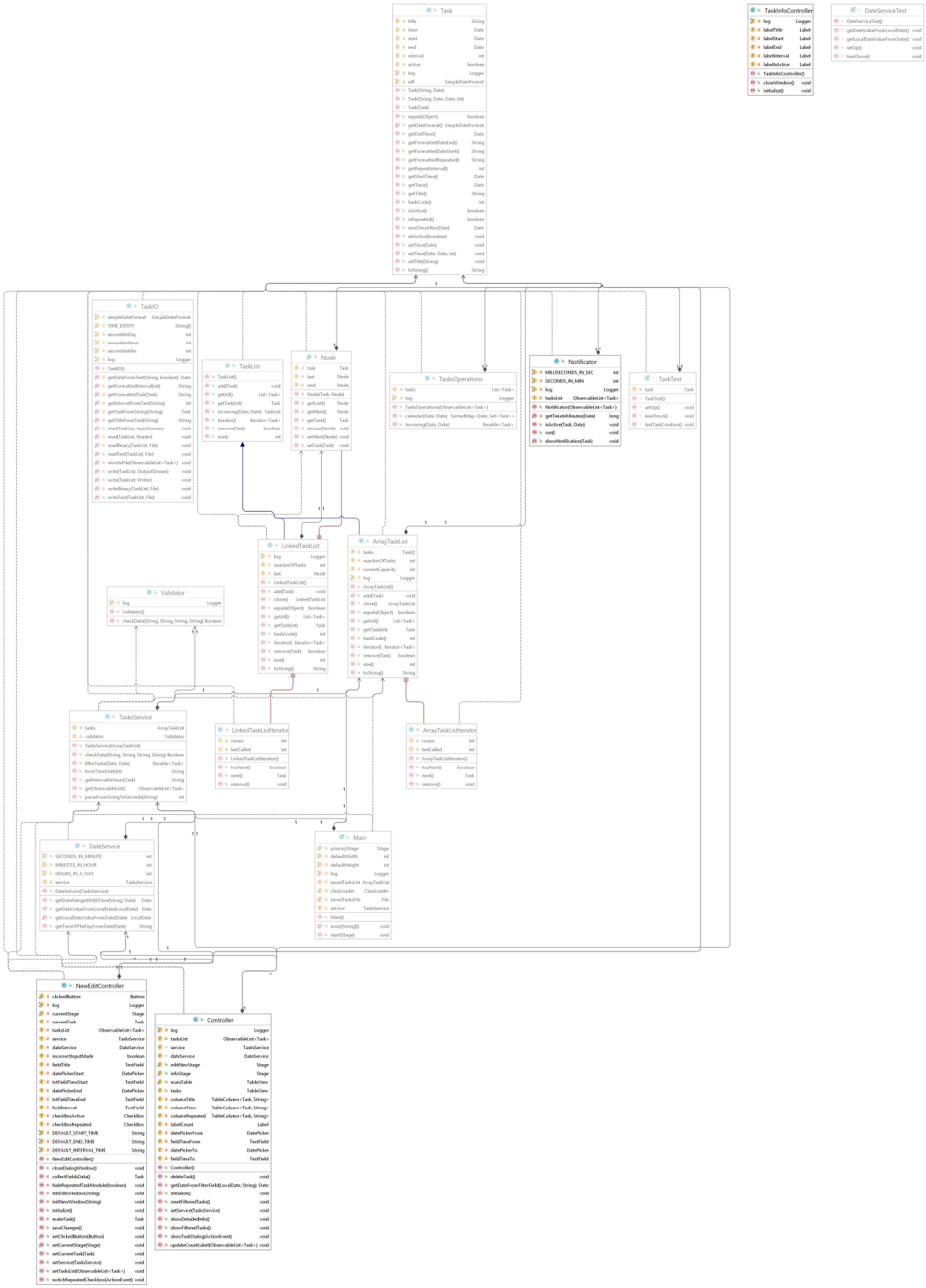
Task
title String
time Date
start Date
end Date
interval int
active boolean
log Logger
sdf SimpleDateFormat
Task(String, Date)
Task(String, Date, Date, int)
Task(Task)
equals(Object) boolean
getDateFormat() SimpleDateFormat
getEndTime() Date
getFormattedDateEnd() String
getFormattedDateStart() String
getFormattedRepeated() String
getRepeatInterval() int
getStartTime() Date
getTime() Date
getTitle() String
hashCode() int
isActive() boolean
isRepeated() boolean
nextTimeAfter(Date) Date
setActive(boolean) void
setTime(Date) void
setTime(Date, Date, int) void
setTitle(String) void
toString() String
TaskIO
simpleDateFormat SimpleDateFormat
TIME_ENTITY String[]
secondsInDay int
secondsInHour int
secondsInMin int
log Logger
TaskIO()
getDateFromText(String, boolean) Date
getFormattedInterval(int) String
getFormattedTask(Task) String
getIntervalFromText(String) int
getTaskFromString(String) Task
getTitleFromText(String) String
read(TaskList, InputStream) void
read(TaskList, Reader) void
readBinary(TaskList, File) void
readText(TaskList, File) void
rewriteFile(ObservableList<Task>) void
write(TaskList, OutputStream) void
write(TaskList, Writer) void
writeBinary(TaskList, File) void
writeText(TaskList, File) void
TaskList
TaskList()
add(Task) void
getAll() List<Task>
getTask(int) Task
incoming(Date, Date) TaskList
iterator() Iterator<Task>
remove(Task) boolean
size() int
Node
task Task
last Node
next Node
Node(Task, Node)
getLast() Node
getNext() Node
getTask() Task
setLast(Node) void
setNext(Node) void
setTask(Task) void
TasksOperations
tasks List<Task>
log Logger
TasksOperations(ObservableList<Task>)
calendar(Date, Date) SortedMap<Date, Set<Task>>
incoming(Date, Date) Iterable<Task>
Notificator
MILLISECONDS_IN_SEC int
SECONDS_IN_MIN int
log Logger
taskList ObservableList<Task>
Notificator(ObservableList<Task>)
getTimeInMinutes(Date) long
isActive(Task, Date) void
run() void
showNotification(Task) void
TaskTest
task Task
TaskTest()
setUp() void
tearDown() void
testTaskCreation() void
LinkedTaskList
log Logger
numberOfTasks int
last Node
LinkedTaskList()
add(Task) void
clone() LinkedTaskList
equals(Object) boolean
getAll() List<Task>
getTask(int) Task
hashCode() int
iterator() Iterator<Task>
remove(Task) boolean
size() int
toString() String
ArrayTaskList
tasks Task[]
numberOfTasks int
currentCapacity int
log Logger
ArrayTaskList()
add(Task) void
clone() ArrayTaskList
equals(Object) boolean
getAll() List<Task>
getTask(int) Task
hashCode() int
iterator() Iterator<Task>
remove(Task) boolean
size() int
toString() String
Validator
log Logger
Validator()
checkData(String, String, String, String) Boolean
TasksService
tasks ArrayTaskList
validator Validator
TasksService(ArrayTaskList)
checkData(String, String, String, String) Boolean
filterTasks(Date, Date) Iterable<Task>
formTimeUnit(int) String
getIntervalInHours(Task) String
getObservableList() ObservableList<Task>
parseFromStringToSeconds(String) int
LinkedTaskListIterator
cursor int
lastCalled int
LinkedTaskListIterator()
hasNext() boolean
next() Task
remove() void
ArrayTaskListIterator
cursor int
lastCalled int
ArrayTaskListIterator()
hasNext() boolean
next() Task
remove() void
DateService
SECONDS_IN_MINUTE int
MINUTES_IN_HOUR int
HOURS_IN_A_DAY int
service TasksService
DateService(TasksService)
getDateMergedWithTime(String, Date) Date
getDateValueFromLocalDate(LocalDate) Date
getLocalDateValueFromDate(Date) LocalDate
getTimeOfTheDayFromDate(Date) String
Main
primaryStage Stage
defaultWidth int
defaultHeight int
log Logger
savedTasksList ArrayTaskList
classLoader ClassLoader
savedTasksFile File
service TasksService
Main()
main(String[]) void
start(Stage) void
NewEditController
clickedButton Button
log Logger
currentStage Stage
currentTask Task
tasksList ObservableList<Task>
service TasksService
dateService DateService
incomeInputMade boolean
fieldTitle TextField
datePickerStart DatePicker
txtFieldTimeStart TextField
datePickerEnd DatePicker
txtFieldTimeEnd TextField
fieldInterval TextField
checkBoxActive CheckBox
checkBoxRepeated CheckBox
DEFAULT_START_TIME String
DEFAULT_END_TIME String
DEFAULT_INTERVAL_TIME String
NewEditController()
closeDialogWindow() void
collectFieldsData() Task
hideRepeatedTaskModule(boolean) void
initCtrlInWindow(String) void
initNewWindow(String) void
initialize() void
makeTask() Task
saveChanges() void
setClickedButton(Button) void
setCurrentStage(Stage) void
setCurrentTask(Task) void
setService(TasksService) void
setTasksList(Observable List<Task>) void
switchRepeatedCheckbox(ActionEvent) void
Controller
log Logger
tasksList ObservableList<Task>
service TasksService
dateService DateService
editNewStage Stage
infoStage Stage
mainTable TableView
tasks TableView
columnTitle TableColumn<Task, String>
columnTime TableColumn<Task, String>
columnRepeated TableColumn<Task, String>
labelCount Label
datePickerFrom DatePicker
fieldTimeFrom TextField
datePickerTo DatePicker
fieldTimeTo TextField
Controller()
deleteTask() void
getDateFromFilterField(LocalDate, String) Date
initialize() void
resetFilteredTasks() void
setService(TasksService) void
showDetailedInfo() void
showFilteredTasks() void
showTaskDialog(ActionEvent) void
updateCountLabel(ObservableList<Task>) void
TaskInfoController
log Logger
labelTitle Label
labelStart Label
labelEnd Label
labelInterval Label
labelIsActive Label
TaskInfoController()
closeWindow() void
initialize() void
DateServiceTest
DateServiceTest()
getDateValueFromLocalDate() void
getLocalDateValueFromDate() void
setUp() void
tearDown() void