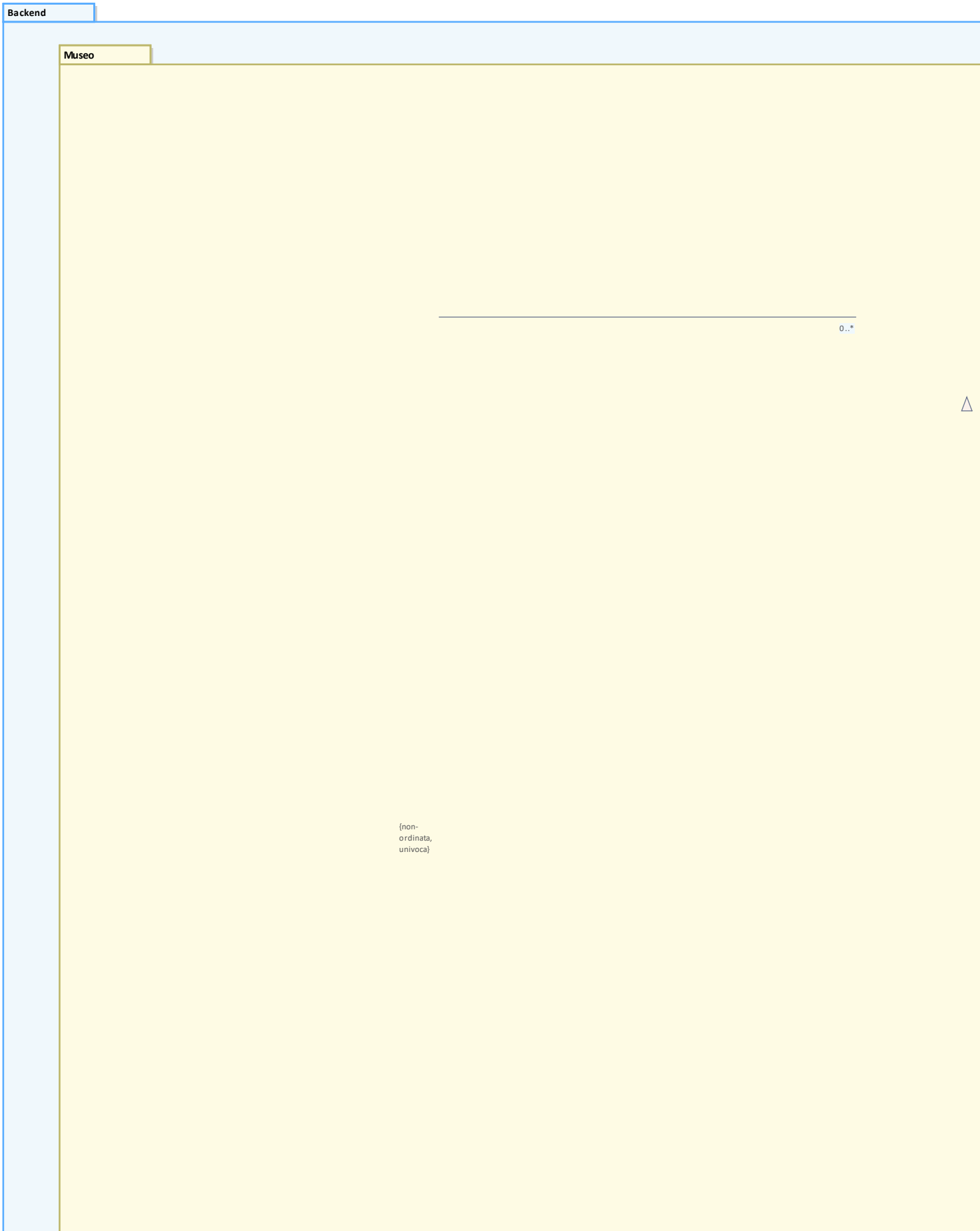
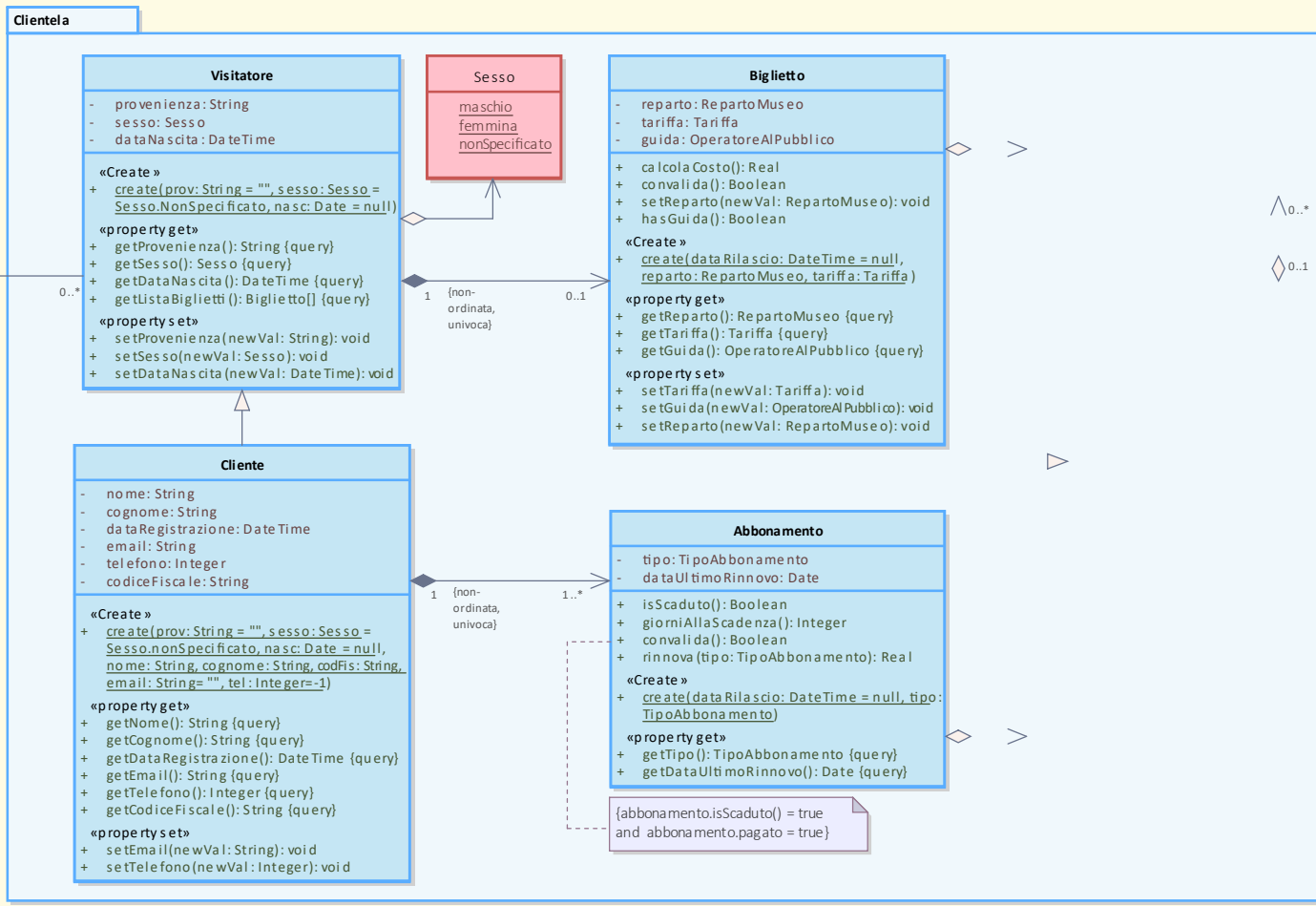

Backend
Museo
0..*
{non-
ordinata,
univoca}




*(from Diagamma Delle Classi)*

{non-ordinata, univoca}

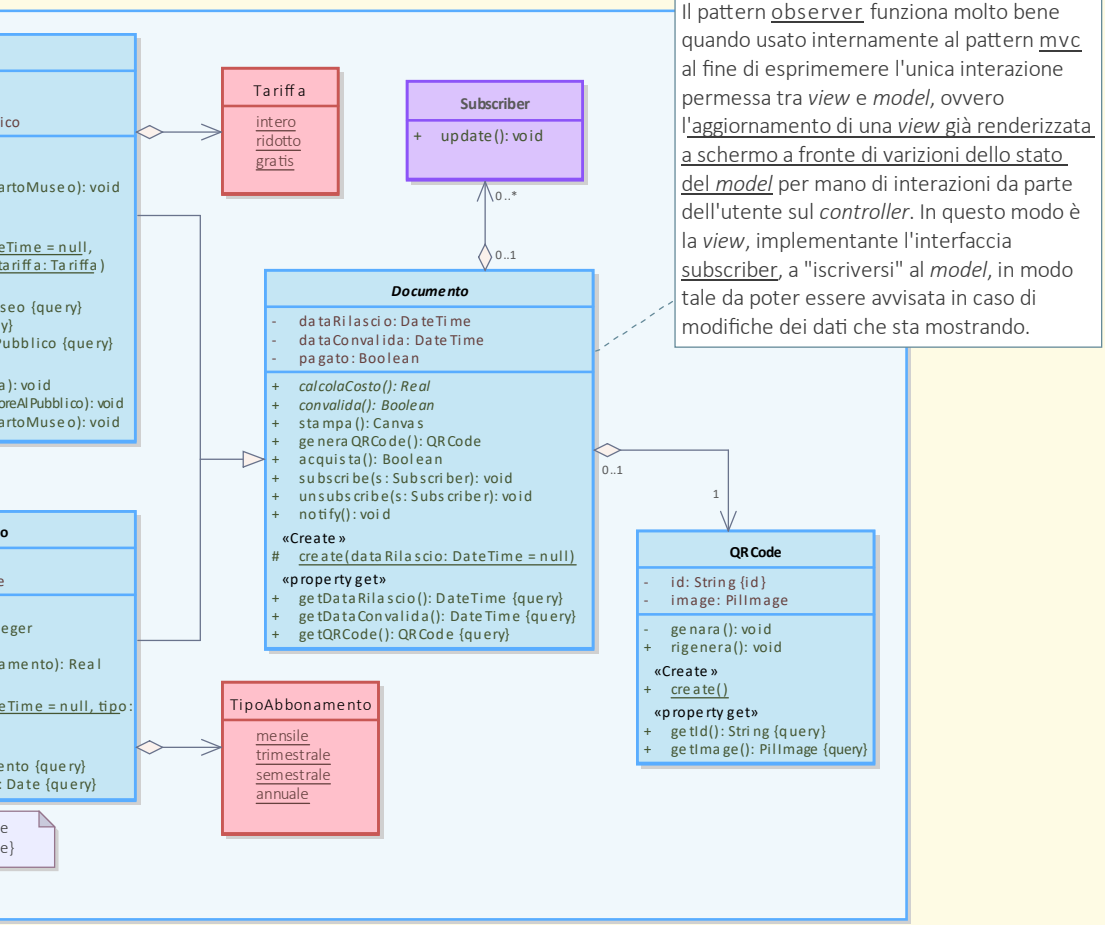
0..*

{non-ordinata, univoca}

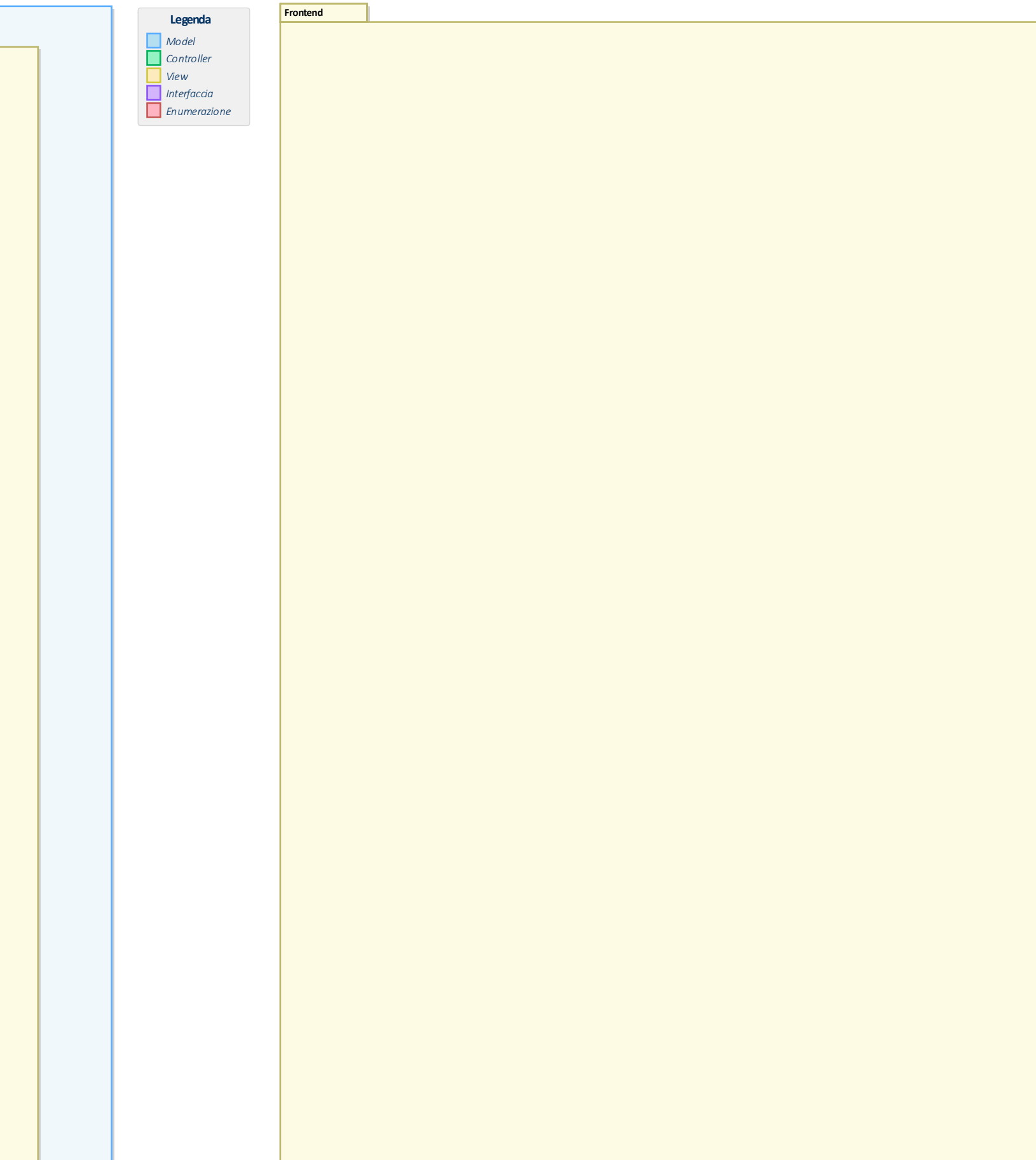
1

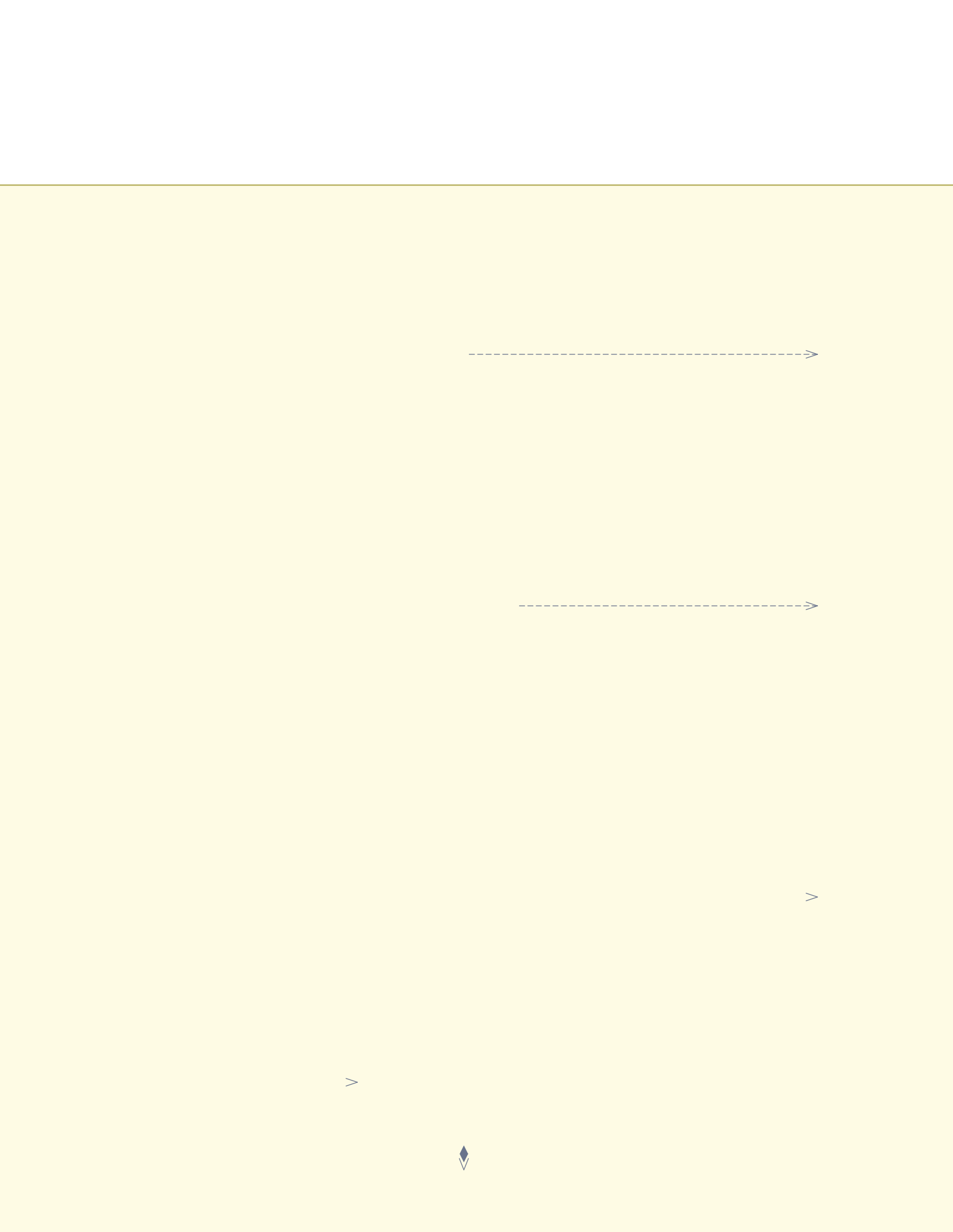
0..1

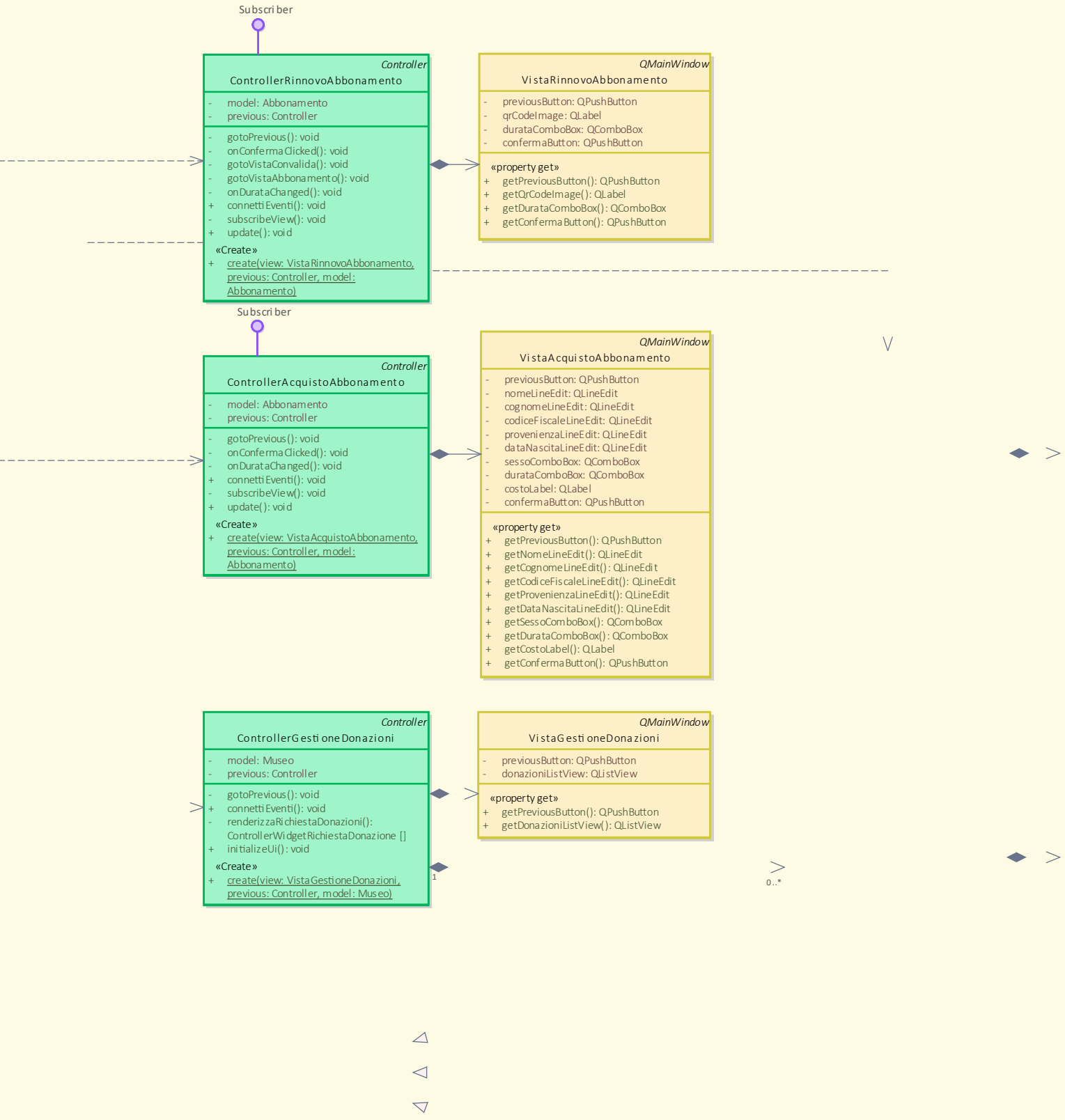
0..*

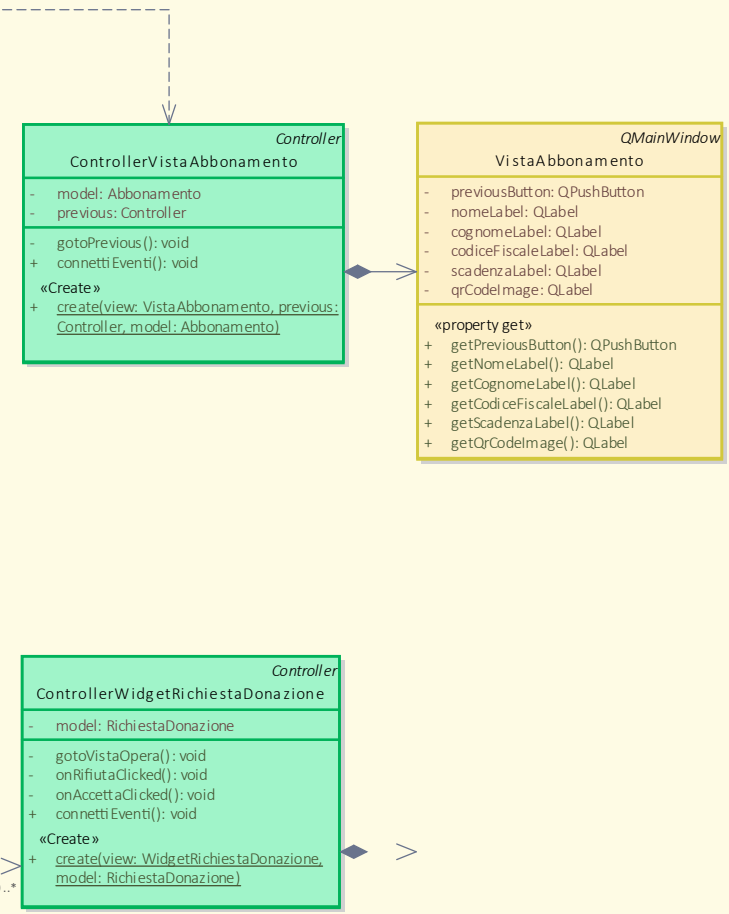



*gamma Delle Classi)*

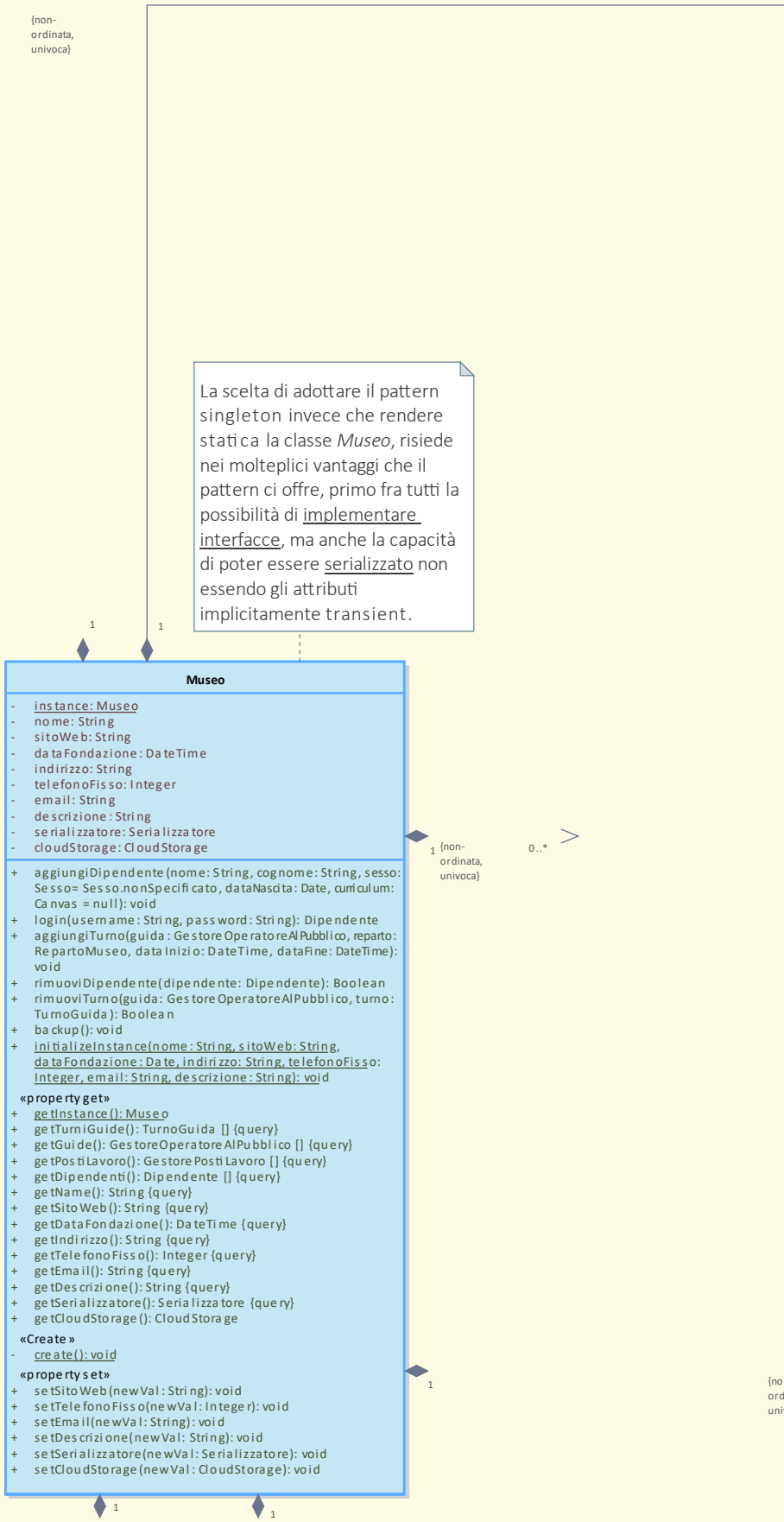

Legenda
Model
Controller
View
Interfaccia
Enumerazione
Frontend

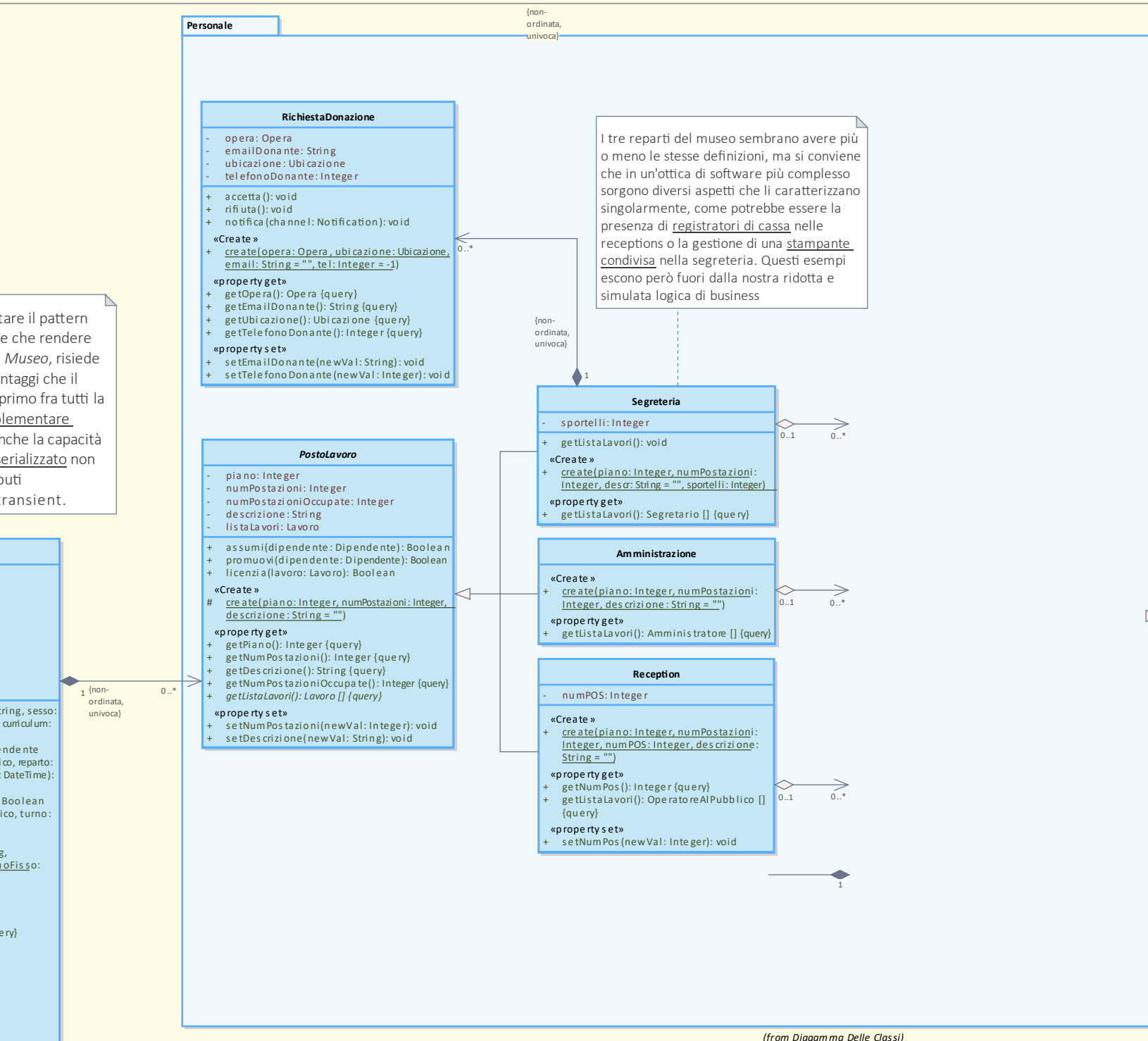


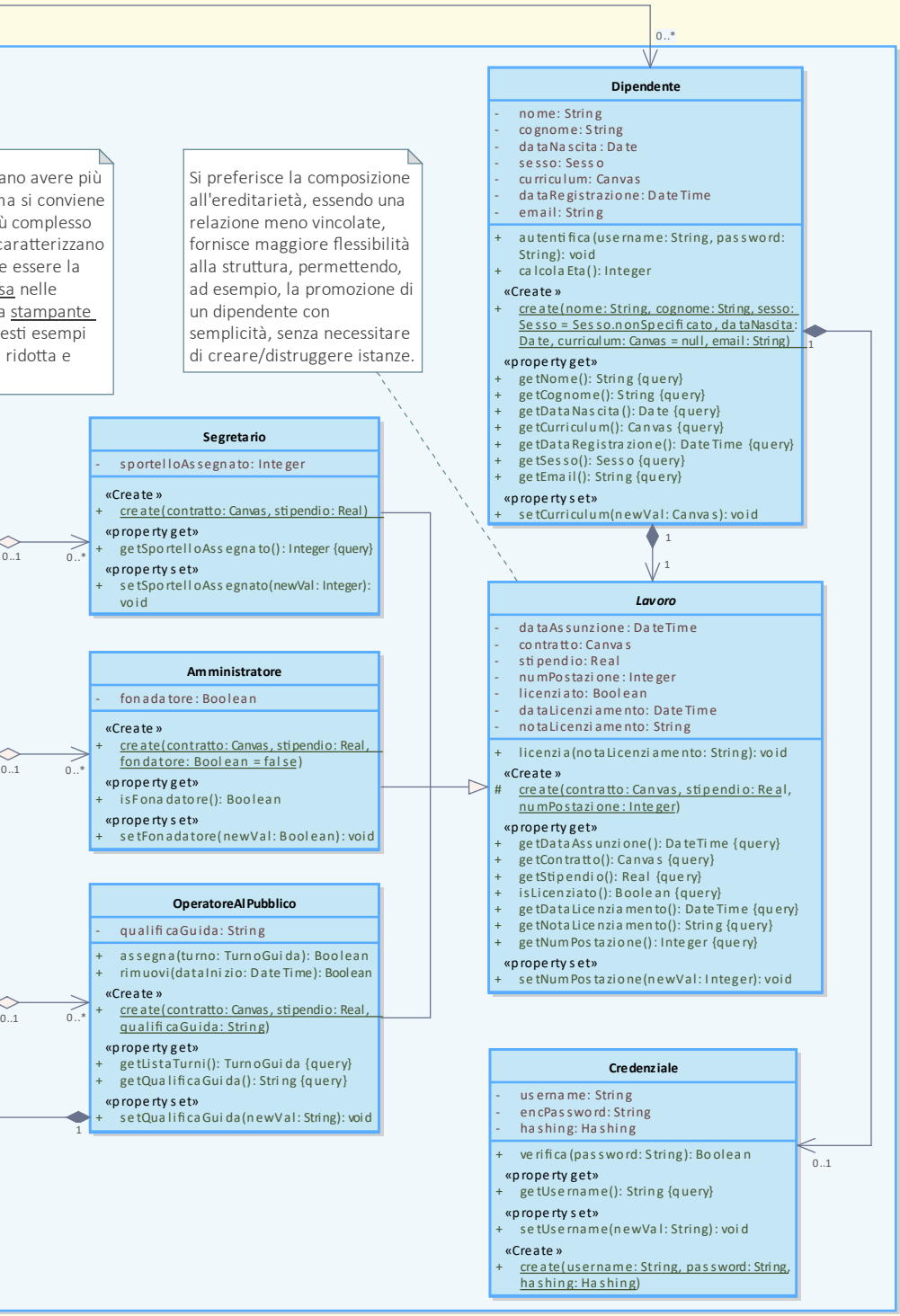

Subscriber
Controller
ControllerRinnovoAbbonamento
- model: Abbonamento
- previous: Controller
- gotoPrevious(): void
- onConfermaClicked(): void
- gotoVistaConvalida(): void
- gotoVistaAbbonamento(): void
- onDurataChanged(): void
+ connettiEventi(): void
- subscribeView(): void
+ update(): void
«Create»
+ create(view: VistaRinnovoAbbonamento, previous: Controller, model: Abbonamento)
QMainWindow
VistaRinnovoAbbonamento
- previousButton: QPushButton
- qrCodeImage: QLabel
- durataComboBox: QComboBox
- confermaButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getQrCodeImage(): QLabel
+ getDurataComboBox(): QComboBox
+ getConfermaButton(): QPushButton
Subscriber
Controller
ControllerAcquistoAbbonamento
- model: Abbonamento
- previous: Controller
- gotoPrevious(): void
- onConfermaClicked(): void
- onDurataChanged(): void
+ connettiEventi(): void
- subscribeView(): void
+ update(): void
«Create»
+ create(view: VistaAcquistoAbbonamento, previous: Controller, model: Abbonamento)
QMainWindow
VistaAcquistoAbbonamento
- previousButton: QPushButton
- nomeLineEdit: QLineEdit
- cognomeLineEdit: QLineEdit
- codiceFiscaleLineEdit: QLineEdit
- provenienzaLineEdit: QLineEdit
- dataNascitaLineEdit: QLineEdit
- sessoComboBox: QComboBox
- durataComboBox: QComboBox
- costoLabel: QLabel
- confermaButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getNomeLineEdit(): QLineEdit
+ getCognomeLineEdit(): QLineEdit
+ getCodiceFiscaleLineEdit(): QLineEdit
+ getProvenienzaLineEdit(): QLineEdit
+ getDataNascitaLineEdit(): QLineEdit
+ getSessoComboBox(): QComboBox
+ getDurataComboBox(): QComboBox
+ getCostoLabel(): QLabel
+ getConfermaButton(): QPushButton
Controller
ControllerGestioneDonazioni
- model: Museo
- previous: Controller
- gotoPrevious(): void
+ connettiEventi(): void
- renderizzaRichiestaDonazioni(): ControllerWidgetRichiestaDonazione []
+ initializeUi(): void
«Create»
+ create(view: VistaGestioneDonazioni, previous: Controller, model: Museo)
QMainWindow
VistaGestioneDonazioni
- previousButton: QPushButton
- donazioniListView: QListView
«property get»
+ getPreviousButton(): QPushButton
+ getDonazioniListView(): QListView
1
0..*


Controller
ControllerVistaAbbonamento
- model: Abbonamento
- previous: Controller
- gotoPrevious(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaAbbonamento, previous: Controller, model: Abbonamento)
QMainWindow
VistaAbbonamento
- previousButton: QPushButton
- nomeLabel: QLabel
- cognomeLabel: QLabel
- codiceFiscaleLabel: QLabel
- scadenzaLabel: QLabel
- qrCodeImage: QLabel
«property get»
+ getPreviousButton(): QPushButton
+ getNomeLabel(): QLabel
+ getCognomeLabel(): QLabel
+ getCodiceFiscaleLabel(): QLabel
+ getScadenzaLabel(): QLabel
+ getQrCodeImage(): QLabel

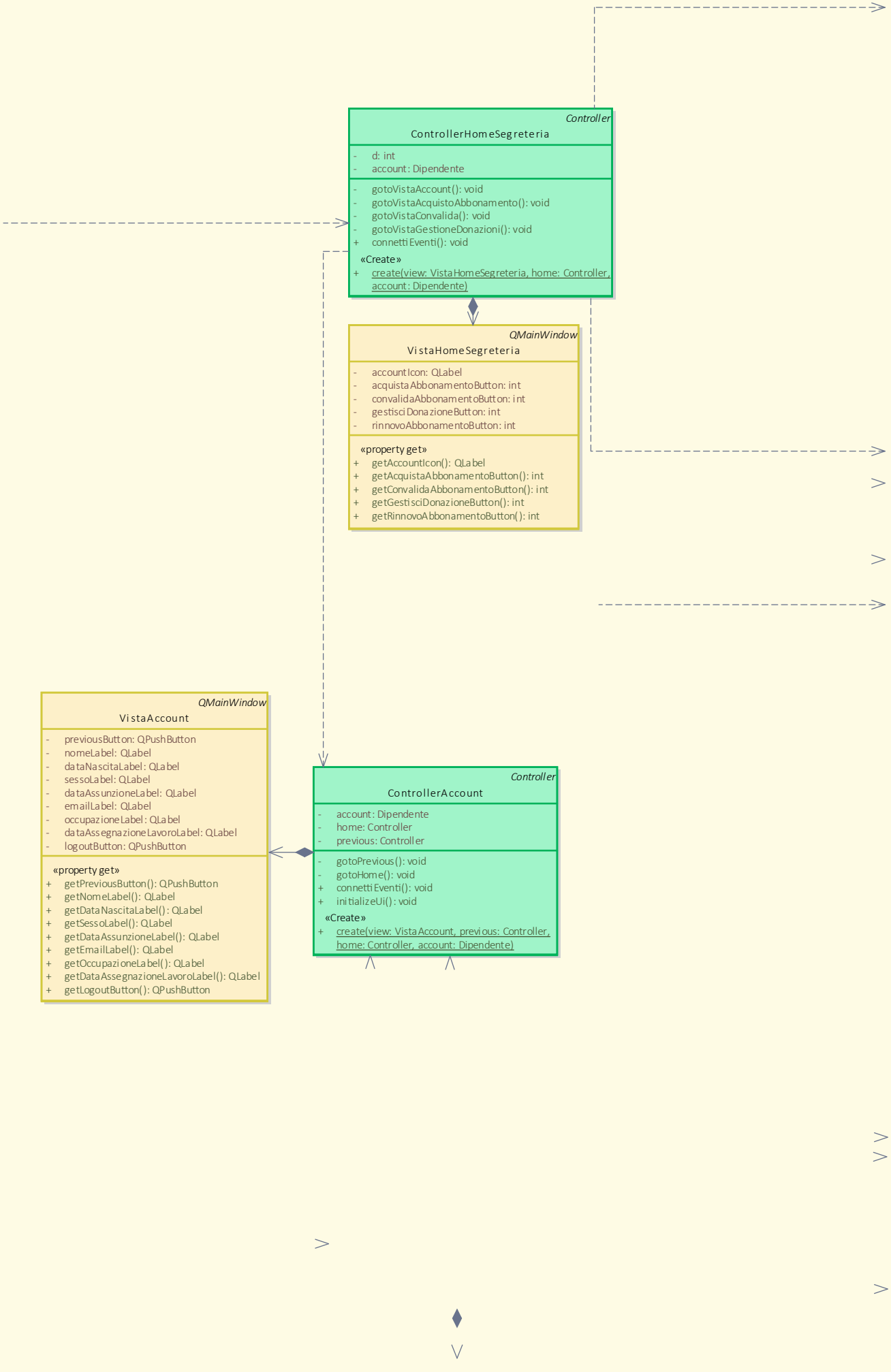

Controller
ControllerWidgetRichiestaDonazione
- model: RichiestaDonazione
- gotoVistaOpera(): void
- onRifiutaClicked(): void
- onAccettaClicked(): void
+ connettiEventi(): void
«Create»
+ create(view: WidgetRichiestaDonazione, model: RichiestaDonazione)

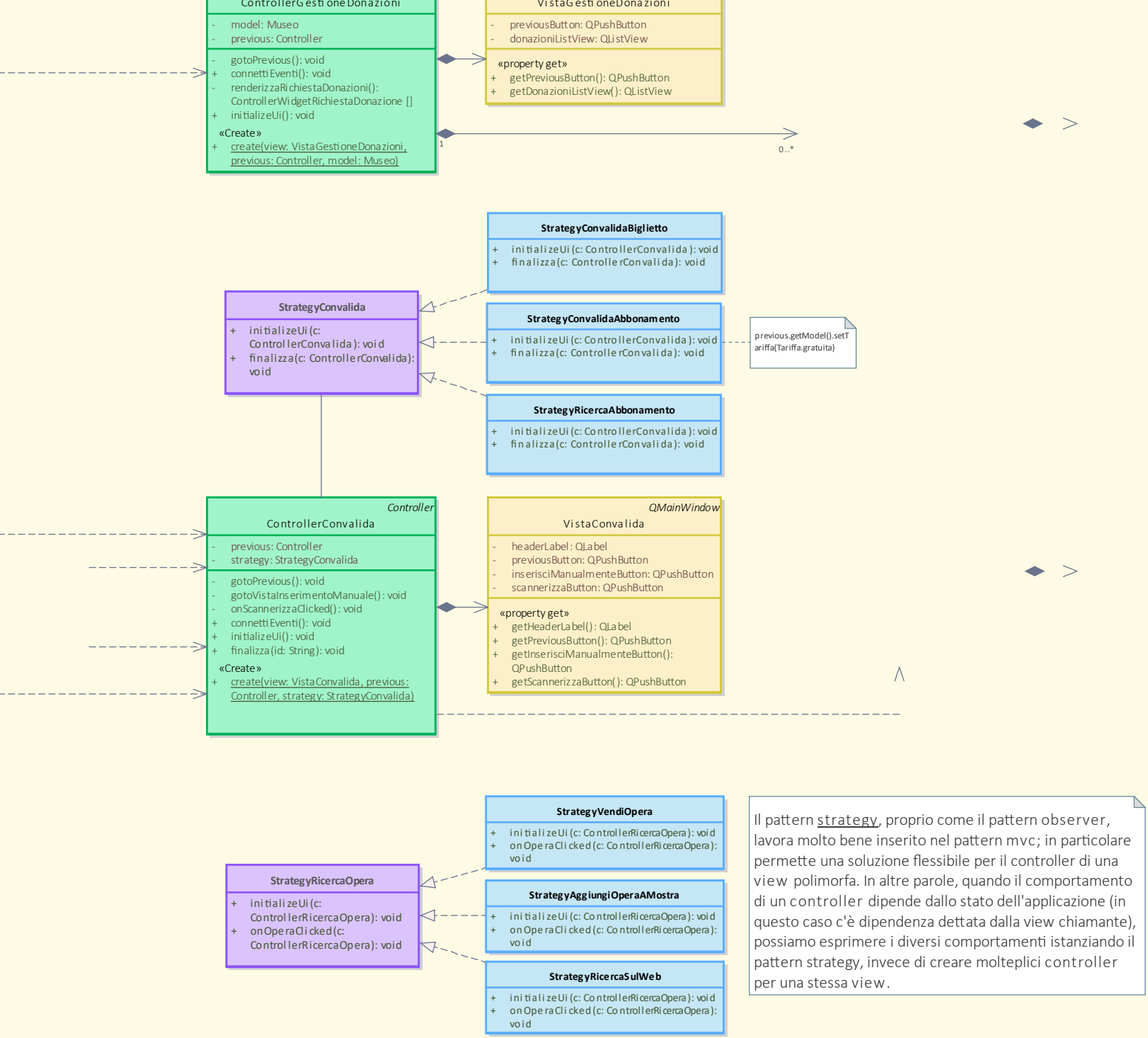

{non-ordinata, univoca}
La scelta di adottare il pattern singleton invece che rendere statica la classe *Museo*, risiede nei molteplici vantaggi che il pattern ci offre, primo fra tutti la possibilità di implementare interfacce, ma anche la capacità di poter essere serializzato non essendo gli attributi implicitamente transient.
1
1
Museo
- instance: Museo
- nome: String
- sitoWeb: String
- dataFondazione: DateTime
- indirizzo: String
- telefonoFisso: Integer
- email: String
- descrizione: String
- serializzatore: Serializzatore
- cloudStorage: CloudStorage
+ aggiungiDipendente(nome: String, cognome: String, sesso: Sesso= Sesso.nonSpecificato, dataNascita: Date, curriculum: Canvas = null): void
+ login(username: String, password: String): Dipendente
+ aggiungiTurno(guida: GestoreOperatoreAlPubblico, reparto: RepartoMuseo, dataInizio: DateTime, dataFine: DateTime): void
+ rimuoviDipendente(dipendente: Dipendente): Boolean
+ rimuoviTurno(guida: GestoreOperatoreAlPubblico, turno: TurnoGuida): Boolean
+ backup(): void
+ initializeInstance(nome: String, sitoWeb: String, dataFondazione: Date, indirizzo: String, telefonoFisso: Integer, email: String, descrizione: String): void
«property get»
+ getInstance(): Museo
+ getTurniGuide(): TurnoGuida [] {query}
+ getGuide(): GestoreOperatoreAlPubblico [] {query}
+ getPostiLavoro(): GestorePostiLavoro [] {query}
+ getDipendenti(): Dipendente [] {query}
+ getName(): String {query}
+ getSitoWeb(): String {query}
+ getDataFondazione(): DateTime {query}
+ getIndirizzo(): String {query}
+ getTelefonoFisso(): Integer {query}
+ getEmail(): String {query}
+ getDescrizione(): String {query}
+ getSerializzatore(): Serializzatore {query}
+ getCloudStorage(): CloudStorage
«Create»
- create(): void
«property set»
+ setSitoWeb(newVal: String): void
+ setTelefonoFisso(newVal: Integer): void
+ setEmail(newVal: String): void
+ setDescrizione(newVal: String): void
+ setSerializzatore(newVal: Serializzatore): void
+ setCloudStorage(newVal: CloudStorage): void
1
{non-ordinata, univoca}
0..*
1
1
1
{non-ordinata, univoca}


Personale
{non-ordinata, univoca}
RichiestaDonazione
- opera: Opera
- emailDonante: String
- ubicazione: Ubicazione
- telefonoDonante: Integer
+ accetta(): void
+ rifiuta(): void
+ notifica(channel: Notification): void
«Create»
+ create(opera: Opera, ubicazione: Ubicazione, email: String = "", tel: Integer = -1)
«property get»
+ getOpera(): Opera {query}
+ getEmailDonante(): String {query}
+ getUbicazione(): Ubicazione {query}
+ getTelefonoDonante(): Integer {query}
«property set»
+ setEmailDonante(newVal: String): void
+ setTelefonoDonante(newVal: Integer): void
0..*
{non-ordinata, univoca}
1
I tre reparti del museo sembrano avere più o meno le stesse definizioni, ma si conviene che in un'ottica di software più complesso sorgono diversi aspetti che li caratterizzano singolarmente, come potrebbe essere la presenza di registratori di cassa nelle receptions o la gestione di una stampante condivisa nella segreteria. Questi esempi escono però fuori dalla nostra ridotta e simulata logica di business
Segreteria
- sportelli: Integer
+ getListaLavori(): void
«Create»
+ create(piano: Integer, numPostazioni: Integer, descr: String = "", sportelli: Integer)
«property get»
+ getListaLavori(): Segretario [] {query}
0..1
0..*
PostoLavoro
- piano: Integer
- numPostazioni: Integer
- numPostazioniOccupate: Integer
- descrizione: String
- listaLavori: Lavoro
+ assumi(dipendente: Dipendente): Boolean
+ promuovi(dipendente: Dipendente): Boolean
+ licenzia(lavoro: Lavoro): Boolean
«Create»
# create(piano: Integer, numPostazioni: Integer, descrizione: String = "")
«property get»
+ getPiano(): Integer {query}
+ getNumPostazioni(): Integer {query}
+ getDescrizione(): String {query}
+ getNumPostazioniOccupate(): Integer {query}
+ getListaLavori(): Lavoro [] {query}
«property set»
+ setNumPostazioni(newVal: Integer): void
+ setDescrizione(newVal: String): void
Amministrazione
«Create»
+ create(piano: Integer, numPostazioni: Integer, descrizione: String = "")
«property get»
+ getListaLavori(): Amministratore [] {query}
0..1
0..*
Reception
- numPOS: Integer
«Create»
+ create(piano: Integer, numPostazioni: Integer, numPOS: Integer, descrizione: String = "")
«property get»
+ getNumPos(): Integer {query}
+ getListaLavori(): OperatoreAlPubblico [] {query}
«property set»
+ setNumPos(newVal: Integer): void
0..1
0..*
1
1
{non-ordinata, univoca}
0..*
(from Diagramma Delle Classi)
tare il pattern
e che rendere
Museo, risiede
ntaggi che il
primo fra tutti la
lementare
nche la capacità
serializzato non
uti
transient.
ring, sesso:
curriculum:
ndente
ico, reparto:
DateTime):
Boolean
ico, turno:
g,
oFisso:
ery}
1
{non-ordinata, univoca}
1
0..1
0..*
0..*

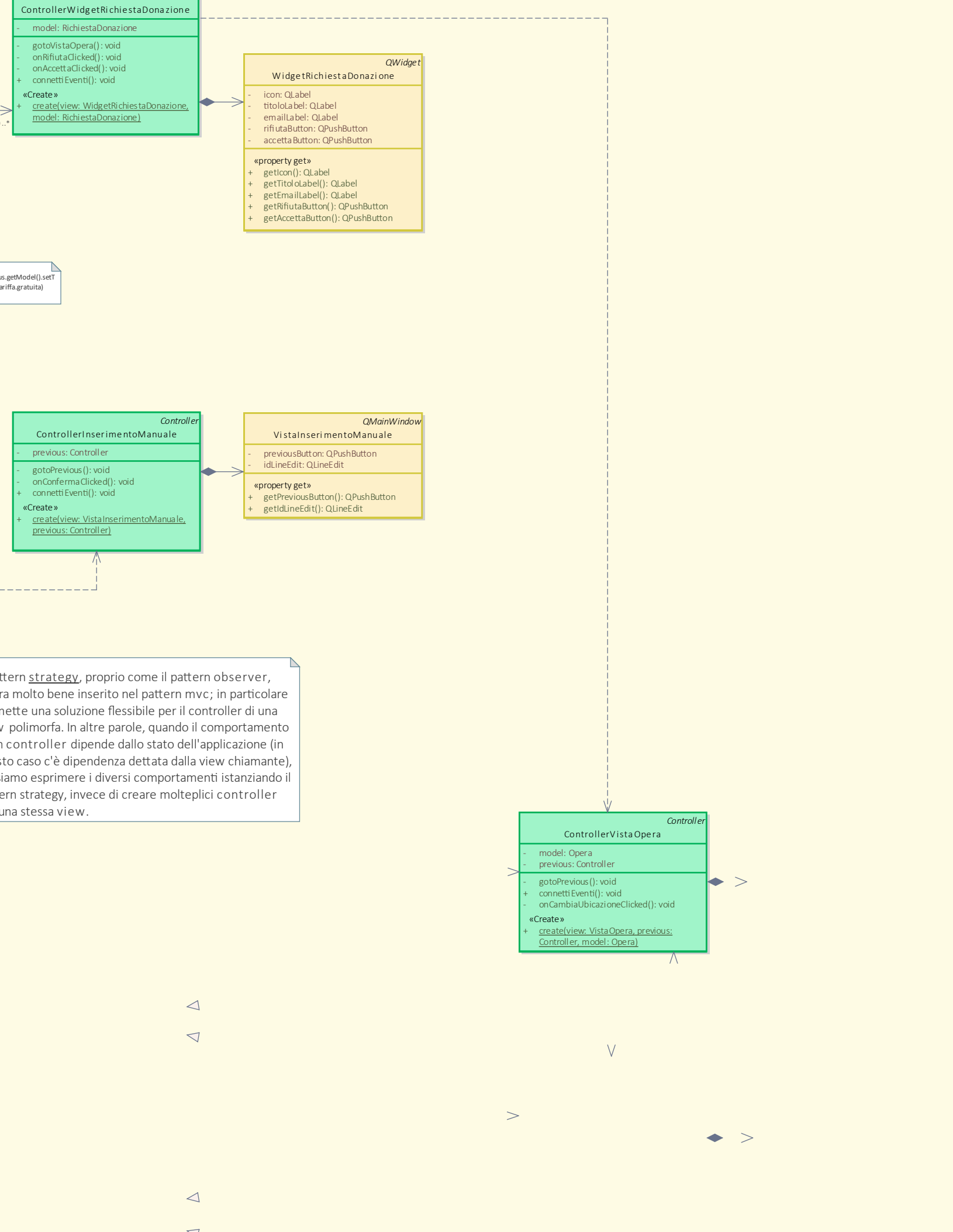


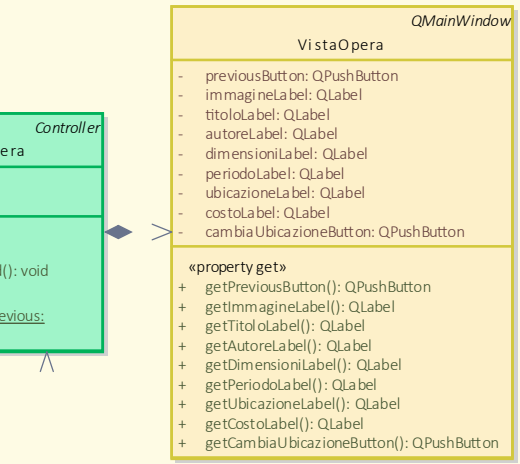

*m Diagamma Delle Classi)*

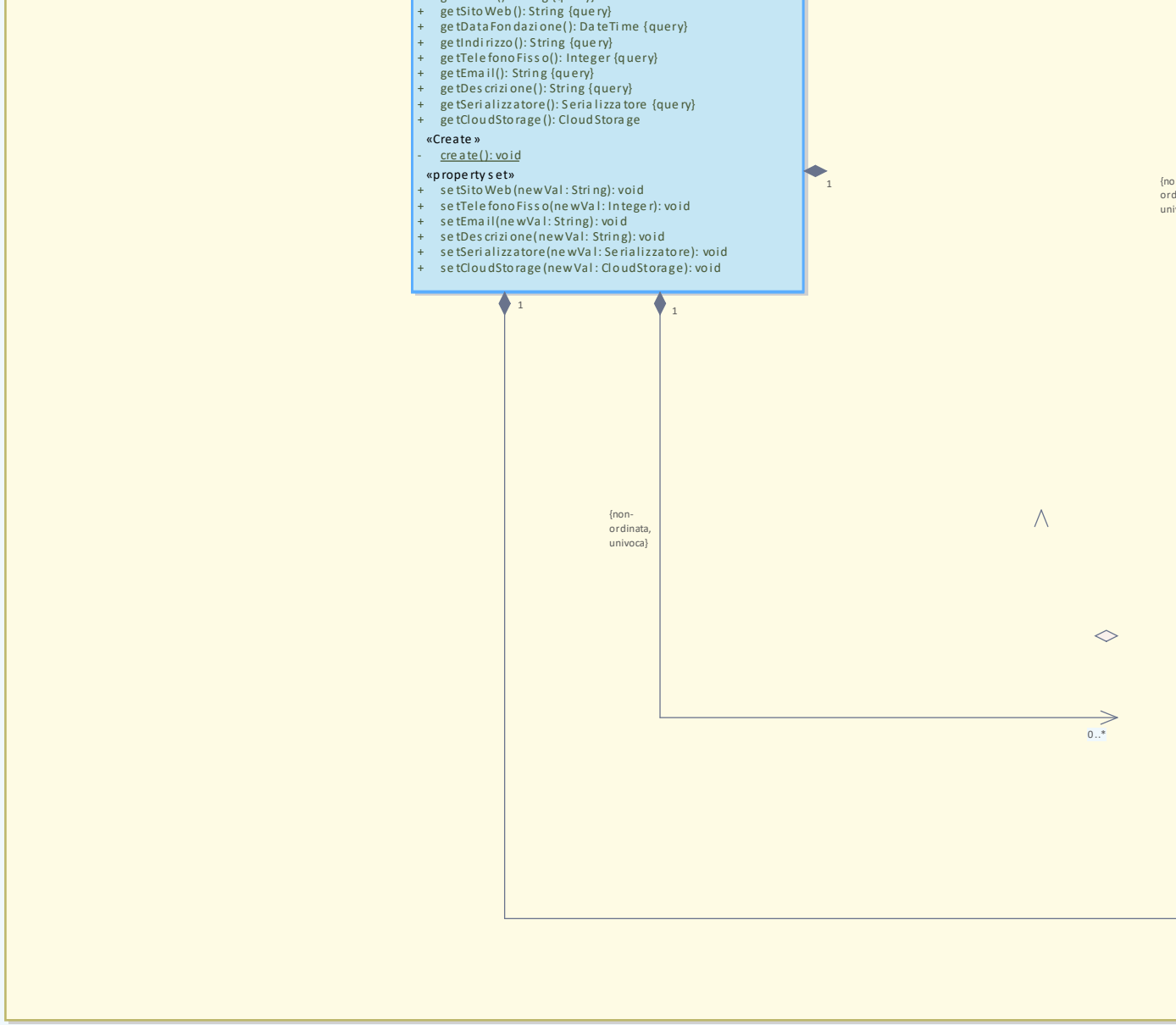

Controller
ControllerHomeSegreteria
- d: int
- account: Dipendente
- gotoVistaAccount(): void
- gotoVistaAcquistoAbbonamento(): void
- gotoVistaConvalida(): void
- gotoVistaGestioneDonazioni(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaHomeSegreteria, home: Controller, account: Dipendente)
QMainWindow
VistaHomeSegreteria
- accountIcon: QLabel
- acquistaAbbonamentoButton: int
- convalidaAbbonamentoButton: int
- gestisciDonazioneButton: int
- rinnovoAbbonamentoButton: int
«property get»
+ getAccountIcon(): QLabel
+ getAcquistaAbbonamentoButton(): int
+ getConvalidaAbbonamentoButton(): int
+ getGestisciDonazioneButton(): int
+ getRinnovoAbbonamentoButton(): int
QMainWindow
VistaAccount
- previousButton: QPushButton
- nomeLabel: QLabel
- dataNascitaLabel: QLabel
- sessoLabel: QLabel
- dataAssunzioneLabel: QLabel
- emailLabel: QLabel
- occupazioneLabel: QLabel
- dataAssegnazioneLavoroLabel: QLabel
- logoutButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getNomeLabel(): QLabel
+ getDataNascitaLabel(): QLabel
+ getSessoLabel(): QLabel
+ getDataAssunzioneLabel(): QLabel
+ getEmailLabel(): QLabel
+ getOccupazioneLabel(): QLabel
+ getDataAssegnazioneLavoroLabel(): QLabel
+ getLogoutButton(): QPushButton
Controller
ControllerAccount
- account: Dipendente
- home: Controller
- previous: Controller
- gotoPrevious(): void
- gotoHome(): void
+ connettiEventi(): void
+ initializeUi(): void
«Create»
+ create(view: VistaAccount, previous: Controller, home: Controller, account: Dipendente)

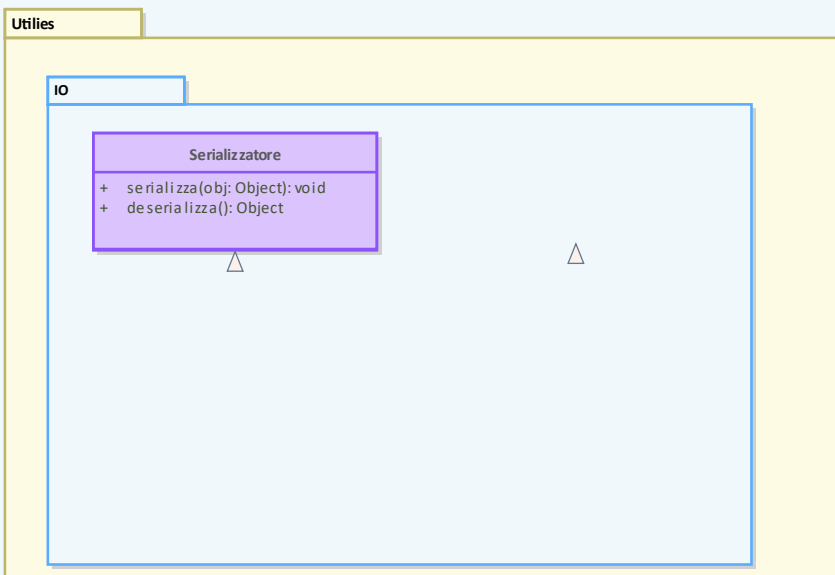



Il pattern strategy, proprio come il pattern observer, lavora molto bene inserito nel pattern mvc; in particolare permette una soluzione flessibile per il controller di una view polimorfa. In altre parole, quando il comportamento di un controller dipende dallo stato dell'applicazione (in questo caso c'è dipendenza dettata dalla view chiamante), possiamo esprimere i diversi comportamenti istanziando il pattern strategy, invece di creare molteplici controller per una stessa view.

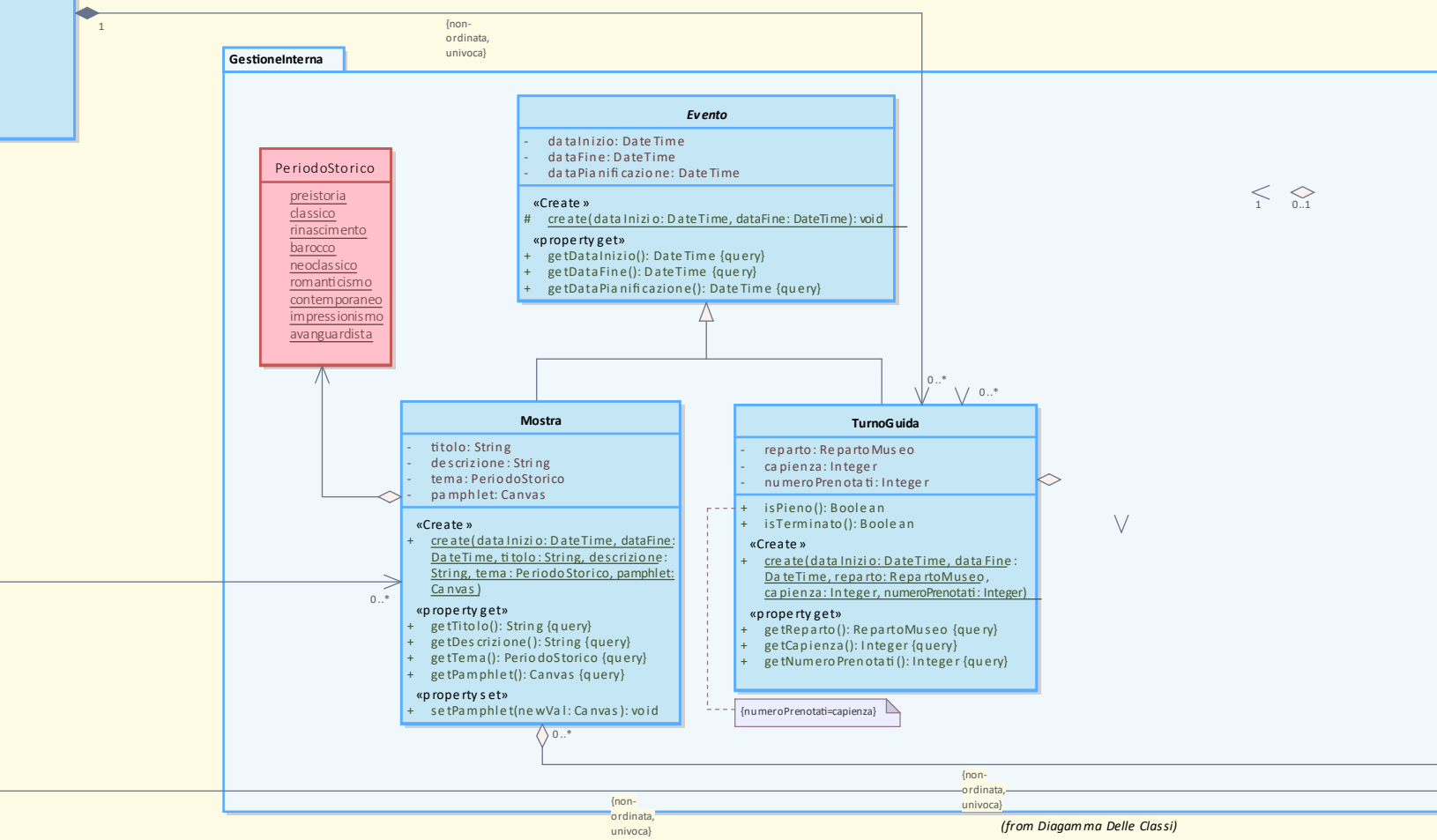

ControllerWidgetRichiestaDonazione
- model: RichiestaDonazione
- gotoVistaOpera(): void
- onRifiutaClicked(): void
- onAccettaClicked(): void
+ connettiEventi(): void
«Create»
+ create(view: WidgetRichiestaDonazione, model: RichiestaDonazione)
QWidget
WidgetRichiestaDonazione
- icon: QLabel
- titoloLabel: QLabel
- emailLabel: QLabel
- rifiutaButton: QPushButton
- accettaButton: QPushButton
«property get»
+ getIcon(): QLabel
+ getTitoloLabel(): QLabel
+ getEmailLabel(): QLabel
+ getRifiutaButton(): QPushButton
+ getAccettaButton(): QPushButton
Controller
ControllerInserimentoManuale
- previous: Controller
- gotoPrevious(): void
- onConfermaClicked(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaInserimentoManuale, previous: Controller)
QMainWindow
VistaInserimentoManuale
- previousButton: QPushButton
- idLineEdit: QLineEdit
«property get»
+ getPreviousButton(): QPushButton
+ getIdLineEdit(): QLineEdit
ttern strategy, proprio come il pattern observer,
ra molto bene inserito nel pattern mvc; in particolare
nette una soluzione flessibile per il controller di una
v polimorfa. In altre parole, quando il comportamento
n controller dipende dallo stato dell'applicazione (in
sto caso c'è dipendenza dettata dalla view chiamante),
iamo esprimere i diversi comportamenti istanziando il
ern strategy, invece di creare molteplici controller
una stessa view.
Controller
ControllerVistaOpera
- model: Opera
- previous: Controller
- gotoPrevious(): void
+ connettiEventi(): void
- onCambiaUbicazioneClicked(): void
«Create»
+ create(view: VistaOpera, previous: Controller, model: Opera)


QMainWindow
VistaOpera
- previousButton: QPushButton
- immagineLabel: QLabel
- titoloLabel: QLabel
- autoreLabel: QLabel
- dimensioniLabel: QLabel
- periodoLabel: QLabel
- ubicazioneLabel: QLabel
- costoLabel: QLabel
- cambiaUbicazioneButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getImmagineLabel(): QLabel
+ getTitoloLabel(): QLabel
+ getAutoreLabel(): QLabel
+ getDimensioniLabel(): QLabel
+ getPeriodoLabel(): QLabel
+ getUbicazioneLabel(): QLabel
+ getCostoLabel(): QLabel
+ getCambiaUbicazioneButton(): QPushButton
Controller
era
(): void
evious:


+ getSitoWeb(): String {query}
+ getDataFondazione(): DateTime {query}
+ getIndirizzo(): String {query}
+ getTelefonoFisso(): Integer {query}
+ getEmail(): String {query}
+ getDescrizione(): String {query}
+ getSerializzatore(): Serializzatore {query}
+ getCloudStorage(): CloudStorage
«Create»
- create(): void
«property set»
+ setSitoWeb(newVal: String): void
+ setTelefonoFisso(newVal: Integer): void
+ setEmail(newVal: String): void
+ setDescrizione(newVal: String): void
+ setSerializzatore(newVal: Serializzatore): void
+ setCloudStorage(newVal: CloudStorage): void
1
1
1
{non-ordinata, univoca}
0..*

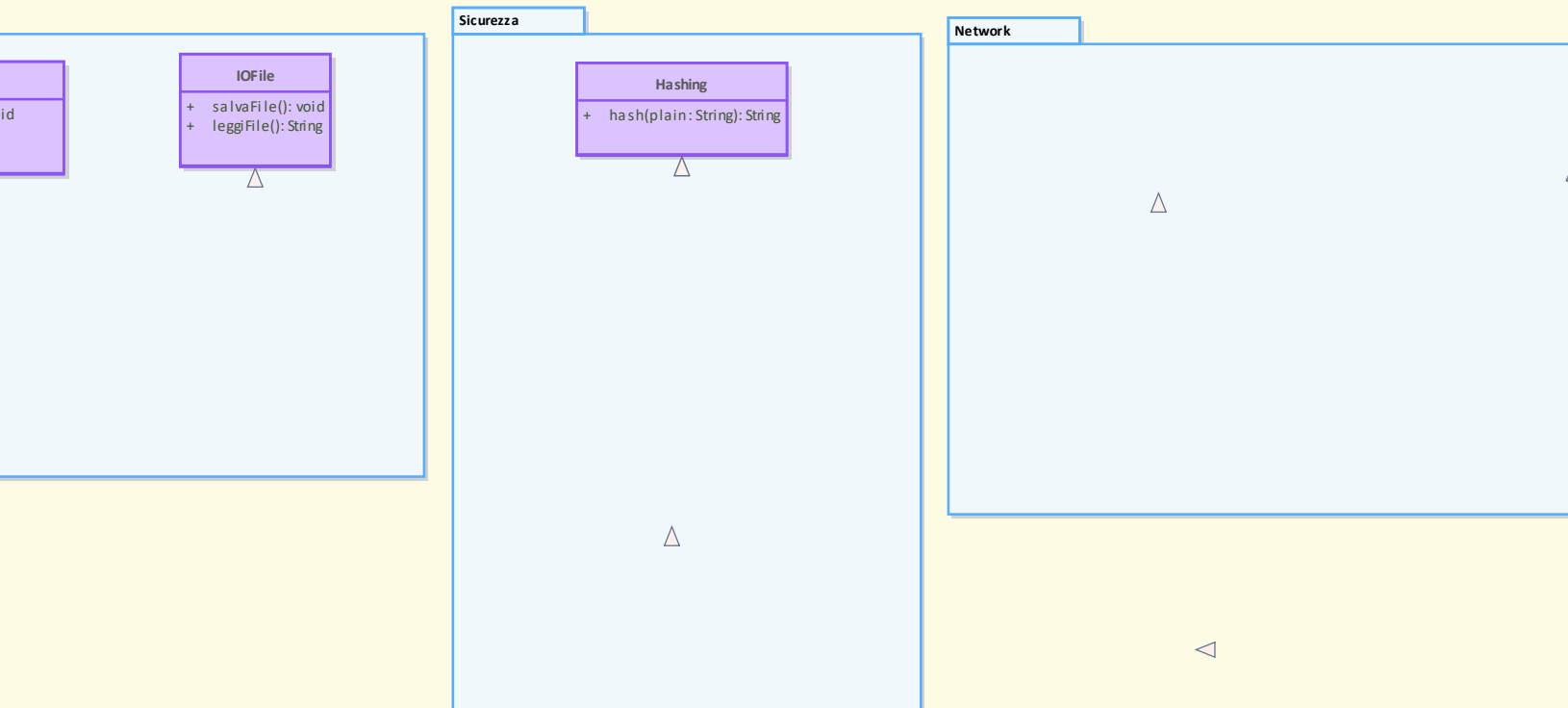

Utilies
IO
Serializzatore
+ serializza(obj: Object): void
+ deserializza(): Object

*(from Diagamma Delle Classi)*



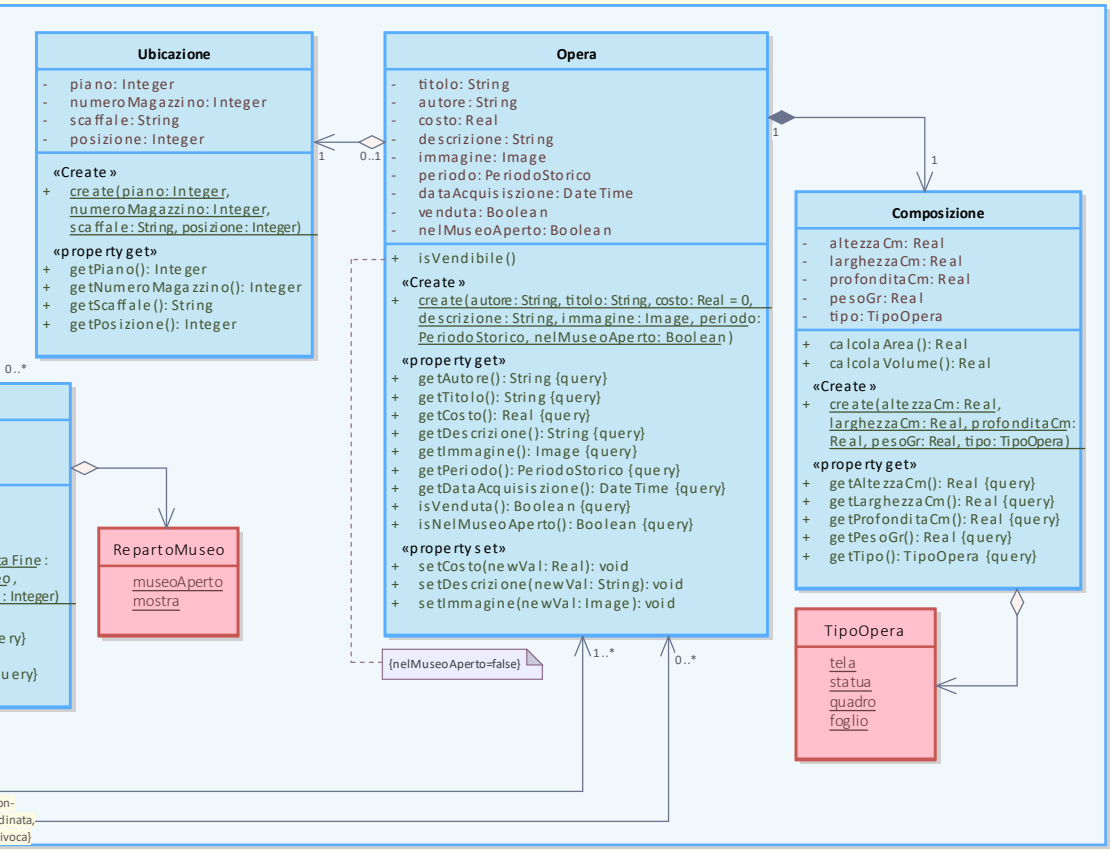

*(from Diagamma Delle Classi)*

*(from Diagamma Delle Classi)*



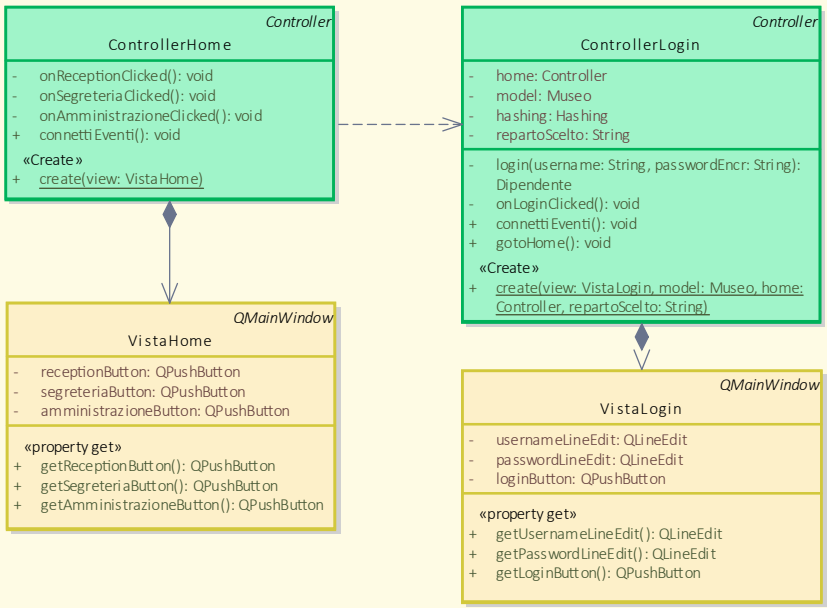
«Create»
+ create(username: String, password: String, hashing: Hashing)

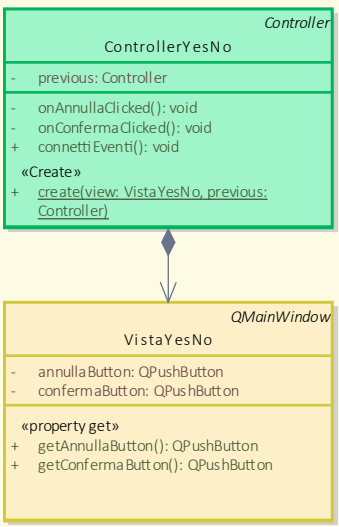
*om Diagamma Delle Classi)*

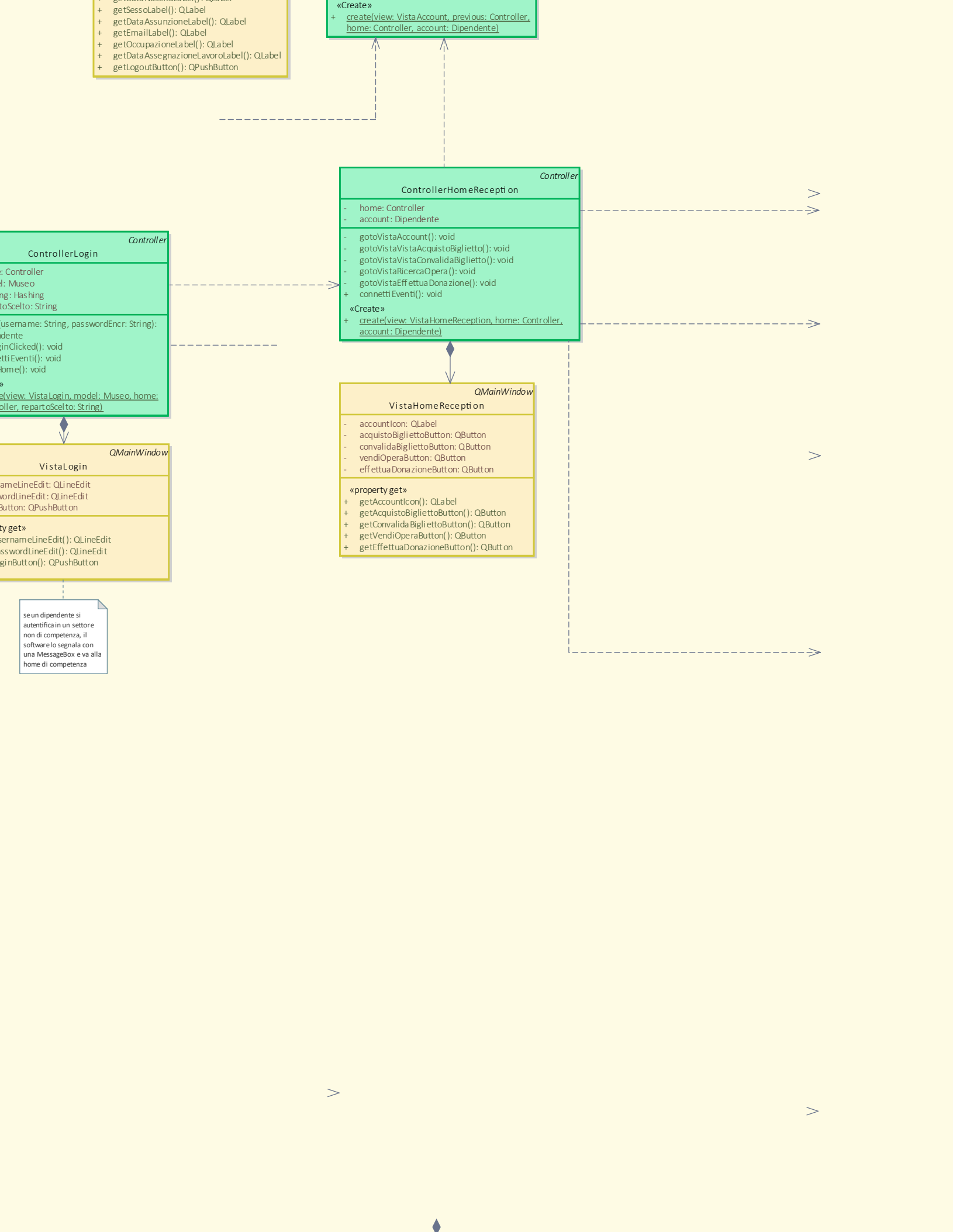


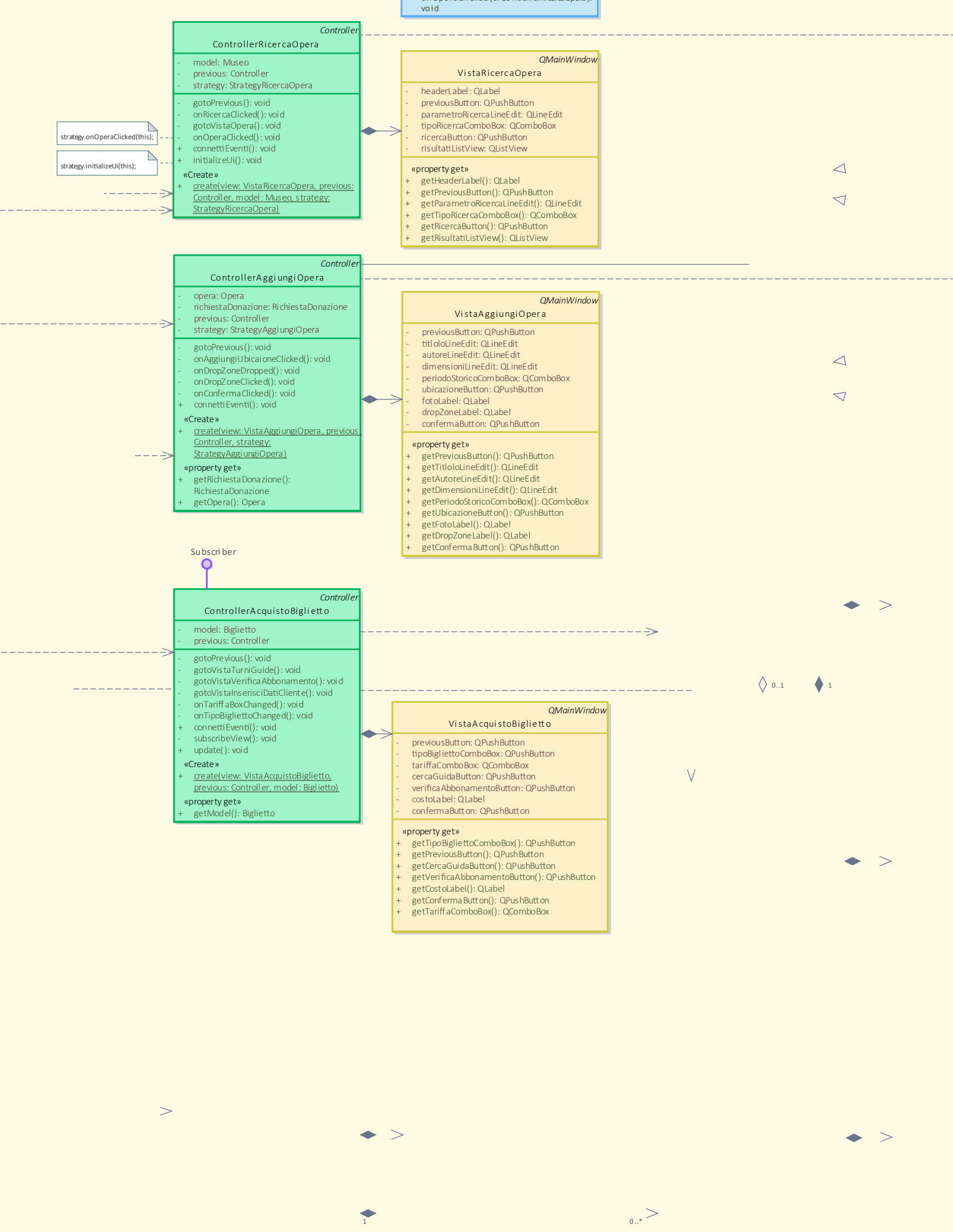

*(from Diagamma Delle Classi)*

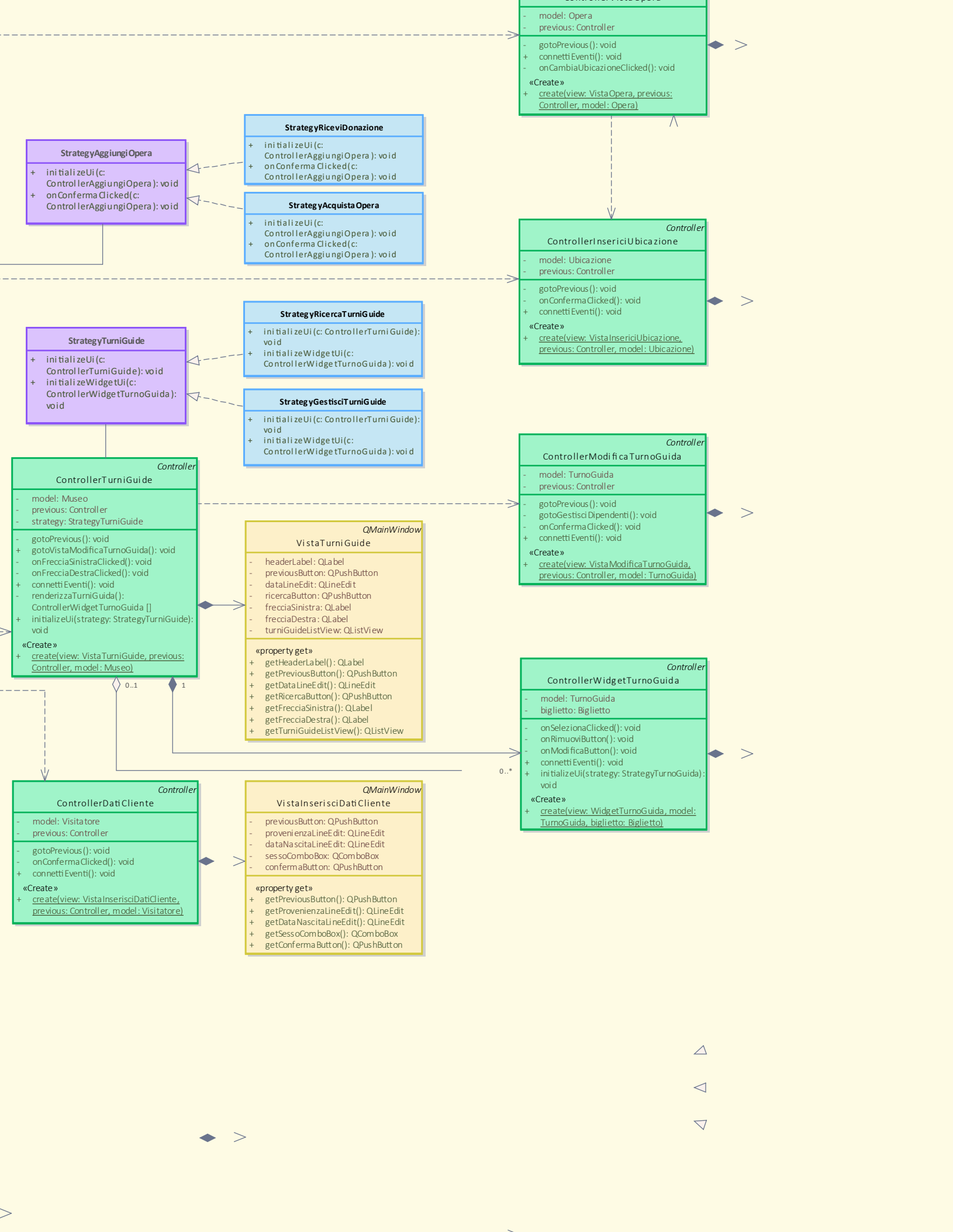
Tutte le classi controller estendono la superclasse *Controller*, questo mi permette di ustilizzare la tecnica del *late-binding*, in modo da poter assegnare un'istanza di sottotipo ad un riferimento di supertipo, secondo il principio di sostituibilità di Liskov.

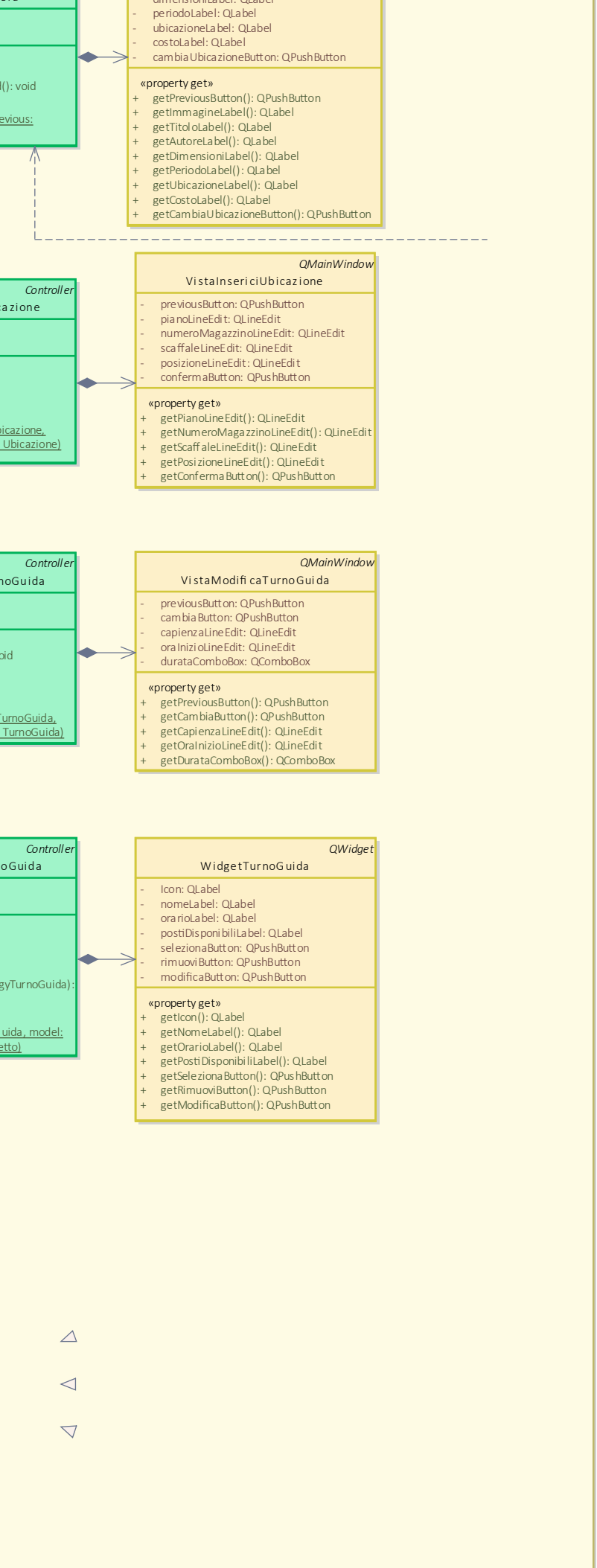


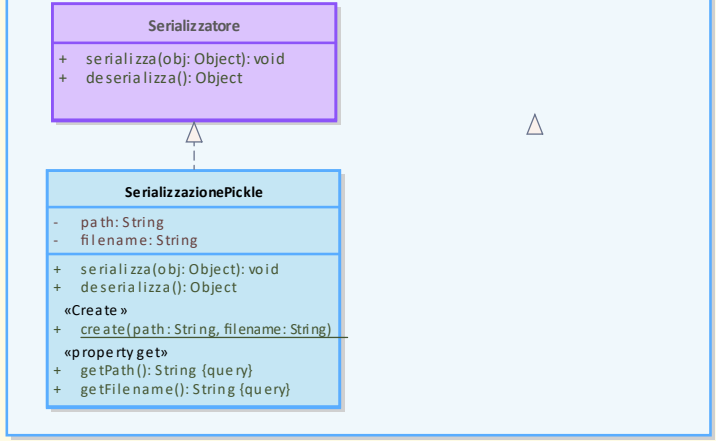




+ getSessoLabel(): QLabel
+ getDataAssunzioneLabel(): QLabel
+ getEmailLabel(): QLabel
+ getOccupazioneLabel(): QLabel
+ getDataAssegnazioneLavoroLabel(): QLabel
+ getLogoutButton(): QPushButton
«Create»
+ create(view: VistaAccount, previous: Controller, home: Controller, account: Dipendente)
Controller
ControllerHomeReception
- home: Controller
- account: Dipendente
- gotoVistaAccount(): void
- gotoVistaVistaAcquistoBiglietto(): void
- gotoVistaVistaConvalidaBiglietto(): void
- gotoVistaRicercaOpera(): void
- gotoVistaEffettuaDonazione(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaHomeReception, home: Controller, account: Dipendente)
Controller
ControllerLogin
: Controller
l: Museo
ng: Hashing
toScelto: String
username: String, passwordEncr: String):
dente
ginClicked(): void
tti Eventi(): void
Home(): void
»
e(view: VistaLogin, model: Museo, home:
oller, repartoScelto: String)
QMainWindow
VistaHomeReception
- accountIcon: QLabel
- acquistoBigliettoButton: QButton
- convalidaBigliettoButton: QButton
- vendiOperaButton: QButton
- effettuaDonazioneButton: QButton
«property get»
+ getAccountIcon(): QLabel
+ getAcquistoBigliettoButton(): QButton
+ getConvalidaBigliettoButton(): QButton
+ getVendiOperaButton(): QButton
+ getEffettuaDonazioneButton(): QButton
QMainWindow
VistaLogin
ameLineEdit: QLineEdit
wordLineEdit: QLineEdit
Button: QPushButton
ty get»
sernameLineEdit(): QLineEdit
sswordLineEdit(): QLineEdit
ginButton(): QPushButton
se un dipendente si autentifica in un settore non di competenza, il software lo segnala con una MessageBox e va alla home di competenza

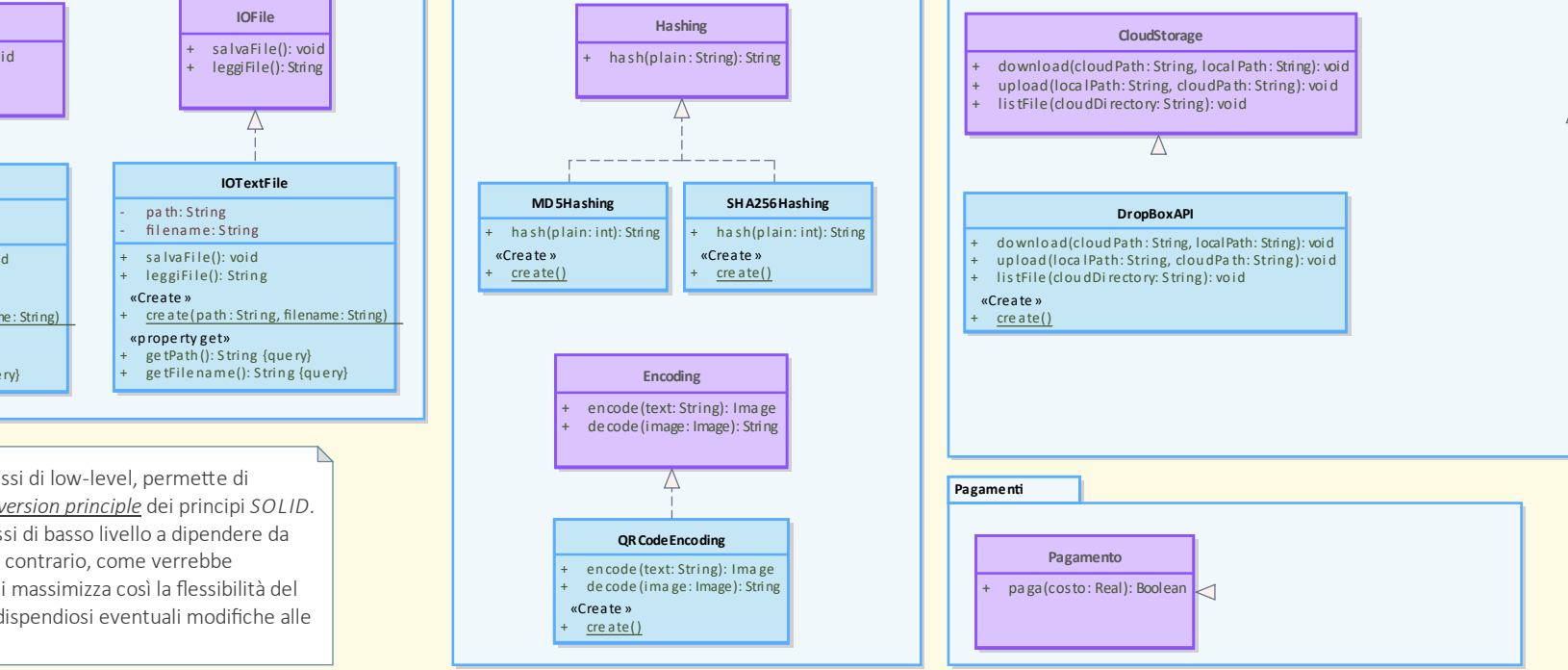

void
Controller
ControllerRicercaOpera
- model: Museo
- previous: Controller
- strategy: StrategyRicercaOpera
- gotoPrevious(): void
- onRicercaClicked(): void
- gotoVistaOpera(): void
- onOperaClicked(): void
+ connettiEventi(): void
+ initializeUi(): void
«Create»
+ create(view: VistaRicercaOpera, previous: Controller, model: Museo, strategy: StrategyRicercaOpera)
strategy.onOperaClicked(this);
strategy.initializeUi(this);
QMainWindow
VistaRicercaOpera
- headerLabel: QLabel
- previousButton: QPushButton
- parametroRicercaLineEdit: QLineEdit
- tipoRicercaComboBox: QComboBox
- ricercaButton: QPushButton
- risultatiListView: QListView
«property get»
+ getHeaderLabel(): QLabel
+ getPreviousButton(): QPushButton
+ getParametroRicercaLineEdit(): QLineEdit
+ getTipoRicercaComboBox(): QComboBox
+ getRicercaButton(): QPushButton
+ getRisultatiListView(): QListView
Controller
ControllerAggiungiOpera
- opera: Opera
- richiestaDonazione: RichiestaDonazione
- previous: Controller
- strategy: StrategyAggiungiOpera
- gotoPrevious(): void
- onAggiungiUbicaioneClicked(): void
- onDropZoneDropped(): void
- onDropZoneClicked(): void
- onConfermaClicked(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaAggiungiOpera, previous Controller, strategy: StrategyAggiungiOpera)
«property get»
+ getRichiestaDonazione(): RichiestaDonazione
+ getOpera(): Opera
QMainWindow
VistaAggiungiOpera
- previousButton: QPushButton
- titloloLineEdit: QLineEdit
- autoreLineEdit: QLineEdit
- dimensioniLineEdit: QLineEdit
- periodoStoricoComboBox: QComboBox
- ubicazioneButton: QPushButton
- fotoLabel: QLabel
- dropZoneLabel: QLabel
- confermaButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getTitloloLineEdit(): QLineEdit
+ getAutoreLineEdit(): QLineEdit
+ getDimensioniLineEdit(): QLineEdit
+ getPeriodoStoricoComboBox(): QComboBox
+ getUbicazioneButton(): QPushButton
+ getFotoLabel(): QLabel
+ getDropZoneLabel(): QLabel
+ getConfermaButton(): QPushButton
Subscriber
Controller
ControllerAcquistoBiglietto
- model: Biglietto
- previous: Controller
- gotoPrevious(): void
- gotoVistaTurniGuide(): void
- gotoVistaVerificaAbbonamento(): void
- gotoVistaInserisciDatiCliente(): void
- onTariffaBoxChanged(): void
- onTipoBiglietoChanged(): void
+ connettiEventi(): void
- subscribeView(): void
+ update(): void
«Create»
+ create(view: VistaAcquistoBiglietto, previous: Controller, model: Biglietto)
«property get»
+ getModel(): Biglietto
QMainWindow
VistaAcquistoBiglietto
- previousButton: QPushButton
- tipoBigliettoComboBox: QPushButton
- tariffaComboBox: QComboBox
- cercaGuidaButton: QPushButton
- verificaAbbonamentoButton: QPushButton
- costoLabel: QLabel
- confermaButton: QPushButton
«property get»
+ getTipoBigliettoComboBox(): QPushButton
+ getPreviousButton(): QPushButton
+ getCercaGuidaButton(): QPushButton
+ getVerificaAbbonamentoButton(): QPushButton
+ getCostoLabel(): QLabel
+ getConfermaButton(): QPushButton
+ getTariffaComboBox(): QComboBox
0..1
1
1
0..*


- model: Opera
- previous: Controller
- gotoPrevious(): void
+ connettiEventi(): void
- onCambiaUbicazioneClicked(): void
«Create»
+ create(view: VistaOpera, previous: Controller, model: Opera)
StrategyRiceviDonazione
+ initializeUi(c: ControllerAggiungiOpera): void
+ onConfermaClicked(c: ControllerAggiungiOpera): void
StrategyAggiungiOpera
+ initializeUi(c: ControllerAggiungiOpera): void
+ onConfermaClicked(c: ControllerAggiungiOpera): void
StrategyAcquistaOpera
+ initializeUi(c: ControllerAggiungiOpera): void
+ onConfermaClicked(c: ControllerAggiungiOpera): void
Controller
ControllerInsericiUbicazione
- model: Ubicazione
- previous: Controller
- gotoPrevious(): void
- onConfermaClicked(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaInsericiUbicazione, previous: Controller, model: Ubicazione)
StrategyRicercaTurniGuide
+ initializeUi(c: ControllerTurniGuide): void
+ initializeWidgetUi(c: ControllerWidgetTurnoGuida): void
StrategyTurniGuide
+ initializeUi(c: ControllerTurniGuide): void
+ initializeWidgetUi(c: ControllerWidgetTurnoGuida): void
StrategyGestisciTurniGuide
+ initializeUi(c: ControllerTurniGuide): void
+ initializeWidgetUi(c: ControllerWidgetTurnoGuida): void
Controller
ControllerModificaTurnoGuida
- model: TurnoGuida
- previous: Controller
- gotoPrevious(): void
- gotoGestisciDipendenti(): void
- onConfermaClicked(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaModificaTurnoGuida, previous: Controller, model: TurnoGuida)
Controller
ControllerTurniGuide
- model: Museo
- previous: Controller
- strategy: StrategyTurniGuide
- gotoPrevious(): void
+ gotoVistaModificaTurnoGuida(): void
- onFrecciaSinistraClicked(): void
- onFrecciaDestraClicked(): void
+ connettiEventi(): void
- renderizzaTurniGuida(): ControllerWidgetTurnoGuida []
+ initializeUi(strategy: StrategyTurniGuide): void
«Create»
+ create(view: VistaTurniGuide, previous: Controller, model: Museo)
QMainWindow
VistaTurniGuide
- headerLabel: QLabel
- previousButton: QPushButton
- dataLineEdit: QLineEdit
- ricercaButton: QPushButton
- frecciaSinistra: QLabel
- frecciaDestra: QLabel
- turniGuideListView: QListView
«property get»
+ getHeaderLabel(): QLabel
+ getPreviousButton(): QPushButton
+ getDataLineEdit(): QLineEdit
+ getRicercaButton(): QPushButton
+ getFrecciaSinistra(): QLabel
+ getFrecciaDestra(): QLabel
+ getTurniGuideListView(): QListView
0..1
1
0..*
Controller
ControllerWidgetTurnoGuida
- model: TurnoGuida
- biglietto: Biglietto
- onSelezionaClicked(): void
- onRimuoviButton(): void
- onModificaButton(): void
+ connettiEventi(): void
+ initializeUi(strategy: StrategyTurnoGuida): void
«Create»
+ create(view: WidgetTurnoGuida, model: TurnoGuida, biglietto: Biglietto)
Controller
ControllerDatiCliente
- model: Visitatore
- previous: Controller
- gotoPrevious(): void
- onConfermaClicked(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaInserisciDatiCliente, previous: Controller, model: Visitatore)
QMainWindow
VistaInserisciDatiCliente
- previousButton: QPushButton
- provenienzaLineEdit: QLineEdit
- dataNascitaLineEdit: QLineEdit
- sessoComboBox: QComboBox
- confermaButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getProvenienzaLineEdit(): QLineEdit
+ getDataNascitaLineEdit(): QLineEdit
+ getSessoComboBox(): QComboBox
+ getConfermaButton(): QPushButton

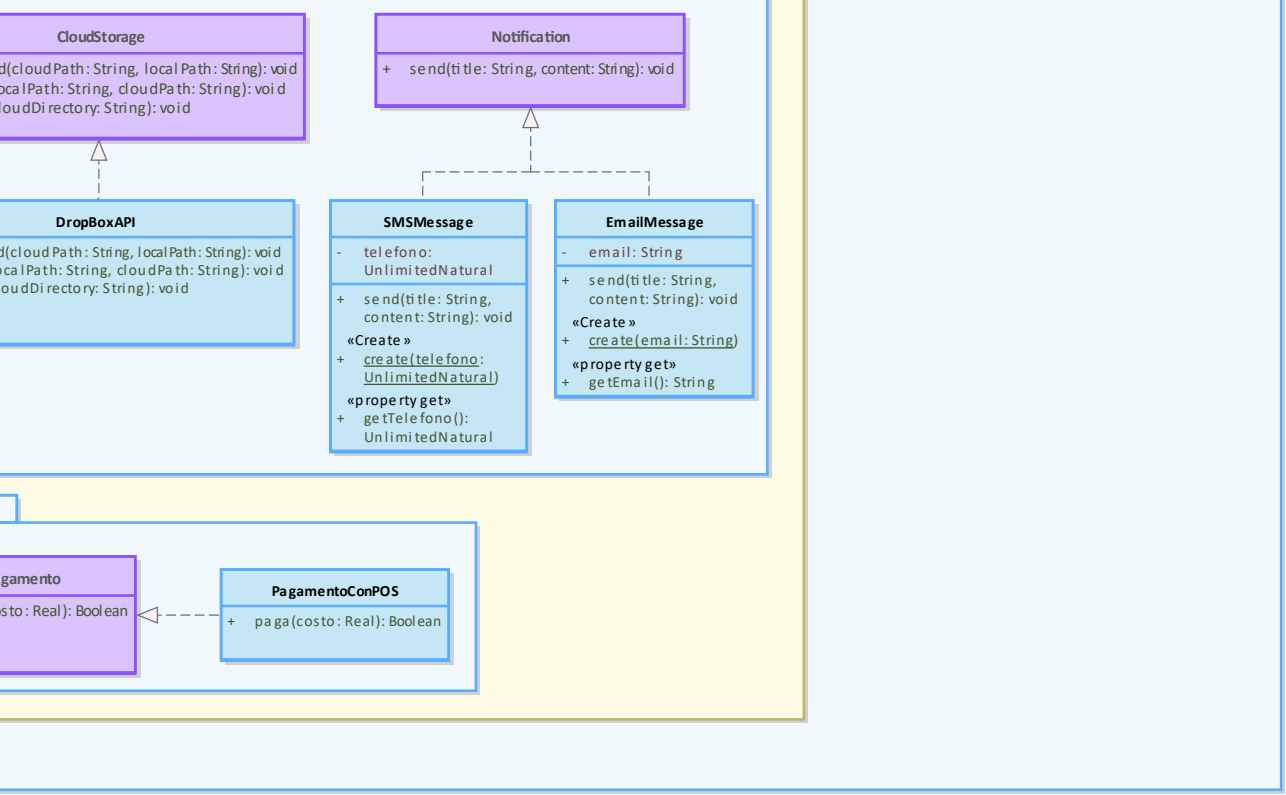

- periodoLabel: QLabel
- ubicazioneLabel: QLabel
- costoLabel: QLabel
- cambiaUbicazioneButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getImmagineLabel(): QLabel
+ getTitoloLabel(): QLabel
+ getAutoreLabel(): QLabel
+ getDimensioniLabel(): QLabel
+ getPeriodoLabel(): QLabel
+ getUbicazioneLabel(): QLabel
+ getCostoLabel(): QLabel
+ getCambiaUbicazioneButton(): QPushButton
QMainWindow
VistaInsericiUbicazione
- previousButton: QPushButton
- pianoLineEdit: QLineEdit
- numeroMagazzinoLineEdit: QLineEdit
- scaffaleLineEdit: QLineEdit
- posizioneLineEdit: QLineEdit
- confermaButton: QPushButton
«property get»
+ getPianoLineEdit(): QLineEdit
+ getNumeroMagazzinoLineEdit(): QLineEdit
+ getScaffaleLineEdit(): QLineEdit
+ getPosizioneLineEdit(): QLineEdit
+ getConfermaButton(): QPushButton
QMainWindow
VistaModificaTurnoGuida
- previousButton: QPushButton
- cambiaButton: QPushButton
- capienzaLineEdit: QLineEdit
- oraInizioLineEdit: QLineEdit
- durataComboBox: QComboBox
«property get»
+ getPreviousButton(): QPushButton
+ getCambiaButton(): QPushButton
+ getCapienzaLineEdit(): QLineEdit
+ getOraInizioLineEdit(): QLineEdit
+ getDurataComboBox(): QComboBox
QWidget
WidgetTurnoGuida
- Icon: QLabel
- nomeLabel: QLabel
- orarioLabel: QLabel
- postiDisponibiliLabel: QLabel
- selezionaButton: QPushButton
- rimuoviButton: QPushButton
- modificaButton: QPushButton
«property get»
+ getIcon(): QLabel
+ getNomeLabel(): QLabel
+ getOrarioLabel(): QLabel
+ getPostiDisponibiliLabel(): QLabel
+ getSelezionaButton(): QPushButton
+ getRimuoviButton(): QPushButton
+ getModificaButton(): QPushButton

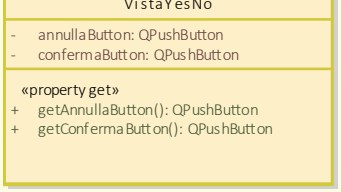


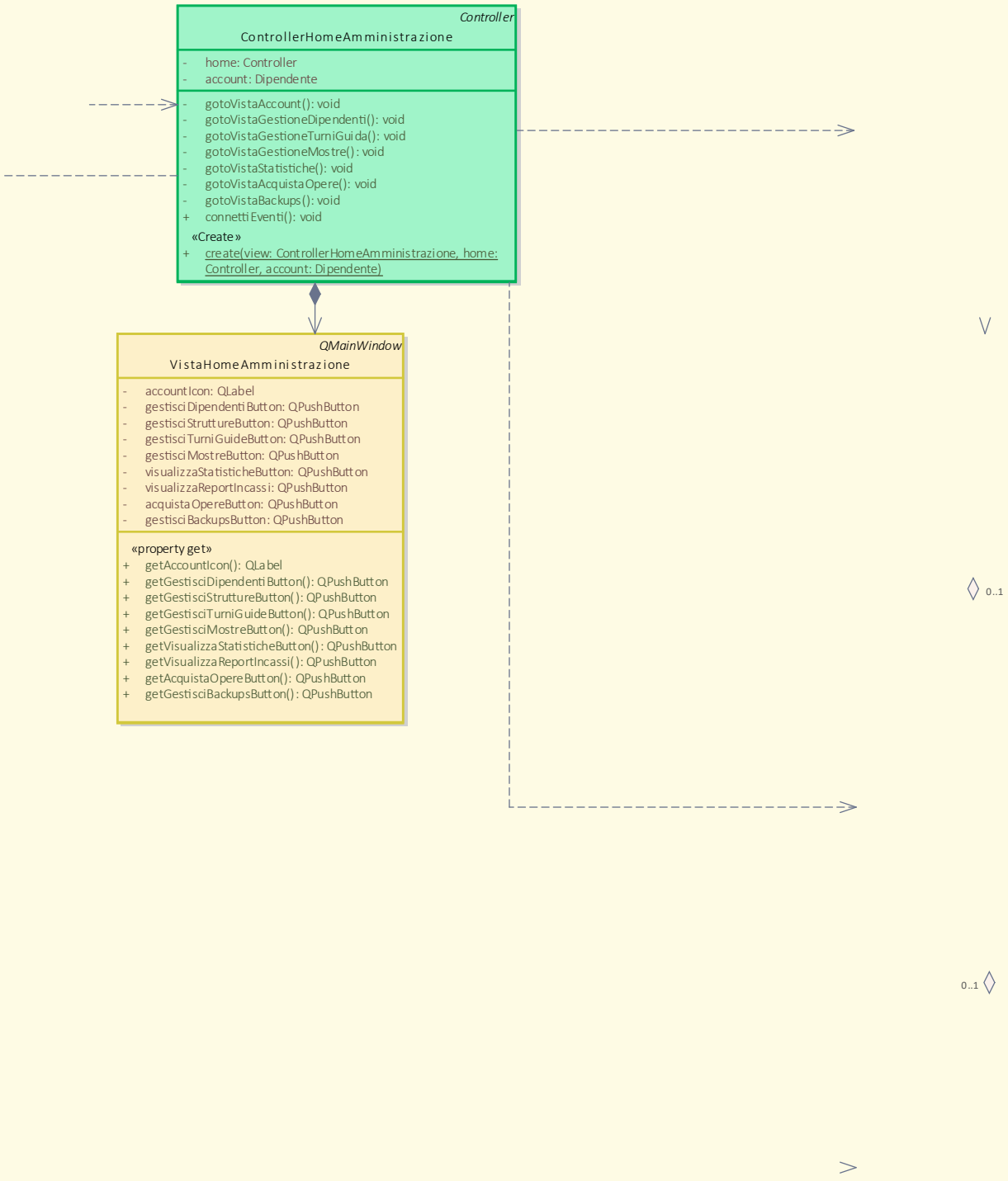

L'uso di interfacce tra le classi di low-level, permette di rispettare il *Dependency inversion principle* dei principi *SOLID*. In questo modo sono le classi di basso livello a dipendere da quelle di alto livello e non il contrario, come verrebbe naturalmente da pensare. Si massimizza così la flessibilità del software, rendendo meno dispendiosi eventuali modifiche alle operazioni di basso livello.

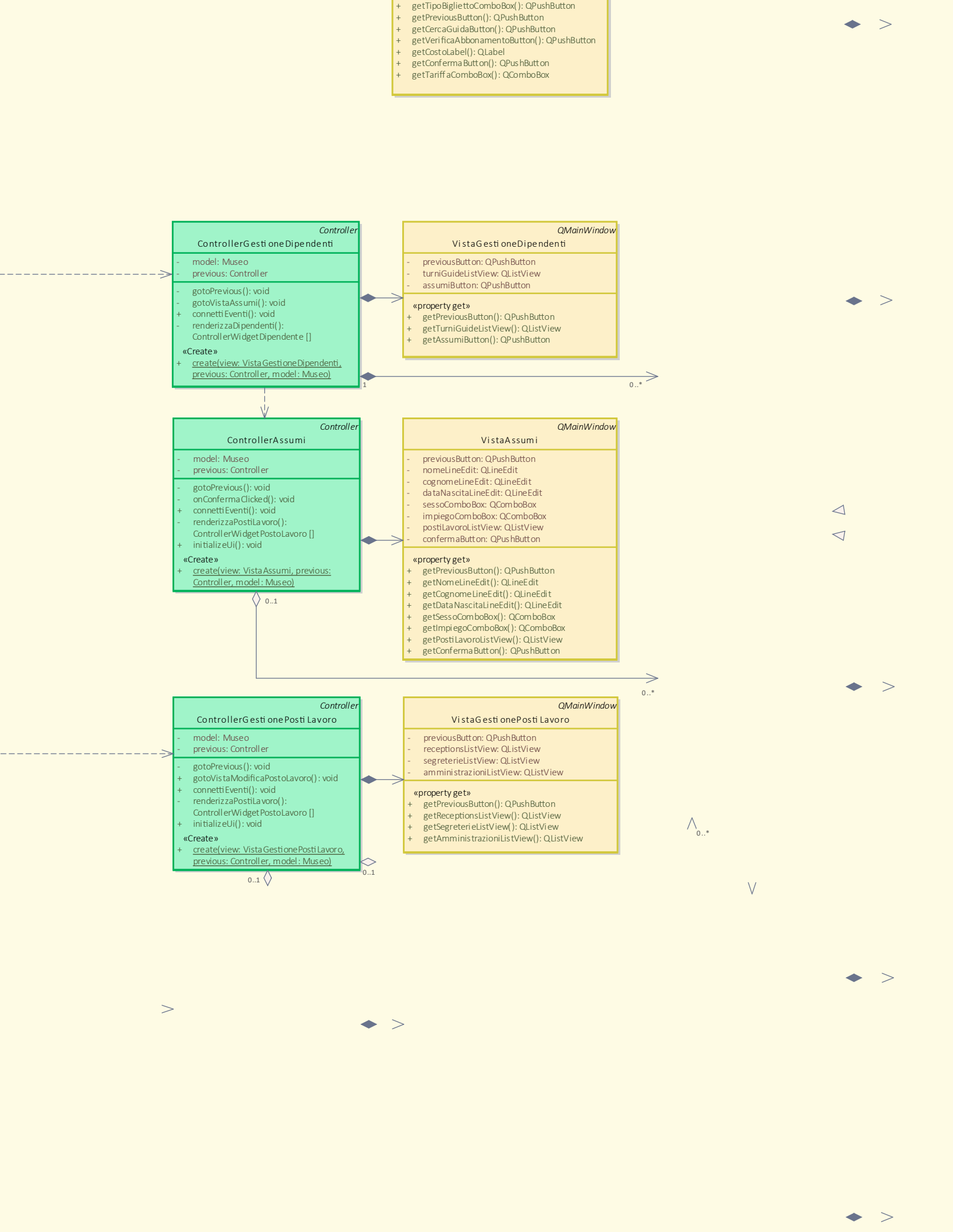


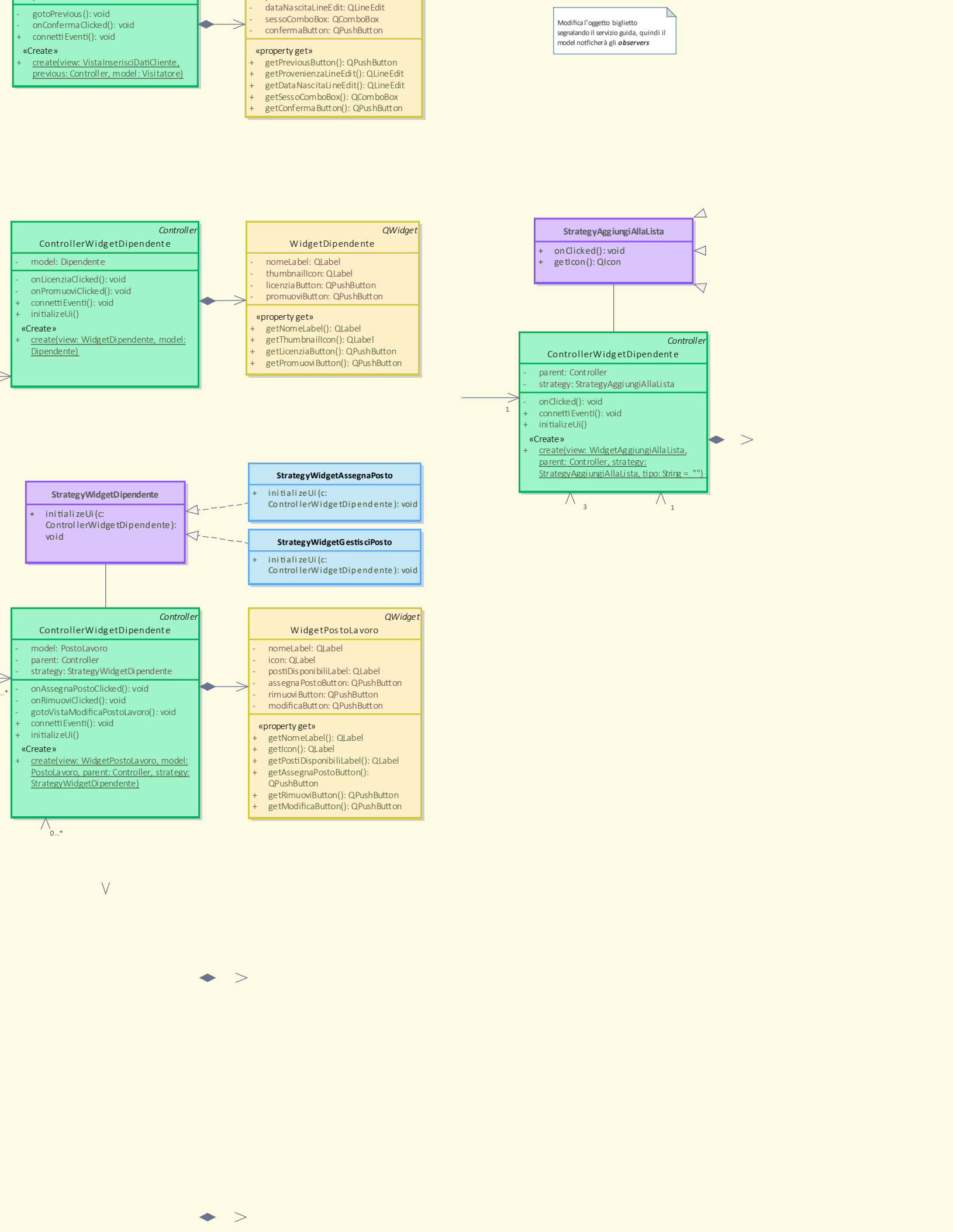

ssi di low-level, permette di
*version principle* dei principi *SOLID*.
ssi di basso livello a dipendere da
contrario, come verrebbe
i massimizza così la flessibilità del
dispendiosi eventuali modifiche alle

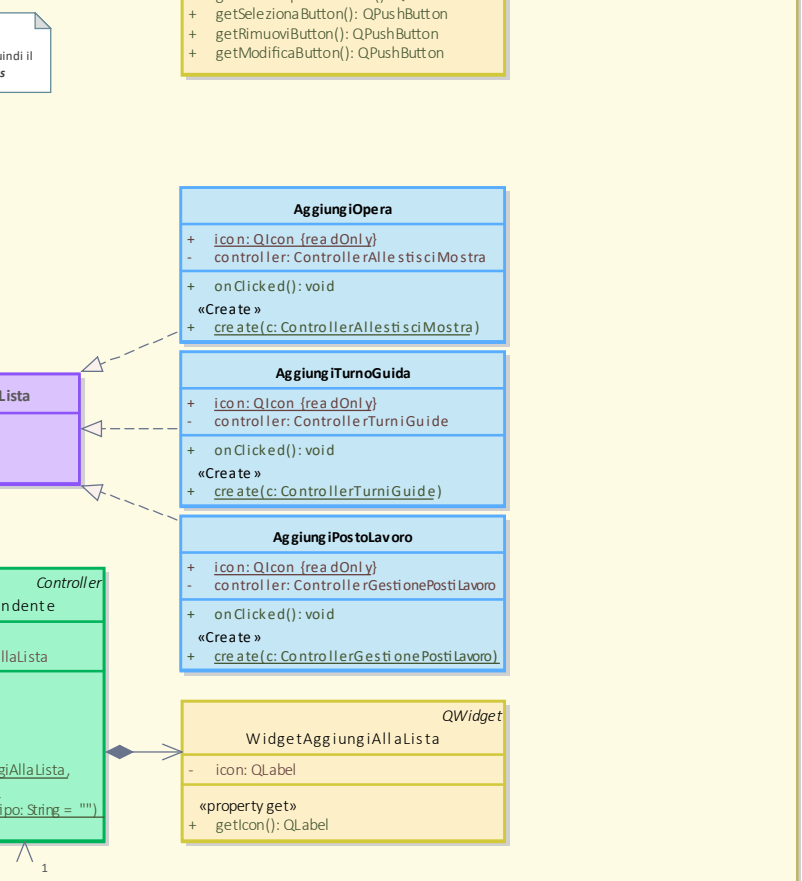

CloudStorage
d(cloudPath: String, localPath: String): void
ocalPath: String, cloudPath: String): void
loudDirectory: String): void
Notification
+ send(title: String, content: String): void
DropBoxAPI
d(cloudPath: String, localPath: String): void
ocalPath: String, cloudPath: String): void
loudDirectory: String): void
SMSMessage
- telefono: UnlimitedNatural
+ send(title: String, content: String): void
«Create»
+ create(telefono: UnlimitedNatural)
«property get»
+ getTelefono(): UnlimitedNatural
EmailMessage
- email: String
+ send(title: String, content: String): void
«Create»
+ create(email: String)
«property get»
+ getEmail(): String
gamento
sto: Real): Boolean
PagamentoConPOS
+ paga(costo: Real): Boolean

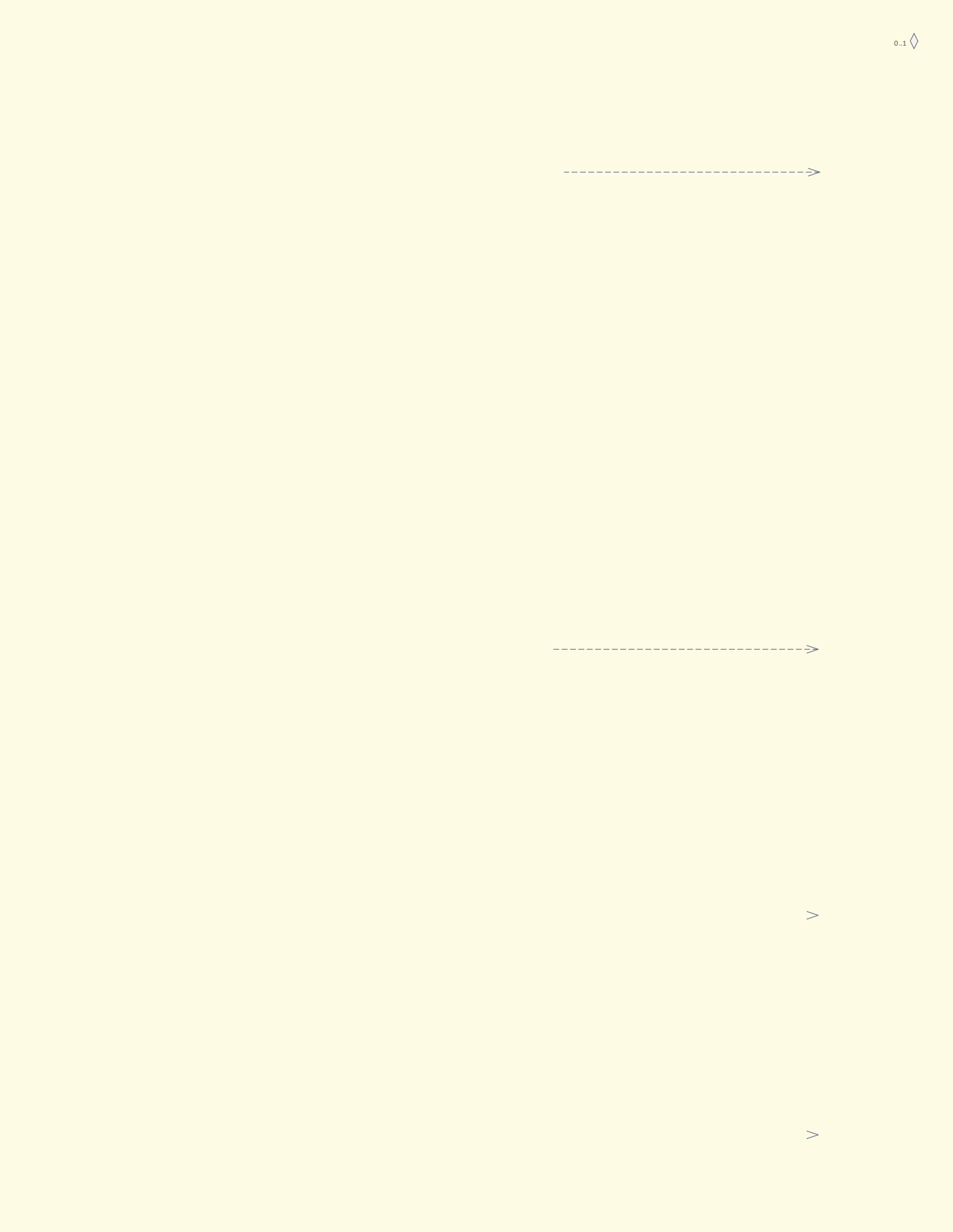

VistaYesNo
- annullaButton: QPushButton
- confermaButton: QPushButton
«property get»
+ getAnnullaButton(): QPushButton
+ getConfermaButton(): QPushButton

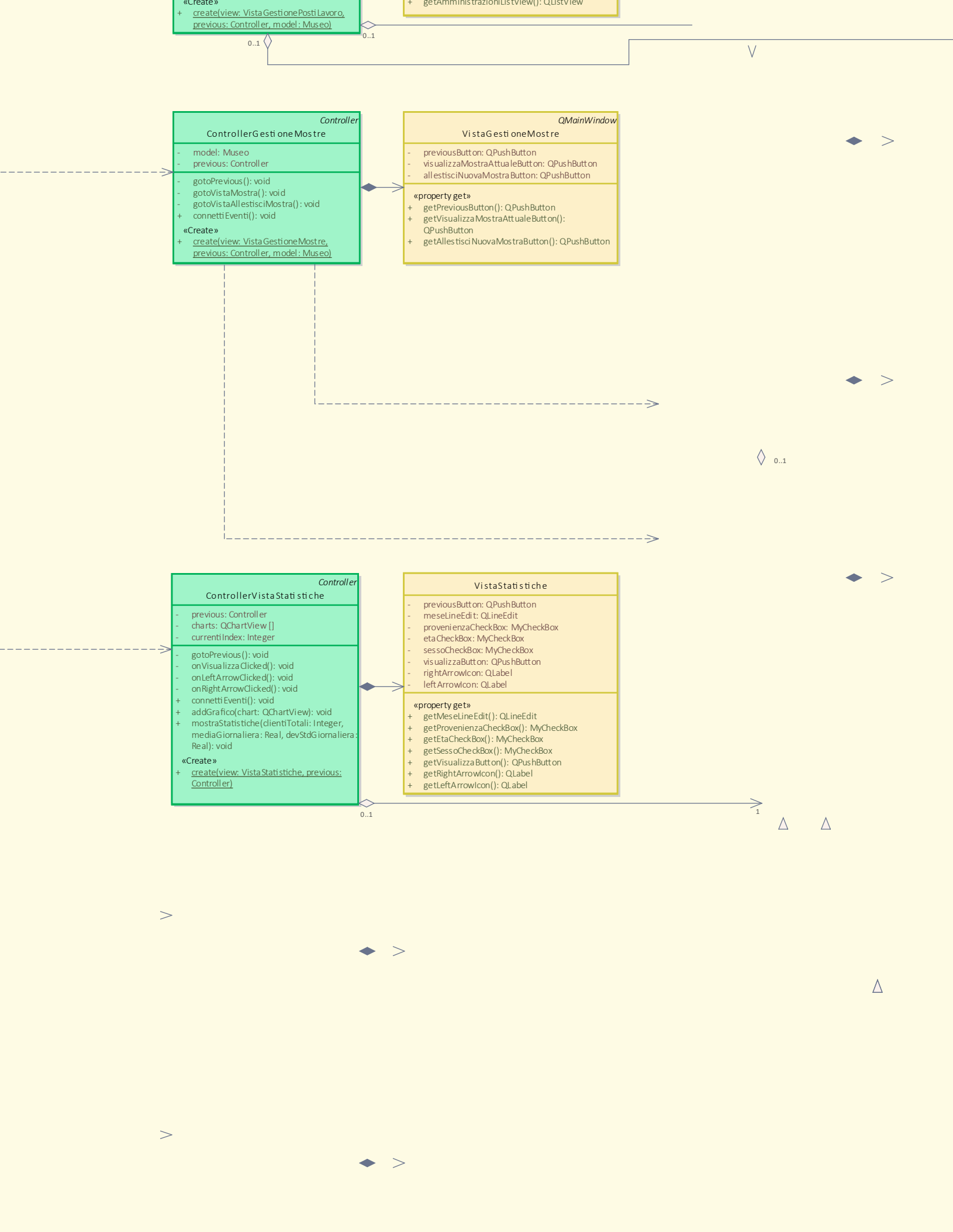

Controller
ControllerHomeAmministrazione
- home: Controller
- account: Dipendente
- gotoVistaAccount(): void
- gotoVistaGestioneDipendenti(): void
- gotoVistaGestioneTurniGuida(): void
- gotoVistaGestioneMostre(): void
- gotoVistaStatistiche(): void
- gotoVistaAcquistaOpere(): void
- gotoVistaBackups(): void
+ connettiEventi(): void
«Create»
+ create(view: ControllerHomeAmministrazione, home: Controller, account: Dipendente)
QMainWindow
VistaHomeAmministrazione
- accountIcon: QLabel
- gestisciDipendentiButton: QPushButton
- gestisciStruttureButton: QPushButton
- gestisciTurniGuideButton: QPushButton
- gestisciMostreButton: QPushButton
- visualizzaStatisticheButton: QPushButton
- visualizzaReportIncassi: QPushButton
- acquistaOpereButton: QPushButton
- gestisciBackupsButton: QPushButton
«property get»
+ getAccountIcon(): QLabel
+ getGestisciDipendentiButton(): QPushButton
+ getGestisciStruttureButton(): QPushButton
+ getGestisciTurniGuideButton(): QPushButton
+ getGestisciMostreButton(): QPushButton
+ getVisualizzaStatisticheButton(): QPushButton
+ getVisualizzaReportIncassi(): QPushButton
+ getAcquistaOpereButton(): QPushButton
+ getGestisciBackupsButton(): QPushButton
0..1
0..1

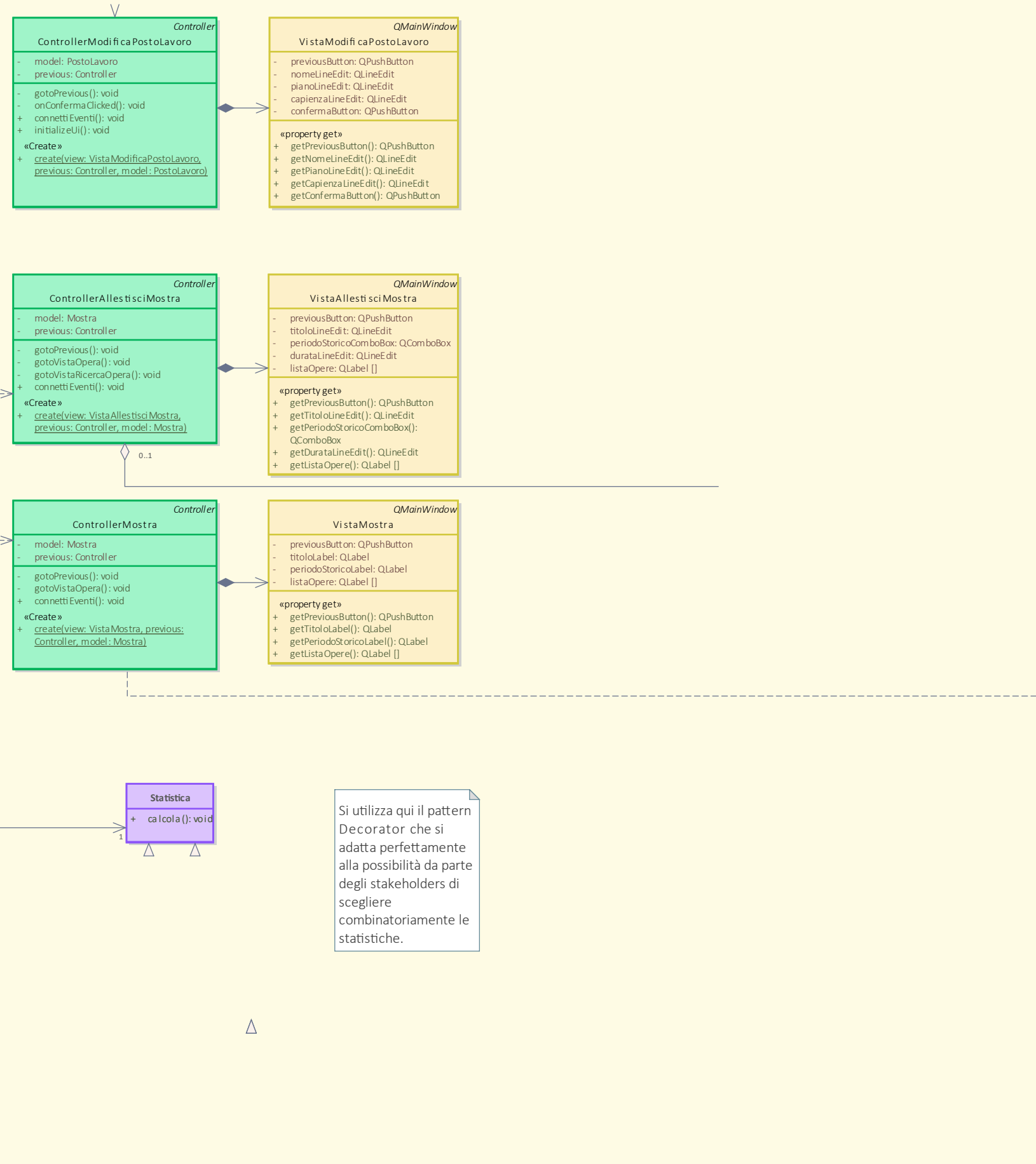

+ getTipoBigliettoComboBox(): QPushButton
+ getPreviousButton(): QPushButton
+ getCercaGuidaButton(): QPushButton
+ getVerificaAbbonamentoButton(): QPushButton
+ getCostoLabel(): QLabel
+ getConfermaButton(): QPushButton
+ getTariffaComboBox(): QComboBox
Controller
ControllerGestioneDipendenti
- model: Museo
- previous: Controller
- gotoPrevious(): void
- gotoVistaAssumi(): void
+ connettiEventi(): void
- renderizzaDipendenti(): ControllerWidgetDipendente []
«Create»
+ create(view: VistaGestioneDipendenti, previous: Controller, model: Museo)
QMainWindow
VistaGestioneDipendenti
- previousButton: QPushButton
- turniGuideListView: QListView
- assumiButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getTurniGuideListView(): QListView
+ getAssumiButton(): QPushButton
1
0..*
Controller
ControllerAssumi
- model: Museo
- previous: Controller
- gotoPrevious(): void
- onConfermaClicked(): void
+ connettiEventi(): void
- renderizzaPostiLavoro(): ControllerWidgetPostoLavoro []
+ initializeUi(): void
«Create»
+ create(view: VistaAssumi, previous: Controller, model: Museo)
QMainWindow
VistaAssumi
- previousButton: QPushButton
- nomeLineEdit: QLineEdit
- cognomeLineEdit: QLineEdit
- dataNascitaLineEdit: QLineEdit
- sessoComboBox: QComboBox
- impiegoComboBox: QComboBox
- postiLavoroListView: QListView
- confermaButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getNomeLineEdit(): QLineEdit
+ getCognomeLineEdit(): QLineEdit
+ getDataNascitaLineEdit(): QLineEdit
+ getSessoComboBox(): QComboBox
+ getImpiegoComboBox(): QComboBox
+ getPostiLavoroListView(): QListView
+ getConfermaButton(): QPushButton
0..1
0..*
Controller
ControllerGestionePostiLavoro
- model: Museo
- previous: Controller
- gotoPrevious(): void
+ gotoVistaModificaPostoLavoro(): void
+ connettiEventi(): void
- renderizzaPostiLavoro(): ControllerWidgetPostoLavoro []
+ initializeUi(): void
«Create»
+ create(view: VistaGestionePostiLavoro, previous: Controller, model: Museo)
QMainWindow
VistaGestionePostiLavoro
- previousButton: QPushButton
- receptionsListView: QListView
- segreterieListView: QListView
- amministrazioniListView: QListView
«property get»
+ getPreviousButton(): QPushButton
+ getReceptionsListView(): QListView
+ getSegreterieListView(): QListView
+ getAmministrazioniListView(): QListView
0..1
0..1
0..*

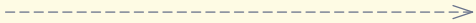
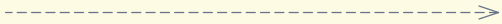

- gotoPrevious(): void
- onConfermaClicked(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaInserisciDatiCliente, previous: Controller, model: Visitatore)
- dataNascitaLineEdit: QLineEdit
- sessoComboBox: QComboBox
- confermaButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getProvenienzaLineEdit(): QLineEdit
+ getDataNascitaLineEdit(): QLineEdit
+ getSessoComboBox(): QComboBox
+ getConfermaButton(): QPushButton
Modifica l'oggetto biglietto segnalando il servizio guida, quindi il model notficherà gli observers
Controller
ControllerWidgetDipendente
- model: Dipendente
- onLicenziaClicked(): void
- onPromuoviClicked(): void
+ connettiEventi(): void
+ initializeUi()
«Create»
+ create(view: WidgetDipendente, model: Dipendente)
QWidget
WidgetDipendente
- nomeLabel: QLabel
- thumbnailIcon: QLabel
- licenziaButton: QPushButton
- promuoviButton: QPushButton
«property get»
+ getNomeLabel(): QLabel
+ getThumbnailIcon(): QLabel
+ getLicenziaButton(): QPushButton
+ getPromuoviButton(): QPushButton
StrategyAggiungiAllaLista
+ onClicked(): void
+ getIcon(): QIcon
Controller
ControllerWidgetDipendente
- parent: Controller
- strategy: StrategyAggiungiAllaLista
- onClicked(): void
+ connettiEventi(): void
+ initializeUi()
«Create»
+ create(view: WidgetAggiungiAllaLista, parent: Controller, strategy: StrategyAggiungiAllaLista, tipo: String = "")
1
3
1
StrategyWidgetDipendente
+ initializeUi(c: ControllerWidgetDipendente): void
StrategyWidgetAssegnaPosto
+ initializeUi(c: ControllerWidgetDipendente): void
StrategyWidgetGestisciPosto
+ initializeUi(c: ControllerWidgetDipendente): void
Controller
ControllerWidgetDipendente
- model: PostoLavoro
- parent: Controller
- strategy: StrategyWidgetDipendente
- onAssegnaPostoClicked(): void
- onRimuoviClicked(): void
- gotoVistaModificaPostoLavoro(): void
+ connettiEventi(): void
+ initializeUi()
«Create»
+ create(view: WidgetPostoLavoro, model: PostoLavoro, parent: Controller, strategy: StrategyWidgetDipendente)
0..*
QWidget
WidgetPostoLavoro
- nomeLabel: QLabel
- icon: QLabel
- postiDisponibiliLabel: QLabel
- assegnaPostoButton: QPushButton
- rimuoviButton: QPushButton
- modificaButton: QPushButton
«property get»
+ getNomeLabel(): QLabel
+ getIcon(): QLabel
+ getPostiDisponibiliLabel(): QLabel
+ getAssegnaPostoButton(): QPushButton
+ getRimuoviButton(): QPushButton
+ getModificaButton(): QPushButton

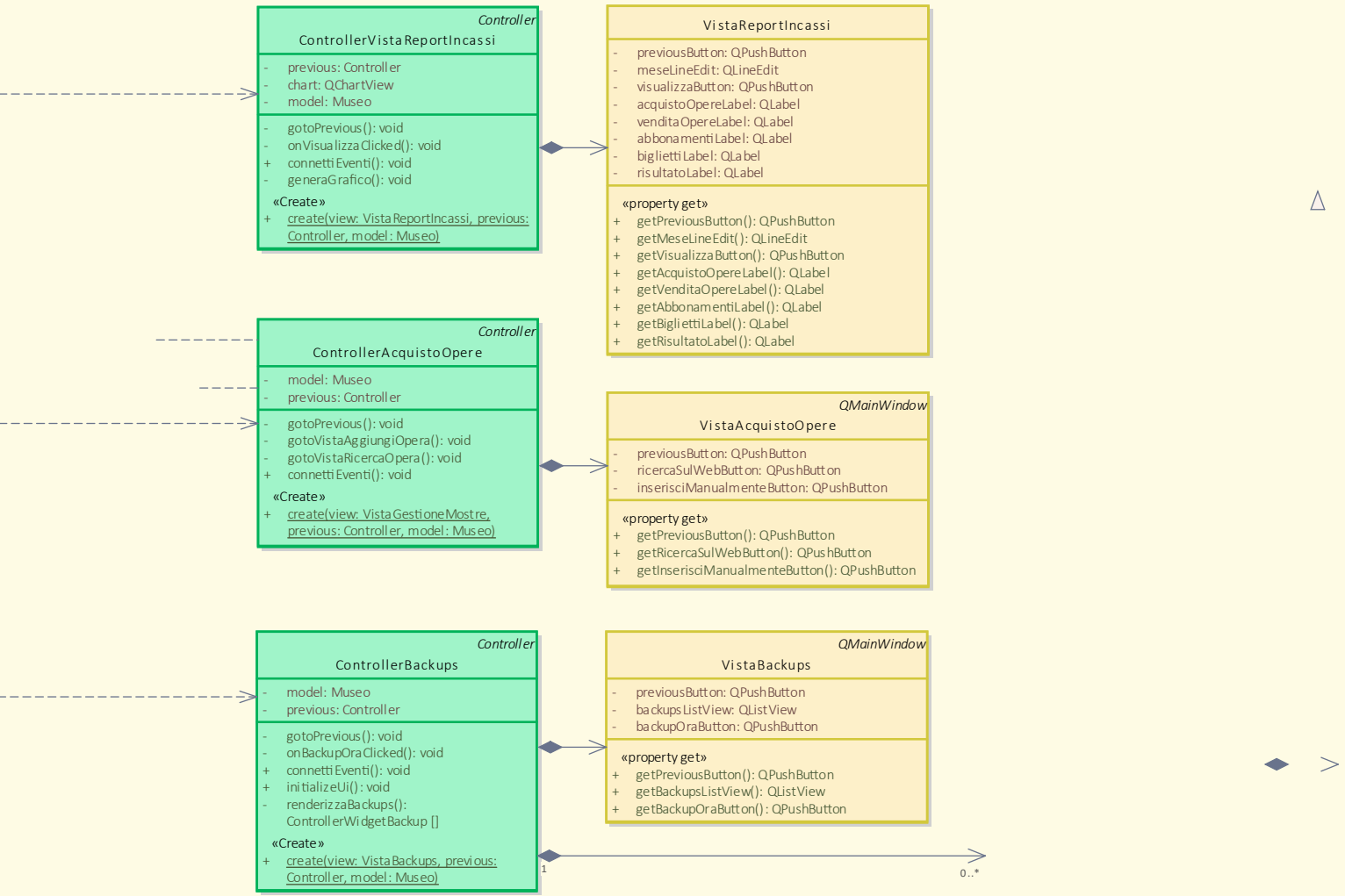

+ getSelezionaButton(): QPushButton
+ getRimuoviButton(): QPushButton
+ getModificaButton(): QPushButton
AggiungiOpera
+ icon: QIcon {readOnly}
- controller: ControllerAllestisciMostra
+ onClicked(): void
«Create»
+ create(c: ControllerAllestisciMostra)
AggiungiTurnoGuida
+ icon: QIcon {readOnly}
- controller: ControllerTurniGuide
+ onClicked(): void
«Create»
+ create(c: ControllerTurniGuide)
AggiungiPostoLavoro
+ icon: QIcon {readOnly}
- controller: ControllerGestionePostiLavoro
+ onClicked(): void
«Create»
+ create(c: ControllerGestionePostiLavoro)
QWidget
WidgetAggiungiAllaLista
- icon: QLabel
«property get»
+ getIcon(): QLabel
1


0..1


«Create»
+ create(view: VistaGestionePostiLavoro, previous: Controller, model: Museo)
0..1
0..1
getAmministrazioniListView(): QListView
Controller
ControllerGestioneMostre
- model: Museo
- previous: Controller
- gotoPrevious(): void
- gotoVistaMostra(): void
- gotoVistaAllestisciMostra(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaGestioneMostre, previous: Controller, model: Museo)
QMainWindow
VistaGestioneMostre
- previousButton: QPushButton
- visualizzaMostraAttualeButton: QPushButton
- allestisciNuovaMostraButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getVisualizzaMostraAttualeButton(): QPushButton
+ getAllestisciNuovaMostraButton(): QPushButton
0..1
Controller
ControllerVistaStatistiche
- previous: Controller
- charts: QChartView []
- currentIndex: Integer
- gotoPrevious(): void
- onVisualizzaClicked(): void
- onLeftArrowClicked(): void
- onRightArrowClicked(): void
+ connettiEventi(): void
+ addGrafico(chart: QChartView): void
+ mostraStatistiche(clientiTotali: Integer, mediaGiornaliera: Real, devStdGiornaliera: Real): void
«Create»
+ create(view: VistaStatistiche, previous: Controller)
VistaStatistiche
- previousButton: QPushButton
- meseLineEdit: QLineEdit
- provenienzaCheckBox: MyCheckBox
- etaCheckBox: MyCheckBox
- sessoCheckBox: MyCheckBox
- visualizzaButton: QPushButton
- rightArrowIcon: QLabel
- leftArrowIcon: QLabel
«property get»
+ getMeseLineEdit(): QLineEdit
+ getProvenienzaCheckBox(): MyCheckBox
+ getEtaCheckBox(): MyCheckBox
+ getSessoCheckBox(): MyCheckBox
+ getVisualizzaButton(): QPushButton
+ getRightArrowIcon(): QLabel
+ getLeftArrowIcon(): QLabel
0..1
1


Controller
ControllerModificaPostoLavoro
- model: PostoLavoro
- previous: Controller
- gotoPrevious(): void
- onConfermaClicked(): void
+ connettiEventi(): void
+ initializeUi(): void
«Create»
+ create(view: VistaModificaPostoLavoro, previous: Controller, model: PostoLavoro)
QMainWindow
VistaModificaPostoLavoro
- previousButton: QPushButton
- nomeLineEdit: QLineEdit
- pianoLineEdit: QLineEdit
- capienzaLineEdit: QLineEdit
- confermaButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getNomeLineEdit(): QLineEdit
+ getPianoLineEdit(): QLineEdit
+ getCapienzaLineEdit(): QLineEdit
+ getConfermaButton(): QPushButton
Controller
ControllerAllestisciMostra
- model: Mostra
- previous: Controller
- gotoPrevious(): void
- gotoVistaOpera(): void
- gotoVistaRicercaOpera(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaAllestisciMostra, previous: Controller, model: Mostra)
QMainWindow
VistaAllestisciMostra
- previousButton: QPushButton
- titoloLineEdit: QLineEdit
- periodoStoricoComboBox: QComboBox
- durataLineEdit: QLineEdit
- listaOpere: QLabel []
«property get»
+ getPreviousButton(): QPushButton
+ getTitoloLineEdit(): QLineEdit
+ getPeriodoStoricoComboBox(): QComboBox
+ getDurataLineEdit(): QLineEdit
+ getListaOpere(): QLabel []
0..1
Controller
ControllerMostra
- model: Mostra
- previous: Controller
- gotoPrevious(): void
- gotoVistaOpera(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaMostra, previous: Controller, model: Mostra)
QMainWindow
VistaMostra
- previousButton: QPushButton
- titoloLabel: QLabel
- periodoStoricoLabel: QLabel
- listaOpere: QLabel []
«property get»
+ getPreviousButton(): QPushButton
+ getTitoloLabel(): QLabel
+ getPeriodoStoricoLabel(): QLabel
+ getListaOpere(): QLabel []
Statistica
+ calcola(): void
1
Si utilizza qui il pattern Decorator che si adatta perfettamente alla possibilità da parte degli stakeholders di scegliere combinatoriamente le statistiche.



Controller
ControllerVistaReportIncassi
- previous: Controller
- chart: QChartView
- model: Museo
- gotoPrevious(): void
- onVisualizzaClicked(): void
+ connettiEventi(): void
- generaGrafico(): void
«Create»
+ create(view: VistaReportIncassi, previous: Controller, model: Museo)
VistaReportIncassi
- previousButton: QPushButton
- meseLineEdit: QLineEdit
- visualizzaButton: QPushButton
- acquistoOpereLabel: QLabel
- venditaOpereLabel: QLabel
- abbonamentiLabel: QLabel
- bigliettiLabel: QLabel
- risultatoLabel: QLabel
«property get»
+ getPreviousButton(): QPushButton
+ getMeseLineEdit(): QLineEdit
+ getVisualizzaButton(): QPushButton
+ getAcquistoOpereLabel(): QLabel
+ getVenditaOpereLabel(): QLabel
+ getAbbonamentiLabel(): QLabel
+ getBigliettiLabel(): QLabel
+ getRisultatoLabel(): QLabel
Controller
ControllerAcquistoOpere
- model: Museo
- previous: Controller
- gotoPrevious(): void
- gotoVistaAggiungiOpera(): void
- gotoVistaRicercaOpera(): void
+ connettiEventi(): void
«Create»
+ create(view: VistaGestioneMostre, previous: Controller, model: Museo)
QMainWindow
VistaAcquistoOpere
- previousButton: QPushButton
- ricercaSulWebButton: QPushButton
- inserisciManualmenteButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getRicercaSulWebButton(): QPushButton
+ getInserisciManualmenteButton(): QPushButton
Controller
ControllerBackups
- model: Museo
- previous: Controller
- gotoPrevious(): void
- onBackupOraClicked(): void
+ connettiEventi(): void
+ initializeUi(): void
- renderizzaBackups(): ControllerWidgetBackup []
«Create»
+ create(view: VistaBackups, previous: Controller, model: Museo)
QMainWindow
VistaBackups
- previousButton: QPushButton
- backupsListView: QListView
- backupOraButton: QPushButton
«property get»
+ getPreviousButton(): QPushButton
+ getBackupsListView(): QListView
+ getBackupOraButton(): QPushButton
1
0..*

alla possibilità da parte degli stakeholders di scegliere combinatoriamente le statistiche.