la gestione di una galleria d'arte.

Nello specifico, il sistema dovrà garantire le seguenti operazioni:

- aggiunta, modifica e cancellazione di un'opera d'arte dall'archivio;
- aggiunta, modifica e cancellazione di un'esposizione dall'archivio, gestendo anche l'aggiunta e la cancellazione delle relative opere d'arte;
- acquisto di uno o più biglietti relativi ad una specifica esposizione, a prezzo intero o ridotto, da parte della clientela;
- visualizzazione e calcolo del bilancio attuale della galleria;
- visualizzazione di bilanci di anni precedenti;
- visualizzazione della classifica delle esposizioni più redditizie.

1.2 Problema

L'applicativo dovrà essere in grado di far fronte ai seguenti problemi:

- un'opera d'arte non potrà essere cancellata dall'archivio se utilizzata in un'esposizione attuale o passata;
- un'esposizione non potrà essere aggiunta se nell'archivio non sono ancora presenti opere d'arte;
- un'esposizione e le relative opere d'arte non potranno essere modificate o cancellate se l'esposizione è già attiva o già terminata;
- potranno essere acquistati solamente i biglietti relativi alle esposizioni in atto;
- ogni esposizione dovrà prevedere l'emissione di un massimo di biglietti giornalieri;
- la classifica dovrà tener conto anche delle esposizioni realizzate in anni passati;
- salvataggio automatico dei dati e relativo caricamento all'apertura dell'applicativo.

Capitolo 2 : Design

In questa fase è stata effettuata la progettazione del software, prendendo decisioni riguardanti la risoluzione dei problemi identificati nell'analisi.

2.1 Architettura

Per tutta la fase di progettazione è stato utilizzato il pattern architetturale MVC (model-view-controller) per garantire una netta separazione fra i diversi aspetti di rappresentazione, di gestione e di memorizzazione dei dati. Questa separazione consente un futuro riutilizzo e una facile manutenibilità del modello, delle viste e del controller.

Descrizione degli aspetti principali del model:

- IArtGallery e ArtGallery: rappresentano l'interfaccia e la relativa implementazione dell'archivio della galleria d'arte. Per tenere traccia delle opere d'arte e delle esposizioni che sono o erano presenti nella galleria si è deciso di utilizzare, rispettivamente, List<Artwork> e List<Exhibit>. ArtGallery implementa anche l'interfaccia Serializable per garantire la persistenza dei dati;
- IArtwork e Artwork: rappresentano interfaccia e relativa implementazione che modella una generica opera d'arte. Artwork implementa anche l'interfaccia Serializable per garantire la persistenza dei dati;
- IExhibit e Exhibit: rappresentano interfaccia e relativa implementazione che modella una generica esposizione nella galleria d'arte. Per memorizzare le opere d'arte presenti in quella mostra si è scelto di usare una semplice lista List<Long>, dove vengono memorizzati solamente i codici delle opere. Exhibit implementa anche l'interfaccia Serializable per garantire la persistenza dei dati;
- IExhibitData e ExhibitData: rappresentano interfaccia e relativa implementazione che modella i dati di un'esposizione che sono di interesse per la gestione della biglietteria, quali il numero di biglietti disponibili ed il saldo relativo. ExhibitData implementa anche l'interfaccia Serializable per garantire la persistenza dei dati;
- ITicket, Ticket e ReducedTicket: rappresentano interfaccia e relative implementazioni che modellano rispettivamente un biglietto a prezzo intero ed uno ridotto. L'importo del biglietto ridotto cambia a seconda della percentuale di sconto derivante dall'enumerazione Discount, in cui sono presenti cinque tipologie di riduzioni;
- IPurchase e Purchase: rappresentano interfaccia e relativa implementazione che modella l'acquisto dei biglietti relativi ad un'esposizione;
- ISalesManagement e SalesManagement: rappresentano interfaccia e relativa implementazione che modella la gestione della biglietteria, effettuando i controlli ad essa relativi, quali ad esempio la disponibilità di biglietti. SalesManagement implementa anche l'interfaccia Serializable per garantire la persistenza dei dati;
- IBalance e Balance: rappresentano interfaccia e relativa implementazione che modella il bilancio attuale e quelli passati della galleria d'arte. Balance implementa anche l'interfaccia Serializable per garantire la persistenza dei dati;

- IClassification e Classification: rappresentano interfaccia e relativa implementazione che modella la classifica delle esposizioni fin ora realizzate.

La rappresentazione UML del modello viene mostrata nelle figure 1 e 2.

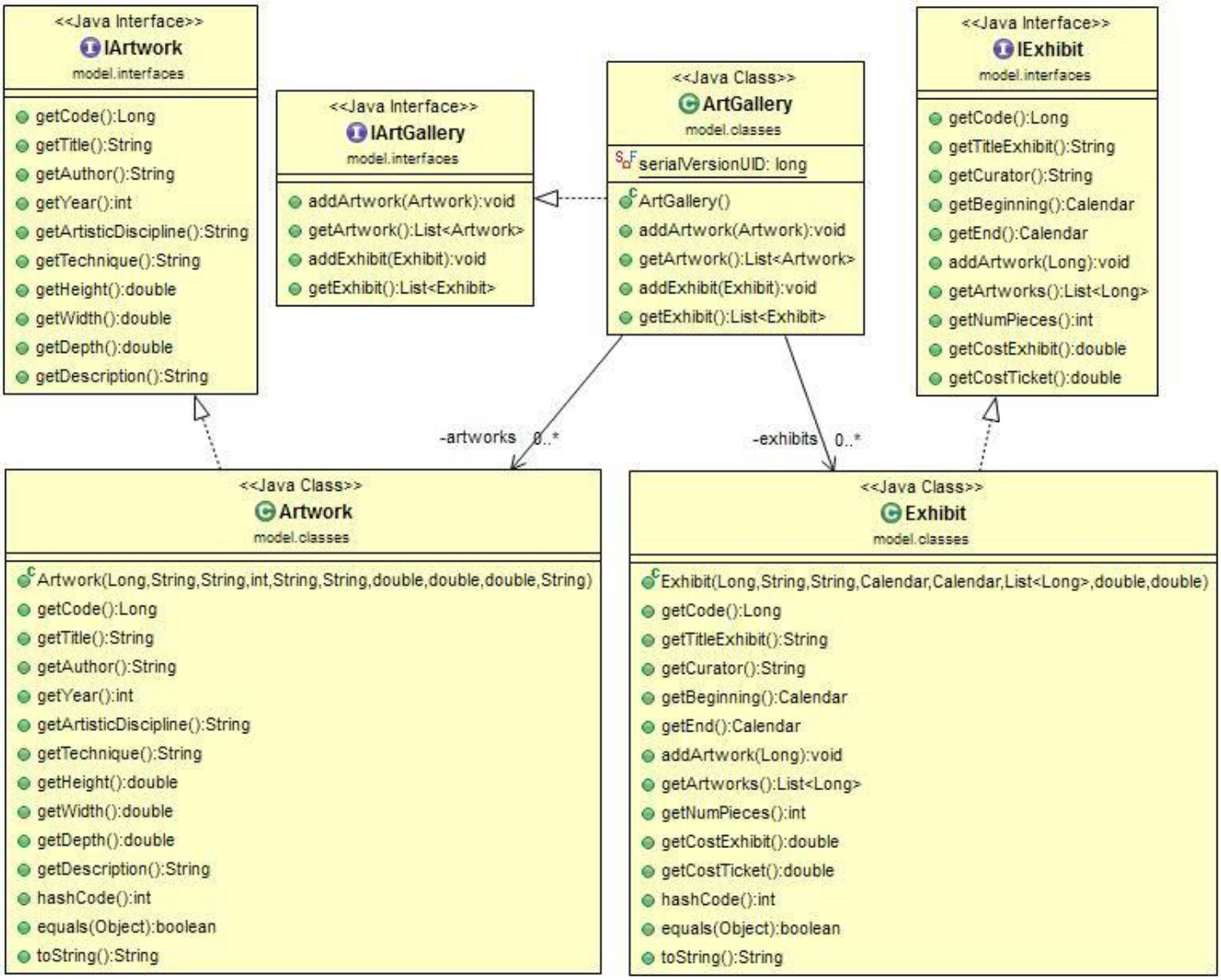

Figura 1: Diagramma UML relativo al modello - parte 1

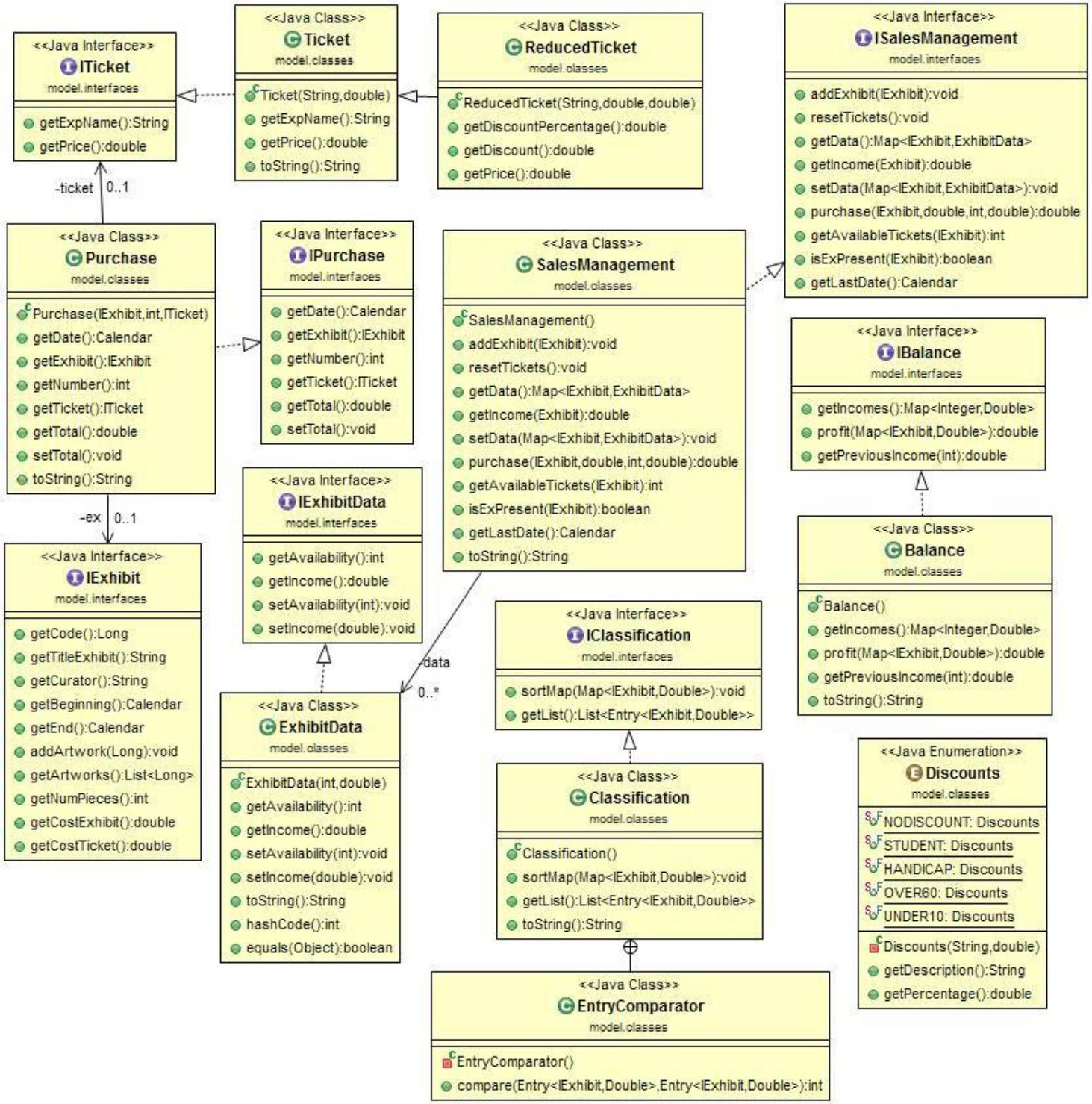

Figura 2: Diagramma UML relativo al modello - parte 2

2.2 Design dettagliato

In questa sezione vengono approfonditi gli elementi di design con maggior dettaglio.

Innanzitutto si è deciso di salvare i dati in automatico su file in un path predefinito ad ogni chiusura delle interfacce relative ad opere d'arte, esposizioni, biglietteria e bilancio. Analogamente, all'apertura delle suddette interfacce, essi vengono caricati in modo da consentire il regolare utilizzo dell'applicativo.

Ogni esposizione presenta una data di inizio ed una data di fine, per cui è stato effettuato un controllo su di esse affinché sia possibile acquistare solamente biglietti per esposizioni in corso.

Il numero di biglietti che possono essere emessi giornalmente è stato deciso arbitrariamente e risulta pari a 500. Il sistema effettua automaticamente e giornalmente un reset dei biglietti disponibili, impedendo inoltre l'acquisto se questi ultimi risultano esauriti.

Il guadagno proveniente da ogni esposizione tiene conto non solo degli incassi relativi alla vendita dei biglietti, ma anche del costo sostenuto dalla galleria d'arte per instaurare la mostra.

Lo storico riguardante le esposizioni ed il relativo guadagno viene mantenuto memorizzandone automaticamente i dati su file. Una volta inserita un'esposizione, essa non viene più rimossa dallo storico, in quanto vengono effettuati i controlli sulle date. In questo modo è possibile mantenere memorizzate tutte le esposizioni, comprese quelle degli anni passati, e stilare la classifica generale.

È stata implementata una classe AdderComponentPanel con il pattern "Singleton", che, attraverso l'unico metodo presente, posiziona un certo componente, come ad esempio una JLabel, un JButton, …, in un pannello che ha come layout il GridBagLayout, settandone tutte le caratteristiche, come ad esempio il numero della riga e della colonna, il numero di celle occupate, la sua posizione nella cella, … È stato scelto di usare questo pattern, perché la stessa istanza veniva chiamata più volte da diverse classi e quindi risultava essere più responsabile tenere traccia di un'unica istanza e impedire che venissero create altri oggetti di questo tipo.

Descrizione degli aspetti principali riguardanti la parte di vista dell'applicazione.

L'applicazione si compone di molteplici viste, per cui è stato deciso di implementare una view principale costituita dall'interfaccia IMainView unitamente alla sua implementazione tramite la classe MainView. Tale vista è in grado di presentare cinque diversi pulsanti tramite i quali è possibile spostarsi tra le sezioni riguardanti: archivio opere d'arte, archivio esposizioni, biglietteria, bilancio, classifica.

La vista relativa all'archivio delle opere d'arte è stata realizzata tramite l'interfaccia IArtworkView e la sua implementazione ArtworkView. In questa sezione vengono visualizzate, in una tabella, tutte le opere d'arte presenti nell'archivio.
Attraverso l'utilizzo di una serie di pulsanti, l'utente può scegliere di tornare alla vista principale, di aggiungere una nuova opera d'arte all'archivio, di modificare un'opera d'arte già presente, di visualizzare i dati completi di un'opera d'arte già presente e di cancellare un'opera d'arte già presente nell'elenco. Nel caso in cui l'utente scelga una delle ultime tre operazioni sarà obbligato a selezionare prima una riga della tabella.
Nel caso in cui venga scelto di aggiungere una nuova opera si aprirà una nuova interfaccia realizzata dalla classe ArtworkForm, che implementa IArtworkForm. Questa interfaccia permette all'utente di compilare tutti i dati riguardanti l'opera, come ad esempio il titolo, l'autore, l'anno di realizzazione, …, e infine di confermarli o annullarli. Essa si apre anche nel caso in cui si voglia modificare o visualizzare i dati di un'opera già presente nell'archivio. In entrambi i casi l'interfaccia sarà già compilata, ma solo nel caso della modifica i cambiamenti apportati saranno salvati.

La vista relativa all'archivio delle esposizioni è stata realizzata tramite l'interfaccia IExhibitView e la sua implementazione ExhibitView. In questa sezione vengono visualizzate, in una tabella, tutte le esposizioni passate, presenti e future presenti nell'archivio della galleria.
Attraverso l'utilizzo di una serie di pulsanti, l'utente può scegliere di tornare alla vista principale, di aggiungere una nuova esposizione, di modificare una nuova esposizione (solo se selezionata precedentemente una riga), di cancellare una esposizione (solo selezionando una riga precedentemente) e di associare le opere d'arte ad una certa esposizione.
Nel caso in cui venga scelto di aggiungere una nuova esposizione si aprirà una nuova interfaccia realizzata dalla classe ExhibitForm, che implementa IExhibitForm. Questa interfaccia permette all'utente di compilare tutti i dati relativi all'esposizione, come il titolo, il curatore, il costo dell'esposizione, …, e infine di confermarli o annullarli. Essa è utilizzata, già compilata, anche per la modifica.
Nel caso in cui venga scelto di associare opere d'arte a una certa esposizione, andrà innanzitutto selezionata l'esposizione prescelta e poi si aprirà una nuova interfaccia realizzata dalla classe ManageArtworkExhibit, che implementa IManageArtworkExhibit. Questa interfaccia permette all'utente di selezionare le opere d'arte presenti nell'archivio (presenti nella tabella di destra) e di aggiungerle all'esposizione (tabella di sinistra) attraverso l'utilizzo di un pulsante "Aggiungi". Dall'elenco delle opere associate all'esposizione possono essere rimosse le opere utilizzando un altro pulsante "Cancella".

La vista relativa alla biglietteria è stata realizzata tramite l'interfaccia ITicketOfficeView e la sua implementazione TicketOfficeView.

In questa sezione vengono visualizzate le esposizioni attive, per cui quindi è possibile acquistare biglietti, attraverso una tabella sulla sinistra in cui una sola riga è selezionabile.
Sulla destra si può trovare una lista contenente quattro (cinque, se si considera anche l'assenza di sconti) possibili sconti applicabili ai biglietti, quali:

- studente, la cui percentuale di sconto è pari al 40%
- disabile, la cui percentuale di sconto è pari al 70%
- over 60, la cui percentuale di sconto è pari al 25%
- under 10, la cui percentuale di sconto è pari al 30%

Sotto tale lista è data la possibilità di inserire il numero di biglietti desiderati per la categoria sopraindicata; tale numero dovrà essere compatibile con l'attuale disponibilità di biglietti, in caso contrario verrà mostrato un messaggio di errore.
Il pulsante "CONFERMA" consente di visualizzare il totale dell'acquisto, mentre il pulsante "HOME" permette di tornare alla vista principale.

La vista relativa al bilancio è stata realizzata tramite l'interfaccia IBalanceView e la sua implementazione BalanceView.
In questa sezione, sulla sinistra, vengono visualizzate (senza particolari ordinamenti) le diverse esposizioni che sono state aperte nell'anno in corso, unitamente al relativo guadagno, che risulterà negativo nel caso in cui l'incasso relativo alla vendita dei biglietti sia minore del costo sostenuto dalla galleria d'arte per instaurare l'esposizione.
Sulla destra è presente il pulsante "Totale anno corrente" che se premuto mostra il totale dei guadagni relativi all'anno corrente.
Più in basso, come indicato dalla scritta, è possibile inserire un anno precedente in modo tale da visualizzarne il relativo saldo una volta premuto il pulsante "Cerca".
Il pulsante "HOME", permette di tornare alla vista principale.

La vista relativa alla classifica delle esposizioni è stata realizzata tramite l'interfaccia IClassificationView e la sua implementazione ClassificationView.
In questa sezione sono visibili tutte le esposizioni che sono state sin ora aperte, con il relativo guadagno, mostrate in ordine decrescente partendo da quella più redditizia.
Analogamente alle due viste precedenti, il pulsante "HOME" consente di tornare alla vista principale.

La rappresentazione UML della vista viene mostrata nelle figure 3 e 4.

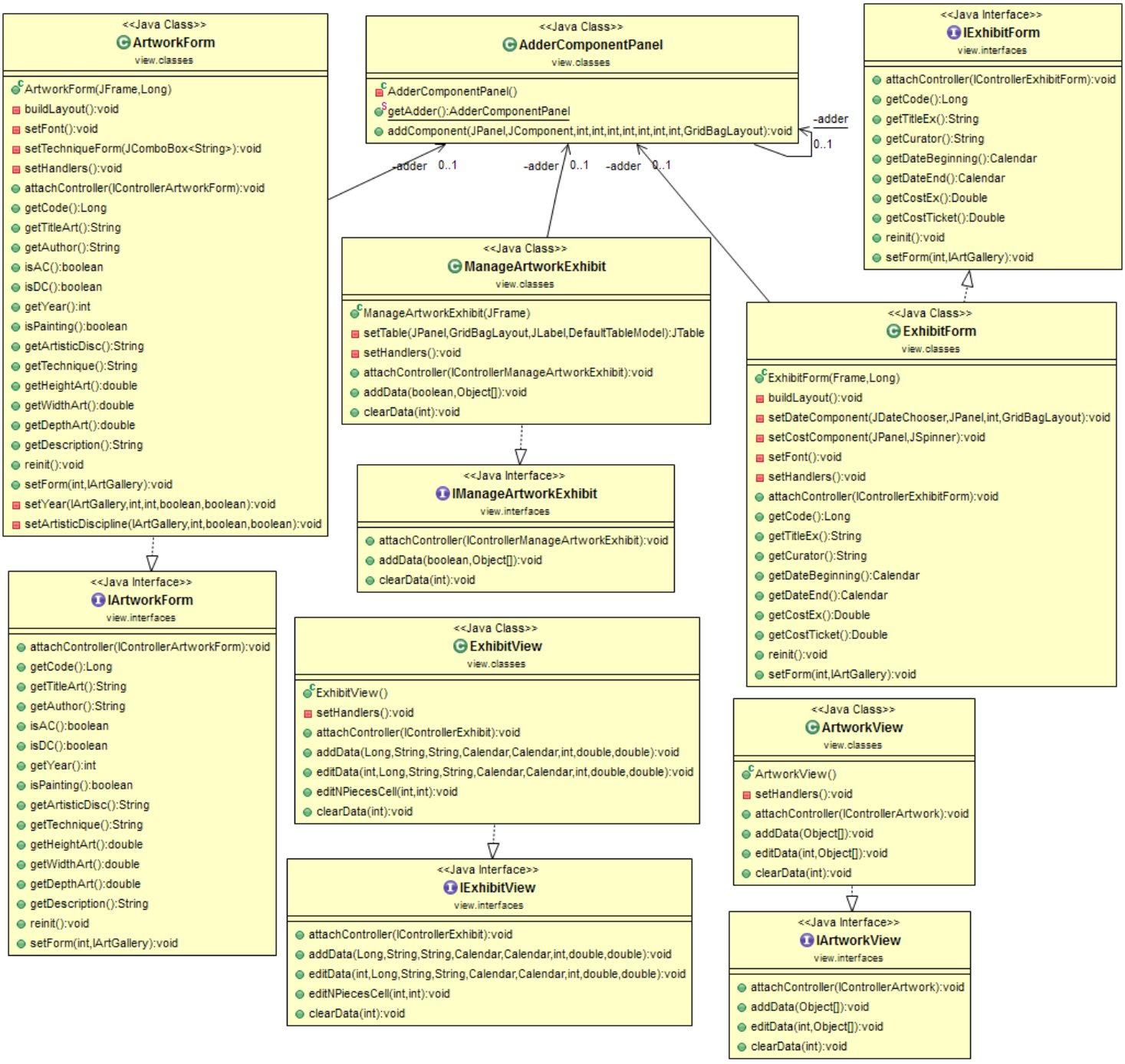

Figura 3: Diagramma UML relativo alla vista - parte 1

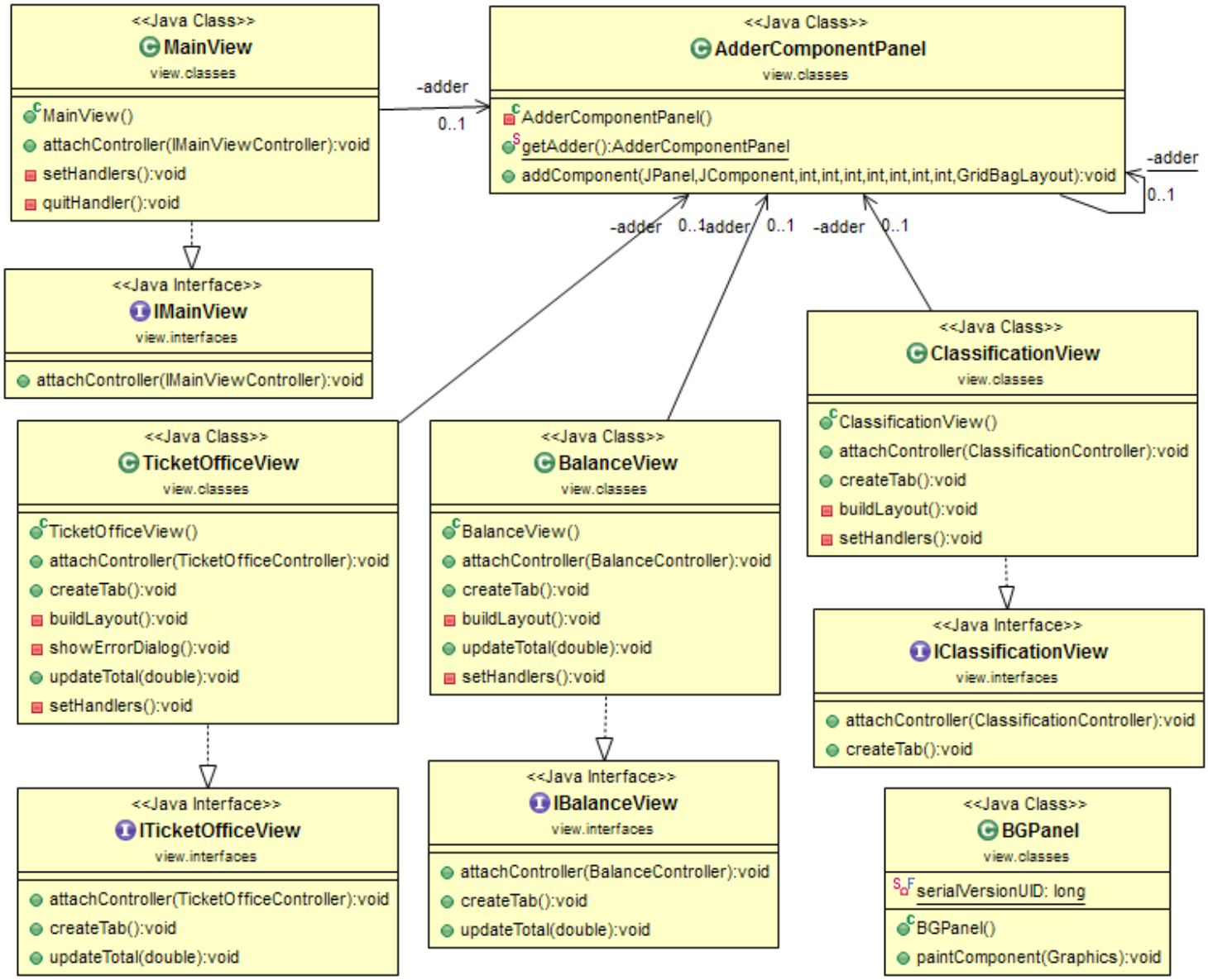

Figura 4: Diagramma UML relativo alla vista - parte 2

Descrizione degli aspetti principali riguardanti la parte di controllo dell'applicazione.

Ogni vista è collegata ad un apposito controller, che ne gestisce il funzionamento.

La vista principale dell'applicativo è controllata dall'interfaccia IMainViewController e dalla sua implementazione MainViewController. Tale controller gestisce la pressione dei cinque pulsanti e si occupa del relativo caricamento dei dati.

La vista inerente all'archivio delle opere d'arte è controllata dall'interfaccia IControllerArtwork e dalla sua implementazione ControllerArtwork. Tale controller gestisce gli eventi derivanti dalla pressione dei pulsanti, come ad esempio permettere la cancellazione di un'opera che non è mai stata utilizzata in un'esposizione, ed il salvataggio dei dati. Nel caso

in cui la correttezza delle operazioni svolte non venga verificata, viene lanciata un'eccezione e viene mostrato a video un messaggio di errore.

La vista inerente alla gestione dei dati di un'opera d'arte è controllata dall'interfaccia IControllerArtworkForm e dalla sua implementazione ControllerArtworkForm. Tale controller gestisce la correttezza dei campi compilati dall'utente, controllando ad esempio che tutti i campi siano stati compilati o che certi campi (come ad esempio la dimensione) rientrino in un certo range di valori. Nel caso in cui la correttezza non venga verificata, viene lanciata un'eccezione e viene mostrato a video un messaggio di errore.

La vista inerente all'archivio delle esposizioni è controllata dall'interfaccia IControllerExhibit e dalla sua implementazione ControllerExhibit. Tale controller gestisce gli eventi derivanti dalla pressione dei pulsanti, come ad esempio permettere la modifica di una esposizione solamente se già in atto, e il salvataggio dei dati su file. Nel caso in cui la correttezza delle operazioni svolte non venga verificata, viene lanciata un'eccezione e viene mostrato a video un messaggio di errore.

La vista inerente alla gestione dei dati di un'esposizione è controllata dall'interfaccia IControllerExhibitForm e dalla sua implementazione ControllerExhibitForm. Tale controller gestisce la correttezza dei campi compilati dall'utente, controllando ad esempio che tutti i campi siano stati compilati o che certi campi (come ad esempio il costo del biglietto) rientrino in un certo range di valori. Nel caso in cui la correttezza non venga verificata, viene lanciata un'eccezione e viene mostrato a video un messaggio di errore.

La vista inerente alla gestione delle opere d'arte di una certa esposizione è controllata dall'interfaccia IControllerManageArtworkExhibit e dalla sua implementazione ControllerManageArtworkExhibit. Tale controller gestisce gli eventi derivanti dalla pressione dei pulsanti, come ad esempio permettere l'aggiunta di un'opera d'arte solo se non è già presente nell'elenco delle opere dell'esposizione. Nel caso in cui la correttezza non venga verificata, viene lanciata un'eccezione e viene mostrato a video un messaggio di errore.

La vista inerente la biglietteria è controllata dall'interfaccia ITicketOfficeController e dalla sua implementazione TicketOfficeController. Tale controller, oltre a gestire le azioni derivanti dalla pressione dei pulsanti, svolge i compiti di salvataggio dei dati, caricamento delle opportune esposizioni in tabella e controllo sulla data corrente in modo da consentire il reset del numero di biglietti giornalieri. Vengono inoltre effettuati i controlli sulla correttezza dei dati inseriti, visualizzando un messaggio di errore in caso contrario.

La vista inerente il bilancio è controllata dall'interfaccia IBalanceController e dalla sua implementazione BalanceController. Tale controller gestisce le azioni derivanti dalla pressione dei pulsanti, consente il salvataggio dei dati e carica le opportune esposizioni ed il loro saldo in tabella. Anche in questo caso vengono effettuati i controlli sulla correttezza dei dati inseriti, visualizzando un messaggio di errore in caso contrario.

La vista inerente la classifica è controllata dall'interfaccia IClassificationController e dalla relativa implementazione ClassificationController. Tale controller si occupa di creare la struttura adeguata per la visualizzazione delle esposizioni in ordine e procede al relativo caricamento in tabella. Essendo presente il solo pulsante "HOME", è necessario solamente gestirne la relativa pressione.

La rappresentazione UML della vista insieme al controller viene mostrata nelle figure 5, 6 e 7.

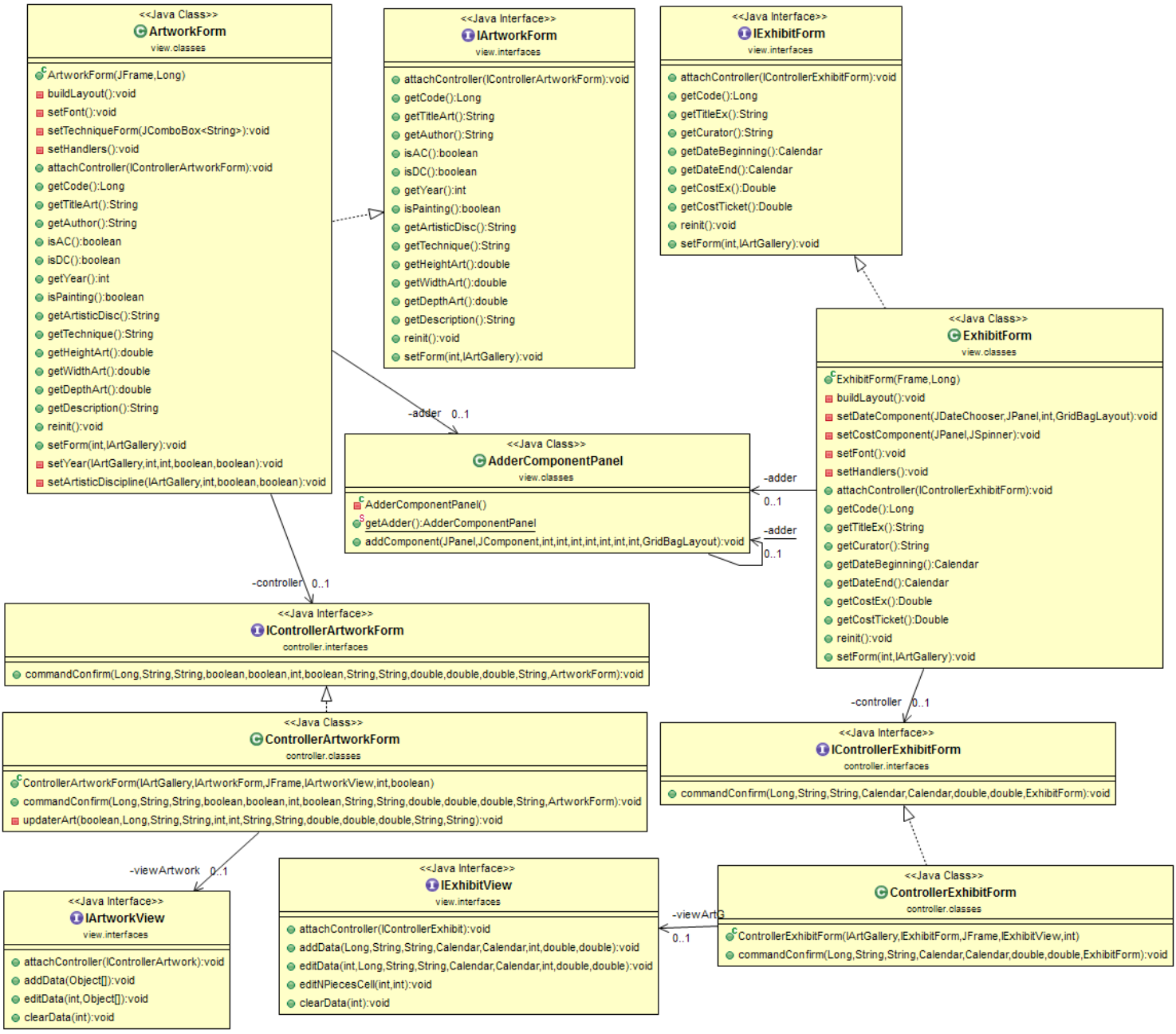

Figura 5: Diagramma UML relativo alla vista e al controller - parte 1

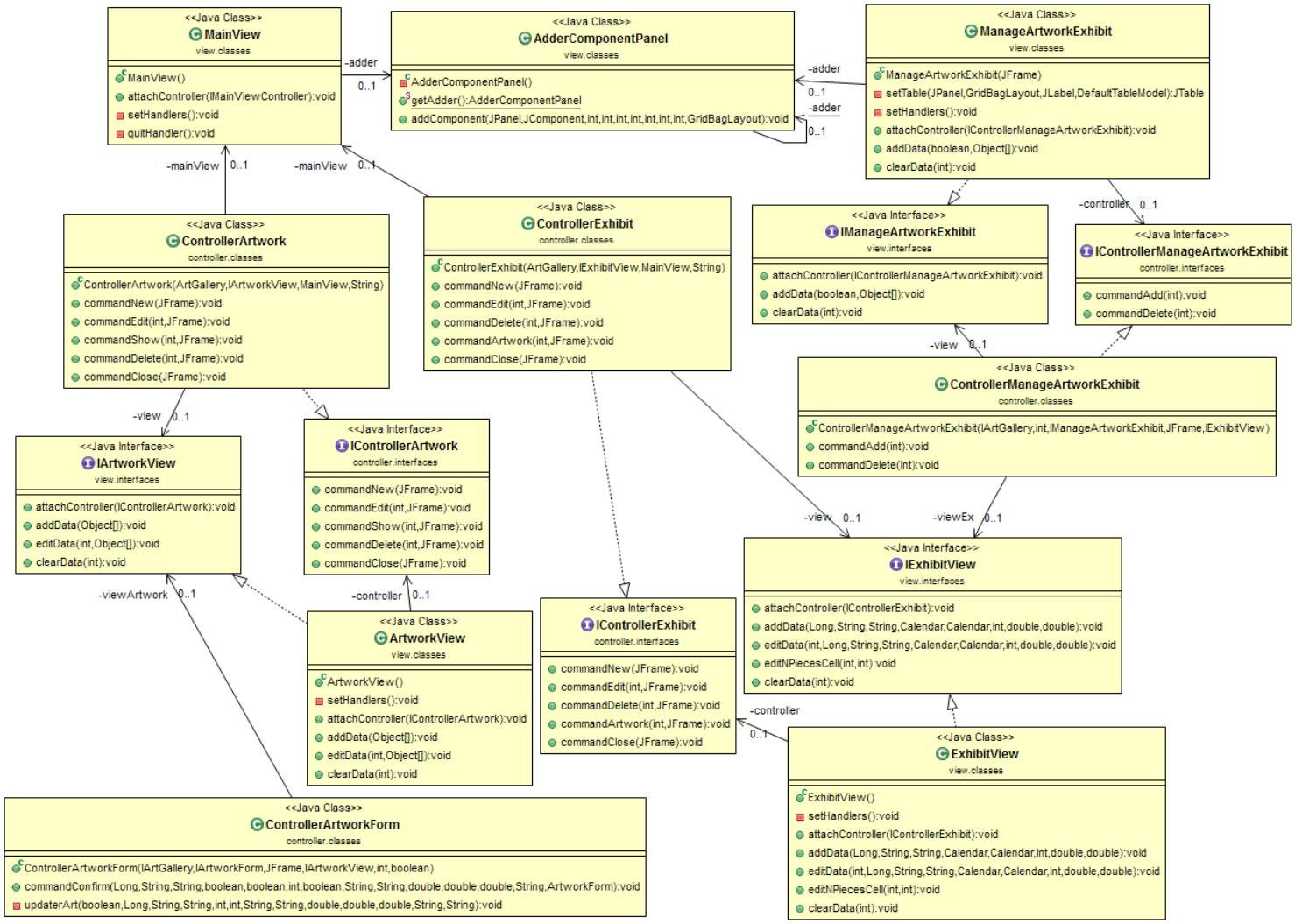

Figura 6: Diagramma UML relativo alla vista e al controller - parte 2

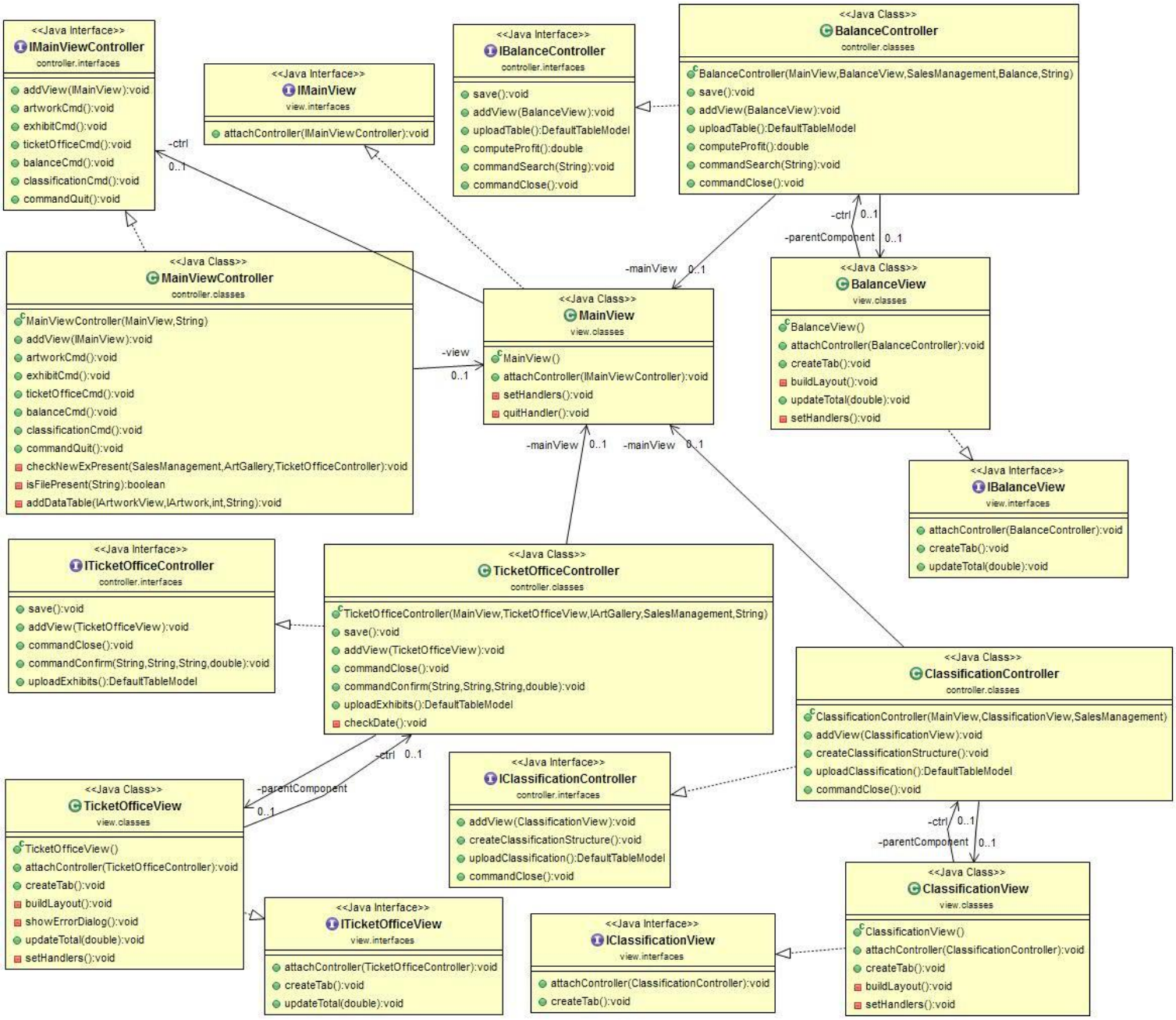

Figura 7: Diagramma UML relativo alla vista e al controller - parte 3

Per gestire al meglio le eccezioni lanciate dall'applicativo, si è deciso di implementate anche due nuove eccezioni quali:

- DatingNotExistException: generata quando si tenta di registrare l'anno di un'opera d'arte con la datazione sbagliata, come ad esempio registrare l'anno 0 nella datazione "Avanti Cristo" (A.C.) oppure un anno superiore a quello corrente come "Dopo Cristo" (D.C.);
- IllegalOperationException: generata quando si tenta di svolgere una certa operazione che risulta essere illegale in quel momento. Ne sono un esempio l'aggiunta di un'opera d'arte ad una esposizione, quando questa è già stata aggiunta, oppure l'eliminazione di un'opera d'arte che è utilizzata in una esposizione, oppure l'aggiunta o l'eliminazione di un'opera d'arte da una esposizione che è già finita, oppure la modifica o l'eliminazione di una esposizione che è già iniziata.

Capitolo 3: Sviluppo

3.1 Divisione dei compiti e metodologia di lavoro

Il progetto è stato sviluppato da un team di due persone.

La parte attinente la gestione dell'archivio delle opere d'arte e l'archivio delle esposizioni è stata svolta da Elisa Casadio. Inizialmente era stato pensato di implementare questa parte con una lista delle esposizioni, le quali salvavano al loro interno una lista delle opere d'arte che venivano utilizzate. Dato che una stessa opera d'arte poteva essere inserita in più esposizioni, si è deciso di implementare due liste, una per le opere e una per le esposizioni. In seguito per gestire al meglio la presenza o meno di un'opera d'arte in una mostra è stato deciso di aggiungere un codice alle opere d'arte e per migliorare i controlli sulle esposizioni, è stato aggiunto un codice anche ad esse.

La parte attinente la gestione della biglietteria, e quindi gli acquisti dei biglietti ridotti e non, la gestione dei bilanci attuale e passati ed il calcolo della classifica delle esposizioni è stata svolta da Sofia Rosetti.

La realizzazione delle suddette parti è stata fatta in modo tale che la parte riguardante biglietteria, bilancio e classifica potesse usufruire dei dati relativi alle esposizioni, e quindi dell'interfaccia IExhibit e della sua implementazione Exhibit.

Entrambi i componenti hanno collaborato nella creazione delle interfacce IMainView e IMainViewController, unitamente alla loro implementazione tramite le classi MainView, MainViewController e Main. Le suddette classi hanno il compito di avviare l'applicativo e consentire l'utilizzo del sistema attraverso una "home" dalla quale è possibile navigare attraverso le varie funzionalità.

La dettagliata separazione delle parti ha permesso un utilizzo del DVCS senza particolari problemi inerenti lo sviluppo contemporaneo del medesimo codice. In alcuni casi ha però creato qualche problema su "Eclipse", obbligando i membri a fare due volte le stesse registrazioni delle modifiche (commit).

3.2 Note di sviluppo

All'interno di questo progetto è stata utilizzata una libreria esterna: jcalendar-1.4.jar.
Essa è stata utilizzata nel form delle esposizioni, per avere una migliore visualizzazione del calendario, così da facilitare la scelta della data di inizio e di fine dell'esposizione all'utente.

Capitolo 4: Commenti finali

4.1 Conclusioni e commenti futuri

Entrambi i componenti del gruppo hanno lavorato per un totale di circa 90 ore a testa.

L'applicativo che è stato realizzato si presenta con una piacevole e chiara interfaccia grafica, che consente all'utente una gestione facile, veloce ed intuitiva.

È stato sviluppato anche un aspetto meno rilevante per quanto riguarda la gestione principale di una galleria d'arte, quale la classifica delle esposizioni, in quanto tale aspetto è per lo più di carattere informativo generale. È stata bensì posta una certa attenzione verso la memorizzazione dei bilanci anche degli anni precedenti, in modo tale da poter consentire una più completa analisi della situazione economica della galleria, e verso la memorizzazione delle opere d'arte e delle esposizioni, in modo tale da avere sempre a portata di mano uno storico.

In futuro, se si vorrà, l'applicativo potrà essere ampliato dando la possibilità all'utente di mettere nella scheda di ogni opera d'arte la relativa fotografia, di aggiungere nella scheda di ogni esposizione la sua locandina e di gestire la posizione delle opere d'arte di una certa esposizione all'interno della galleria, attraverso una rappresentazione grafica delle stanze e imponendo quindi un limite al numero di opere d'arte da esporre.