CRC CARDS

# 1. Entità

## Dipendente

|  |  |  |
| --- | --- | --- |
| **Dipendente** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| - Contiene tutte le informazioni anagrafiche del dipendente  -Contiene tutte le informazioni aziendali del dipendente (salario, valutazione e credenziali di accesso)  -Tiene traccia dei meeting, progetti e skill associati al dipendente  - Si occupa di generare il codice fiscale di un dipendente a partire dai suoi dati anagrafici | | -LuogoNascita  -Skill  -PartecipazioneMeeting  -CollaborazioneProgetto |

## Meeting

|  |  |  |
| --- | --- | --- |
| **Meeting** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene tutte le informazioni necessarie per rappresentare un meeting (dataInizio, dataFine, orarioInizio,orarioFine,Piattaforma ecc..)  -Tiene traccia della sala runione in cui viene ospitato il meeting, nel caso questo sia fisico  -Tiene traccia del progetto relativo che sarà discusso nel meeting  -Tiene traccia di tutti i partecipanti al meeting | | -SalaRiunione  -Progetto  -PartecipazioneMeeting |

|  |  |  |
| --- | --- | --- |
| **Progetto** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene tutte le informazioni necessarie per rappresentare un progetto (nomeProgetto, descrizioneProgetto, dataCreazione ecc..)  -Tiene traccia degli ambiti relativi al progetto  -Tiene traccia dei meeting in cui si discute di questo progetto  -Tiene traccia dei dipendenti che collaborano al progetto con un determinato ruolo | | -AmbitoProgetto  -Meeting  -CollaborazioneProgetto |

## Progetto

## LuogoNascita

|  |  |  |
| --- | --- | --- |
| **LuogoNascita** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Rappresenta il luogo di nascita di un dipendente attraverso il nome , la provincia ed il codice del comune | | -Dipendente |

## Skill

|  |  |  |
| --- | --- | --- |
| **Skill** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Rappresenta le skill lavorative associate ad un dipendente | | -Dipendente |

## SalaRunione

|  |  |  |
| --- | --- | --- |
| **SalaRiunione** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene tutte le informazioni per rappresentare le sale riunioni aziendali , che potranno essere utlizzate per ospitare dei meeting fisici (Codice sala, indirizzo, piano e capienza)  -Tiene traccia dei meeting che vengono ospitati nella sala | | -Meeting |

## PartecipazioneMeeting

|  |  |  |
| --- | --- | --- |
| **PartecipazioneMeeting** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Si occupa di mantenere le informazioni relative alla partecipazione di un dipendente ad un meeting  -Tiene traccia della presenza del partecipante al relativo meeting  -Specifica se il partecipante assuma il ruolo di organizzatore o meno del meeting | | -Meeting  -Dipendente |

|  |  |  |
| --- | --- | --- |
| **CollaborazioneProgetto** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Si occupa di mantenere le informazione relative ai collaboratori in un progetto  -Tiene traccia del ruolo che un dipendente può assumere all’interno di un progetto (Project Manger, Team member..) | | -Progetto  -Dipendente |

## CollaborazioneProgetto

## AmbitoProgetto

|  |  |  |
| --- | --- | --- |
| **AmbitoProgetto** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Rappresenta gli ambiti che un progetto può interessare  -Tiene traccia di tutti i progetti che hanno un particolare ambito | | -Progetto |

# 2. Controller

## ControllerScelta

|  |  |  |
| --- | --- | --- |
| **ControllerScelta** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È il primo controller che viene inizializzato all avvio del programma , ha il compito di mostrare la finestra iniziale e di indirizzare l’utente verso la finestra desiderata in base alla scelta fatta (finestra di accesso o finestra della segreteria) | | -Iplanner (GUI)  -ControllerAccesso  -ControllerDipendentiSegreteria  -Tutti i DAO |

## ControllerAccesso

|  |  |  |
| --- | --- | --- |
| **ControllerAccesso** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Si occupa di gestire l accesso del dipendente al sistema e di memorizzarlo  -Nel caso in cui le credenziali inserite siano giuste richiama ControllerGestioneProfilo che mostra la prossima finestra  -Nel caso in cui l utente decida di annullare il login richiama ControllerScelta che mostra di nuovo la prima schermata | | -Login (GUI)  -Dipendente  -ControllerGestioneProfilo  -ControllerScelta  -Tutti i DAO |

## ControllerGestioneProfilo

|  |  |  |
| --- | --- | --- |
| **ControllerGestioneProfilo** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Si occupa di gestire la Home page e la finestra del proprio account  -Ricava le informazioni per popolare le liste di progetti e meeting relativi al dipendente  -Nella schermata Il mio account consente di aggiornare le info del dipendente nel database  -Nel caso in cui l utente prema sui bottoni “Miei Progetti” oppure ”Miei Meeting” richiama i rispettivi ControllerProgetto e ControllerMeeting, che mostreranno le relative finestre | | -Home (GUI)  -MioAccount (GUI)  -Dipendente  -CollaborazioneProgetto  -Meeting  -ControllerProgetto  -ControllerMeeting  -Tutti i DAO |

## ControllerProgetto

|  |  |  |
| --- | --- | --- |
| **ControllerProgetto** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Mostra le schermate Miei Progetti e Gestione Progetti in cui un dipendente puo visualizzare, modificare e creare progetti di cui è il Project Manager  -Ha una serie di metodi per ricavare le informazioni di un progetto(getPartecipanti , getMeetingRelativi ecc…)  -Ha una serie di metodi per modificare, aggiungere e rimuovere progetti dal database | | -MieiProgetti (GUI)  -GestioneProgettiDipendente (GUI)  -Dipendente  -CollaborazioneProgetto  -AmbitoProgetto  -Meeting  -Tutti i DAO |

## ControllerMeeting

|  |  |  |
| --- | --- | --- |
| **ControllerMeeting** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Mostra le schermate Miei Meeting e Gestione Meeting in cui un dipendente puo visualizzare, modificare e creare meeting  -Gestisce la schermata di inserimento dei partecipanti in un meeting  -Ha una serie di metodi per ricavare le informazioni di un meeting (ottieniSale, ottieniPiattaforme, ottieniInvitati ecc…)  -Ha una serie di metodi per modificare, aggiungere e rimuovere meeting dal database | | -MieiMeeting (GUI)  -GestioneMeetingDipendente (GUI)  -InserisciPartecipantiMeeting (GUI)  -Dipendente  -Meeting  -SalaRiunione  -Skill  -Tutti i DAO |

## ControllerDipendentiSegreteria

|  |  |  |
| --- | --- | --- |
| **ControllerDipendentiSegreteria** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Si occupa di gestire le finestre e le funzionalità relative alla segreteria  -Mostra la finestra di gestione dipendenti oppure torna indietro alla finestra iniziale richiamando ControllerScelta  -Consente di creare un nuovo account ad un dipendente inserendolo nel database  -Consente di creare nuove skills da associare ai dipendenti | | -iPlanner (GUI)  -GestioneDipendenti (GUI)  -ControllerScelta  -LuogoNascita  -Skill  -Tutti i DAO |

## ControllerPartecipantiProgetto

|  |  |  |
| --- | --- | --- |
| **ControllerPartecipantiProgetto** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Si occupa di gestire la finestra InserisciPartecipantiProgetto  -Ha una serie di metodi per ottenere le info dei partecipanti come ottieniRuoli(), ottieniSkillDipendente()…  -Ha dei metodi per eliminare ed inserire dei partecipanti nel progetto | | -InserisciPartecipantiProgetto (GUI)  -Dipendente  -Progetto  -Skill  -Tutti i DAO |

## ControllerPartecipantiMeeting

|  |  |  |
| --- | --- | --- |
| **ControllerPartecipantiMeeting** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Si occupa di gestire la finestra InserisciPartecipantiMeeting  -Ha una serie di metodi per ottenere le info dei partecipanti come ottieniInvitati()…  -Ha dei metodi per eliminare ed inserire dei partecipanti nel meeting | | -InserisciPartecipantiMeeting (GUI)  -Dipendente  -Meeting  -Skill  -Tutti i DAO |

# 3. GUI

## iPlanner

|  |  |  |
| --- | --- | --- |
| **iPlanner** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la prima finestra che si apre all avvio del programma e cambia in base all argomento in input nel main (-d = dipendente , -s = segreteria)  -Nel caso venga inserito -d la finestra mostra solo il bottone per fare il login  -Nel caso venga inserito -s la finestra mostra tre bottoni per accedere alla gestione di dipendenti, meeting, oppure dei progetti | | -ControllerScelta |

## Login

|  |  |  |
| --- | --- | --- |
| **Login** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la finestra in cui il dipendente andrà ad inserire i dati per fare l’accesso  -Ha un metodo login() che , oltre a richiamare verificaCredenziali() nel controller, si occupa di evidenziare in rosso i campi non inseriti e di stampare eventuali messaggi di errore  -Ha un metodo svuotaCampi() che viene chiamato nel caso il dipendente sbagli ad inserire le credenziali | | -ControllerAccesso |

## Home

|  |  |  |
| --- | --- | --- |
| **Home** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la home page dell’intero programma in cui il dipendente ha una panoramica di tutti i progetti e meeting che lo riguardano  -Da questa schermata, tramite i bottoni in alto a destra, è possibile passare alle finestre relative all’account , ai meeting ed ai progetti | | -ControllerAccesso  -Dipendente  -Meeting  -Progetto |

## MioAccount

|  |  |  |
| --- | --- | --- |
| **MioAccount** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la finestra in cui il dipendente può visualizzare le sue informazioni personali ed aziendali (valutazione ,salario e skill)  -Il dipendente può anche modificare le sue info personali , mentre quelle aziendali sono modificabili solo dalla segreteria  -Ha un metodo updateAccount() che va a ricavare tutte le informazioni dai campi e con queste richiama il metodo aggiornaInfoDipendente() nel controller | | -ControllerGestioneProfilo  -Dipendente  -Skill  -LuogoNascita |

## MieiProgetti

|  |  |  |
| --- | --- | --- |
| **MieiProgetti** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la finestra in cui il dipendente può visualizzare nel dettaglio le informazioni dei suoi progetti  -Premendo sul tasto Inserisci/Modifica Progetto il dipendente verrà indirizzato nella finestra di gestione progetti | | -ControllerProgetto  -Progetto  -AmbitoProgetto |

## MieiMeeting

|  |  |  |
| --- | --- | --- |
| **MieiMeeting** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la finestra in cui il dipendente può visualizzare nel dettaglio le informazioni dei suoi meeting  --Premendo sul tasto Inserisci/Modifica Meeting il dipendente verrà indirizzato nella finestra di gestione meeting | | -ControllerMeeting  -Meeting  -MeetingListRenderer |

## GestioneProgettiDipendente

|  |  |  |
| --- | --- | --- |
| **GestioneProgettiDipendente** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la finestra in cui il dipendente può gestire i progetti di cui è il project manager  -Tutti i progetti sono organizzati in una tabella in cui vengono mostrate le info  -Qui si potranno modificare tutte le info di un progetto, creare nuovi progetti ed eliminare completamente progetti  -Premendo sul bottone Inserisci partecipanti il dipendente verrà indirizzato alla finestra InserisciPartecipantiProgetto | | -ControllerProgetto  -ProgettoTableModel  -Progetto  -AmbitoProgetto  -MeetingListRenderer  -Dipendente  -Meeting |

## GestisciMeetingDipendente

|  |  |  |
| --- | --- | --- |
| **GestisciMeetingDipendente** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la finestra in cui il dipendente può gestire i propri meeting  -Tutti i meeting sono organizzati in una tabella in cui vengono mostrate le info  -Qui si potranno modificare tutte le info di un meeting, creare nuovi meeting ed eliminare completamente dei meeting  -Premendo sul bottone Inserisci partecipanti il dipendente verrà indirizzato alla finestra InserisciPartecipantiMeeting | | -ControllerMeeting  -MeetingTableModel  -SalaRiunione  -Meeting  -Progetto |

## InserisciPartecipantiProgetto

|  |  |  |
| --- | --- | --- |
| **InserisciPartecipantiProgetto** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la finestra in cui un dipendente può aggiungere ed eliminare partecipanti ad un progetto di cui è il project manager  -Nella parte superiore si trovano tutte le info utili del dipendente (salario,valutazione, skill…) e del progetto, mentre nella parte inferiore vengono mostrati tutti i dipendenti in una tabella | | -ControllerPartecipantiProgetto  -Progetto  -PartecipantiTableModel  -Dipendente |

## InserisciPartecipantiMeeting

|  |  |  |
| --- | --- | --- |
| **InserisciPartecipantiMeeting** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la finestra in cui un dipendente può aggiungere ed eliminare partecipanti ad un meeting  --Nella parte superiore si trovano tutte le info utili del dipendente (salario,valutazione, skill…) e del meeting, mentre nella parte inferiore vengono mostrati tutti i dipendenti in una tabella | | -ControllerPartecipantiMeeting  -Dipendente  -Meeting  -PartecipantiTableModel |

## GestioneDipendenti

|  |  |  |
| --- | --- | --- |
| **GestioneDipendenti** | | |
| Superclassi: | JFrame | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -In questa finestra la segreteria può gestire tutte le informazioni legate ai dipendenti  -Si possono aggiungere nuovi dipendenti ed eliminare dipendenti gia presenti  -Si possono modificare tutte le info del dipendente comprese quelle aziendali non modificabili dal dipendente stesso  -È possibile inoltre filtrare i dipendenti in base a dei criteri come salario , valutazione ed età | | -ControllerDipendentiSegreteria  -Dipendente  -LuogoNascita  -Skill  -DipendentiTableModel |

## MeetingListRenderer

|  |  |  |
| --- | --- | --- |
| **MeetingListRenderer** | | |
| Superclassi: | Object , IMPLEMENTAZIONI : ListCellRender (Interfaccia) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Ha lo scopo di far visualizzare i meeting all’interno delle JList in maniera corretta  -Fa apparire i meeting di colore rosso quando questi sono scaduti | | -Meeting |

## DipendenteInvitatoListRenderer

|  |  |  |
| --- | --- | --- |
| **DipendenteInvitatoListRenderer** | | |
| Superclassi: | Object , IMPLEMENTAZIONI :ListCellRender (Interfaccia) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Ha lo scopo di far visualizzare correttamente i dipendenti invitati ad un meeting all’interno delle JList | | -Dipendente |

## DipendentiTableModel

|  |  |  |
| --- | --- | --- |
| **DipendentiTableModel** | | |
| Superclassi: | AbstractTableModel (Classe astratta) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Ha lo scopo di gestire i dati contenuti nella tabella dei dipendenti  -Implementa una serie di metodi ereditati da AbstractTableModel che però sono compatibili con il tipo Dipendente , ad esempio il metodo getSelected() ritorna la riga selezionata sotto forma di un oggetto Dipendente | | -Dipendente |

## MeetingTableModel

|  |  |  |
| --- | --- | --- |
| **MeetingTableModel** | | |
| Superclassi: | AbstractTableModel (Classe astratta) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Ha lo scopo di gestire i dati contenuti nella tabella dei meeting  -Implementa una serie di metodi ereditati da AbstractTableModel che però sono compatibili con il tipo Meeting, ad esempio il metodo getValueAt(int colonna,int riga) ritorna un singolo campo del meeting in base alla cella data in input | | -Meeting |

## PartecipantiTableModel

|  |  |  |
| --- | --- | --- |
| **PartecipantiTableModel** | | |
| Superclassi: | AbstractTableModel (Classe astratta) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Ha lo scopo di gestire i dati contenuti nella tabella dei partecipanti di meeting e progetti  -Implementa una serie di metodi ereditati da AbstractTableModel che però sono compatibili con il tipo Dipendente | | -Dipendente |

## ProgettoTableModel

|  |  |  |
| --- | --- | --- |
| **ProgettoTableModel** | | |
| Superclassi: | AbstractTableModel (Classe astratta) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Ha lo scopo di gestire i dati contenuti nella tabella dei progetti  -Implementa una serie di metodi ereditati da AbstractTableModel che però sono compatibili con il tipo Progetto | | -Progetto |

# 4. InterfacceDAO

## DipendenteDAO

|  |  |  |
| --- | --- | --- |
| **DipendenteDAO** | | |
| Superclassi: |  | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene le signature dei metodi che andranno ad effettuare operazioni sui dipendenti nel database  -Ha lo scopo di “forzare “ l’inserimento di questi metodi nelle classi che andranno ad implementarla  -Viene utilizzata come tipo astratto indipendetemente dal modo in cui vengono inizializzati i DAO (ad esempio DipendenteDAO può essere sia un DipendenteDAOPSQL che un ipotetico DipendenteDAOORACLE) | | -Dipendente  -Skill |

## ProgettoDAO

|  |  |  |
| --- | --- | --- |
| **ProgettoDAO** | | |
| Superclassi: |  | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene le signature dei metodi che andranno ad effettuare operazioni sui progetti nel database  -Ha lo scopo di “forzare “ l’inserimento di questi metodi nelle classi che andranno ad implementarla  -Viene utilizzata come tipo astratto indipendetemente dal modo in cui vengono inizializzati i DAO (ad esempio ProgettoDAO può essere sia un ProgettoDAOPSQL che un ipotetico ProgettoDAOORACLE) | | -AmbitoProgetto  -CollaborazioneProgetto  -Dipendente  -Meeting  -Progetto |

## MeetingDAO

|  |  |  |
| --- | --- | --- |
| **MeetingDAO** | | |
| Superclassi: |  | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene le signature dei metodi che andranno ad effettuare operazioni sui meeting nel database  -Ha lo scopo di “forzare “ l’inserimento di questi metodi nelle classi che andranno ad implementarla  -Viene utilizzata come tipo astratto indipendetemente dal modo in cui vengono inizializzati i DAO (ad esempio MeetingDAO può essere sia un MeetingDAOPSQL che un ipotetico MeetingDAOORACLE) | | -Dipendente  -Meeting  -Progetto  -SalaRiunione |

## AmbitoProgettoDAO

|  |  |  |
| --- | --- | --- |
| **AmbitoProgettoDAO** | | |
| Superclassi: |  | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene le signature dei metodi che andranno ad effettuare operazioni sugli Ambiti nel database  -Ha lo scopo di “forzare “ l’inserimento di questi metodi nelle classi che andranno ad implementarla  -Viene utilizzata come tipo astratto indipendetemente dal modo in cui vengono inizializzati i DAO (ad esempio AmbitoProgettoDAO può essere sia un AmbitoProgettoDAOPSQL che un ipotetico AmbitoProgettoDAOORACLE) | | -AmbitoProgetto  -Progetto |

## LuogoNascitaDAO

|  |  |  |
| --- | --- | --- |
| **LuogoNascitaDAO** | | |
| Superclassi: |  | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene le signature dei metodi che andranno ad ottenere informazioni sui Luogi di nascita dei dipendenti nel database  -Ha lo scopo di “forzare “ l’inserimento di questi metodi nelle classi che andranno ad implementarla  -Viene utilizzata come tipo astratto indipendetemente dal modo in cui vengono inizializzati i DAO (ad esempio LuogoNascitaDAO può essere sia un LuogoNascitaDAOPSQL che un ipotetico LuogoNascitaDAOORACLE) | | -Dipendente  -LuogoNascita |

## SalaRiunioneDAO

|  |  |  |
| --- | --- | --- |
| **SalaRiunioneDAO** | | |
| Superclassi: |  | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene le signature dei metodi che andranno ad effettuare operazioni sulle Sale riunioni nel database  -Ha lo scopo di “forzare “ l’inserimento di questi metodi nelle classi che andranno ad implementarla  -Viene utilizzata come tipo astratto indipendetemente dal modo in cui vengono inizializzati i DAO (ad esempio SalaRiunioneDAO può essere sia un SalaRiunioneDAOPSQL che un ipotetico SalaRiunioneDAOORACLE) | | -SalaRiunione |

## SkillDAO

|  |  |  |
| --- | --- | --- |
| **SkillDAO** | | |
| Superclassi: |  | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene le signature dei metodi che andranno ad effettuare operazioni sulle skill nel database  -Ha lo scopo di “forzare “ l’inserimento di questi metodi nelle classi che andranno ad implementarla  -Viene utilizzata come tipo astratto indipendetemente dal modo in cui vengono inizializzati i DAO (ad esempio SkillDAO può essere sia un SkillDAOPSQL che un ipotetico SkillDAOORACLE) | | -Dipendente  -Skill |

# 5. ImplementazioniDAO

## AmbitoProgettoDAOPSQL

|  |  |  |
| --- | --- | --- |
| **AmbitoProgettoDAOPSQL** | | |
| Superclassi: | Object , IMPLEMENTAZIONI: AmbitoProgettoDAO (interfaccia) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene tutte le implementazioni dei metodi presenti in AmbitoProgettoDAO per postgreSQL  -Ha il compito di fare da “tramite” tra il database e le altre classi (non DAO), trasformando il risultato delle query in tipi compatibili con java  -Ha una serie di PreparedStatement che permettono di parametrizzare le query oltre che portare vantaggi in termini di efficienza sul database | | -AmbitoProgetto  -Progetto |

## DipendenteDAOPSQL

|  |  |  |
| --- | --- | --- |
| **DipendenteDAOPSQL** | | |
| Superclassi: | Object , IMPLEMENTAZIONI: DipendenteDAO (interfaccia) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene tutte le implementazioni dei metodi presenti in DipendenteDAO per postgreSQL  -Ha il compito di fare da “tramite” tra il database e le altre classi (non DAO), trasformando il risultato delle query in tipi compatibili con java  -Ha una serie di PreparedStatement che permettono di parametrizzare le query oltre che portare vantaggi in termini di efficienza sul database | | -Dipendente  -LuogoNascita  -Skill |

## LuogoNascitaDAOPSQL

|  |  |  |
| --- | --- | --- |
| **LuogoNascitaDAOPSQL** | | |
| Superclassi: | Object , IMPLEMENTAZIONI: LuogoNascitaDAO (interfaccia) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene tutte le implementazioni dei metodi presenti in LuogoNascitaDAO per postgreSQL  -Ha il compito di fare da “tramite” tra il database e le altre classi (non DAO), trasformando il risultato delle query in tipi compatibili con java  -Ha una serie di PreparedStatement che permettono di parametrizzare le query oltre che portare vantaggi in termini di efficienza sul database | | -Dipendente  -LuogoNascita  -DipendenteDAO |

## MeetingDAOPSQL

|  |  |  |
| --- | --- | --- |
| **MeetingDAOPSQL** | | |
| Superclassi: | Object , IMPLEMENTAZIONI: MeetingDAO (interfaccia) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene tutte le implementazioni dei metodi presenti in MeetingDAO per postgreSQL  -Ha il compito di fare da “tramite” tra il database e le altre classi (non DAO), trasformando il risultato delle query in tipi compatibili con java  -Ha una serie di PreparedStatement che permettono di parametrizzare le query oltre che portare vantaggi in termini di efficienza sul database | | -Dipendente  -LuogoNascita  -Meeting  -Progetto  -DipendenteDAO  -LuogoNascitaDAO  -ProgettoDAO  -SalaRiunioneDAO |

## ProgettoDAOPSQL

|  |  |  |
| --- | --- | --- |
| **ProgettoDAOPSQL** | | |
| Superclassi: | Object , IMPLEMENTAZIONI: ProgettoDAO (interfaccia) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene tutte le implementazioni dei metodi presenti in ProgettoDAO per postgreSQL  -Ha il compito di fare da “tramite” tra il database e le altre classi (non DAO), trasformando il risultato delle query in tipi compatibili con java  -Ha una serie di PreparedStatement che permettono di parametrizzare le query oltre che portare vantaggi in termini di efficienza sul database | | -Dipendente  -AmbitoProgetto  -Meeting  -Progetto  -CollaborazioneProgetto  -DipendenteDAO  -AmbitoProgettoDAO  -ProgettoDAO  -SalaRiunioneDAO |

## SalaRiunioneDAOPSQL

|  |  |  |
| --- | --- | --- |
| **SalaRiunioneDAOPSQL** | | |
| Superclassi: | Object , IMPLEMENTAZIONI: SalaRiunioneDAO (interfaccia) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene tutte le implementazioni dei metodi presenti in SalaRiunioneDAO per postgreSQL  -Ha il compito di fare da “tramite” tra il database e le altre classi (non DAO), trasformando il risultato delle query in tipi compatibili con java  -Ha una serie di PreparedStatement che permettono di parametrizzare le query oltre che portare vantaggi in termini di efficienza sul database | | -SalaRiunione |

## SkillDAOPSQL

|  |  |  |
| --- | --- | --- |
| **SkillDAOPSQL** | | |
| Superclassi: | Object , IMPLEMENTAZIONI: SkillDAO (interfaccia) | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene tutte le implementazioni dei metodi presenti in SkillDAO per postgreSQL  -Ha il compito di fare da “tramite” tra il database e le altre classi (non DAO), trasformando il risultato delle query in tipi compatibili con java  -Ha una serie di PreparedStatement che permettono di parametrizzare le query oltre che portare vantaggi in termini di efficienza sul database | | -Dipendente  -Skill |

# 6. dbManager

## ManagerConnessioneDB

|  |  |  |
| --- | --- | --- |
| **ManagerConnessioneDB** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Ha il compito di aprire e chiudere la connessione con il database  -Ha una serie di attributi che conservano le proprietà del database cioè username, password, nome ed url  -Ha un metodo getInstance() che restituisce la singola istanza del ManagerConnessioneDB richiamando il costruttore privato  -Ha un metodo creaDatabase() che va a creare il database nel caso questo non esista già  -Ha un metodo getConnection() che restituisce semplicemente la connessione creata nel costruttore privato | |  |

## CostruttoreDB

|  |  |  |
| --- | --- | --- |
| **CostruttoreDB** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -Contiene la definizione delle tabelle, dei vincoli, dei trigger, e di tutto ciò che è necessario per inizializzare il database prima dell’utilizzo del software  -Ha tutta una serie di metodi per andare a creare le singole tabelle ,le funzioni ed i trigger  -Ha un metodo generale creaTabelle() che richiama tutti gli altri metodi e va a creare tutte le tabelle  -Ha un metodo generale creaFunzioniTrigger() che va a richiamare tutti gli altri metodi per creare tutte le funzioni esterne  -Ha un metodo importaLuoghi() che va ad importare nella tabella LuogoNascita tutti i comuni italiani (con nome ,provincia e codice) da un file CSV esterno | |  |

# Starter

|  |  |  |
| --- | --- | --- |
| **Starter** | | |
| Superclassi: | Object | |
| Sottoclassi: |  | |
| Responsabilità | | Collaboratori |
| -È la classe che contiene il main che quindi fa partire l’intero programma  -Si connette al database e nel caso questo non esista lo crea, con tutte le tabelle e funzioni  -Inizializza tutti i DAO per postgreSQL  -Ottiene l’autorizzazione (-s segreteria o -d dipendente) e la passa a ControllerScelta che, in relazione a questa, andrà a mostrare la prima finestra | | -ControllerScelta  -CostruttoreDB  -ManagerConnessioneDB  -tutti i DAO |