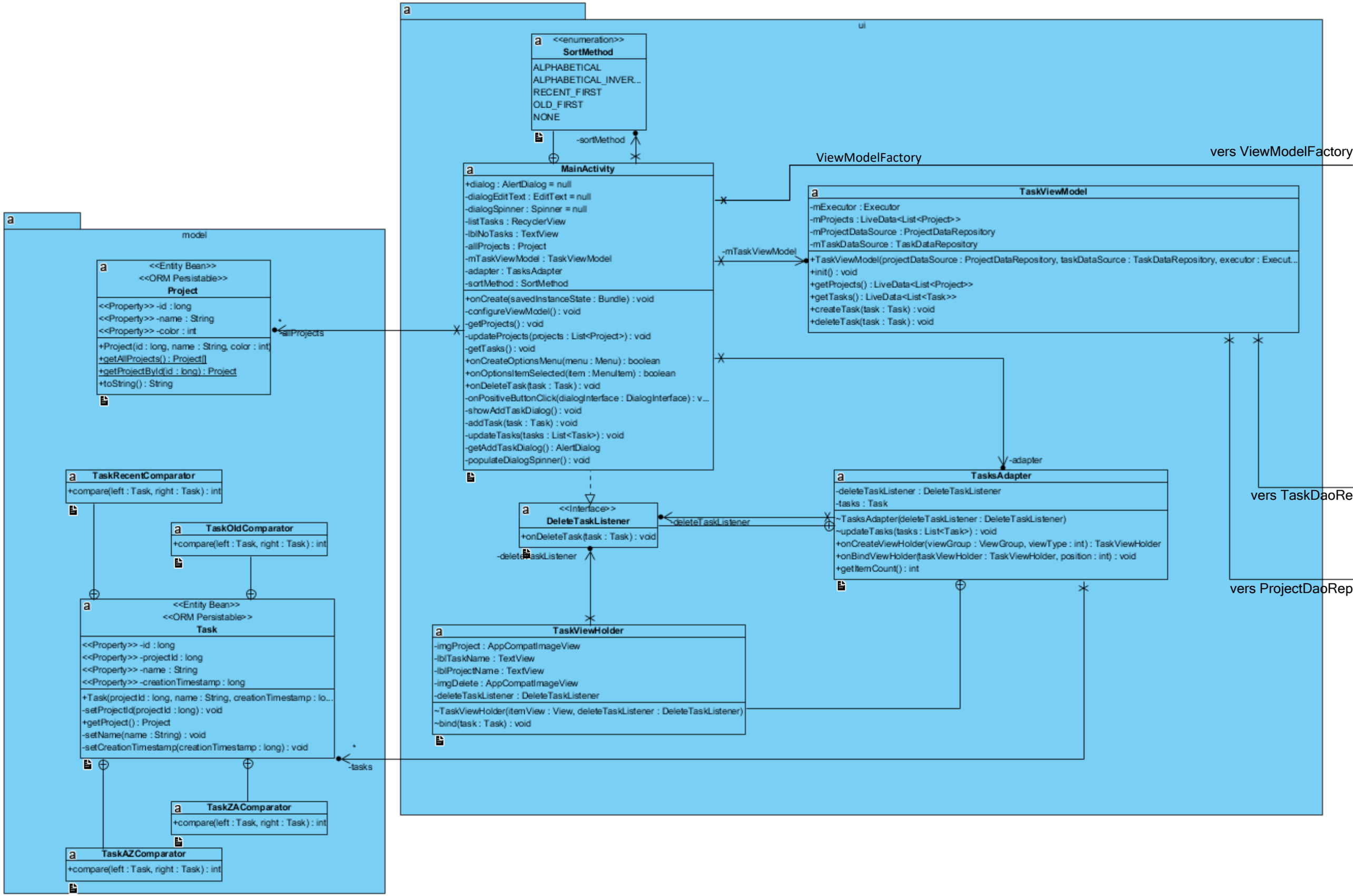
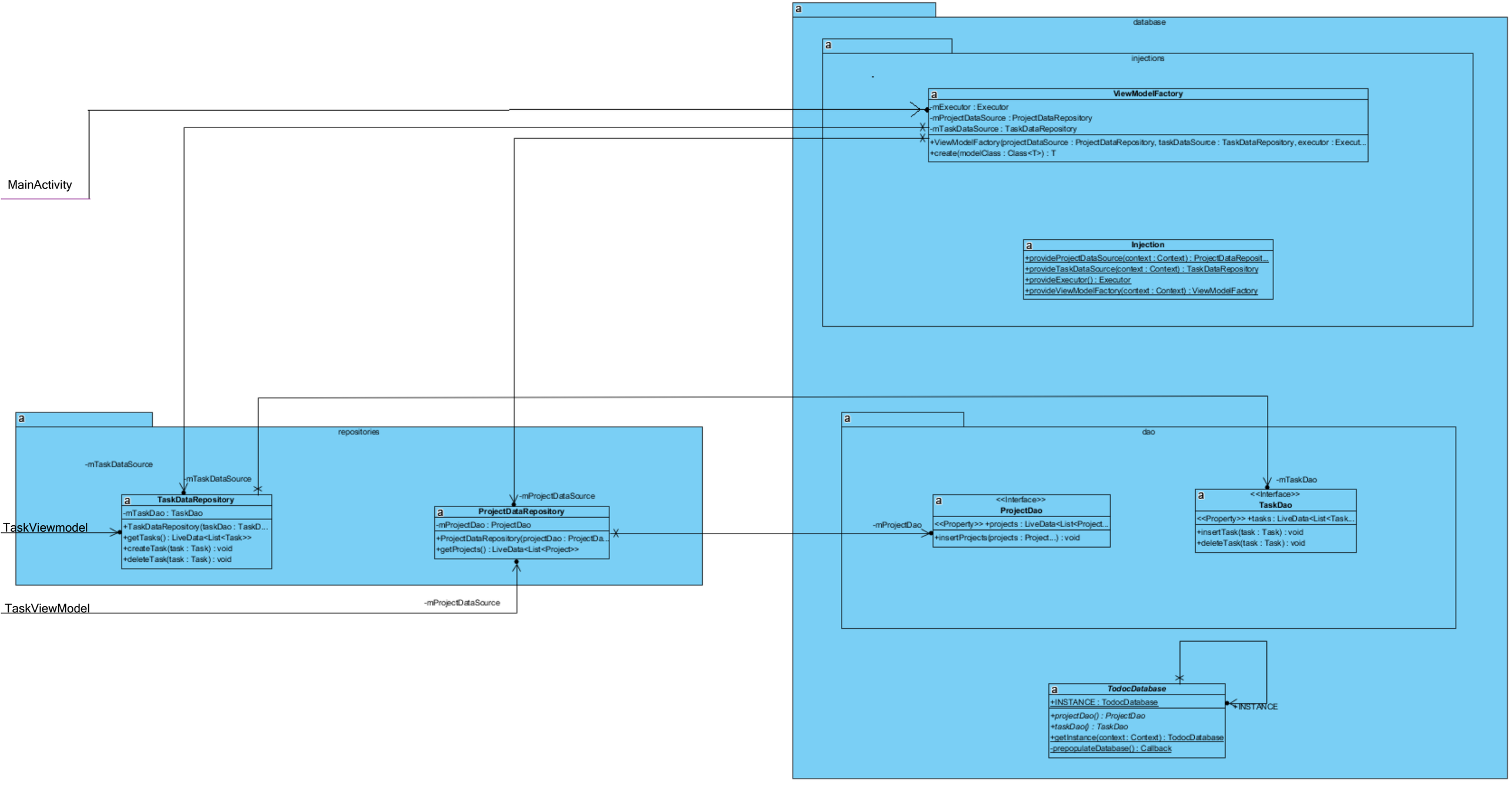

ui
<<enumeration>>
SortMethod
ALPHABETICAL
ALPHABETICAL_INVER...
RECENT_FIRST
OLD_FIRST
NONE
-sortMethod
MainActivity
+dialog : AlertDialog = null
-dialogEditText : EditText = null
-dialogSpinner : Spinner = null
-listTasks : RecyclerView
-lblNoTasks : TextView
-allProjects : Project
-mTaskViewModel : TaskViewModel
-adapter : TasksAdapter
-sortMethod : SortMethod
+onCreate(savedInstanceState : Bundle) : void
-configureViewModel() : void
-getProjects() : void
-updateProjects(projects : List<Project>) : void
-getTasks() : void
+onCreateOptionsMenu(menu : Menu) : boolean
+onOptionsItemSelected(item : MenuItem) : boolean
+onDeleteTask(task : Task) : void
-onPositiveButtonClick(dialogInterface : DialogInterface) : v...
-showAddTaskDialog() : void
-addTask(task : Task) : void
-updateTasks(tasks : List<Task>) : void
-getAddTaskDialog() : AlertDialog
-populateDialogSpinner() : void
ViewModelFactory
vers ViewModelFactory
TaskViewModel
-mExecutor : Executor
-mProjects : LiveData<List<Project>>
-mProjectDataSource : ProjectDataRepository
-mTaskDataSource : TaskDataRepository
+TaskViewModel(projectDataSource : ProjectDataRepository, taskDataSource : TaskDataRepository, executor : Execut...
+init() : void
+getProjects() : LiveData<List<Project>>
+getTasks() : LiveData<List<Task>>
+createTask(task : Task) : void
+deleteTask(task : Task) : void
-mTaskViewModel
-adapter
TasksAdapter
-deleteTaskListener : DeleteTaskListener
-tasks : Task
~TasksAdapter(deleteTaskListener : DeleteTaskListener)
~updateTasks(tasks : List<Task>) : void
+onCreateViewHolder(viewGroup : ViewGroup, viewType : int) : TaskViewHolder
+onBindViewHolder(taskViewHolder : TaskViewHolder, position : int) : void
+getItemCount() : int
vers TaskDaoRe
vers ProjectDaoRep
<<Interface>>
DeleteTaskListener
+onDeleteTask(task : Task) : void
-deleteTaskListener
-deleteTaskListener
TaskViewHolder
-imgProject : AppCompatImageView
-lblTaskName : TextView
-lblProjectName : TextView
-imgDelete : AppCompatImageView
-deleteTaskListener : DeleteTaskListener
~TaskViewHolder(itemView : View, deleteTaskListener : DeleteTaskListener)
~bind(task : Task) : void
model
<<Entity Bean>>
<<ORM Persistable>>
Project
<<Property>> -id : long
<<Property>> -name : String
<<Property>> -color : int
+Project(id : long, name : String, color : int)
+getAllProjects() : Project[]
+getProjectById(id : long) : Project
+toString() : String
-allProjects
TaskRecentComparator
+compare(left : Task, right : Task) : int
TaskOldComparator
+compare(left : Task, right : Task) : int
<<Entity Bean>>
<<ORM Persistable>>
Task
<<Property>> -id : long
<<Property>> -projectId : long
<<Property>> -name : String
<<Property>> -creationTimestamp : long
+Task(projectId : long, name : String, creationTimestamp : lo...
-setProjectId(projectId : long) : void
+getProject() : Project
-setName(name : String) : void
-setCreationTimestamp(creationTimestamp : long) : void
-tasks
TaskZAComparator
+compare(left : Task, right : Task) : int
TaskAZComparator
+compare(left : Task, right : Task) : int


database
injections
ViewModelFactory
-mExecutor : Executor
-mProjectDataSource : ProjectDataRepository
-mTaskDataSource : TaskDataRepository
+ViewModelFactory(projectDataSource : ProjectDataRepository, taskDataSource : TaskDataRepository, executor : Execut...
+create(modelClass : Class<T>) : T
Injection
+provideProjectDataSource(context : Context) : ProjectDataReposit...
+provideTaskDataSource(context : Context) : TaskDataRepository
+provideExecutor() : Executor
+provideViewModelFactory(context : Context) : ViewModelFactory
MainActivity
repositories
-mTaskDataSource
-mTaskDataSource
TaskDataRepository
-mTaskDao : TaskDao
+TaskDataRepository(taskDao : TaskD...
+getTasks() : LiveData<List<Task>>
+createTask(task : Task) : void
+deleteTask(task : Task) : void
-mProjectDataSource
ProjectDataRepository
-mProjectDao : ProjectDao
+ProjectDataRepository(projectDao : ProjectDa...
+getProjects() : LiveData<List<Project>>
TaskViewmodel
TaskViewModel
-mProjectDataSource
dao
-mProjectDao
<<Interface>>
ProjectDao
<<Property>> +projects : LiveData<List<Project...
+insertProjects(projects : Project...) : void
-mTaskDao
<<Interface>>
TaskDao
<<Property>> +tasks : LiveData<List<Task...
+insertTask(task : Task) : void
+deleteTask(task : Task) : void
TodocDatabase
+INSTANCE : TodocDatabase
+projectDao() : ProjectDao
+taskDao() : TaskDao
+getInstance(context : Context) : TodocDatabase
-prepopulateDatabase() : Callback
+INSTANCE