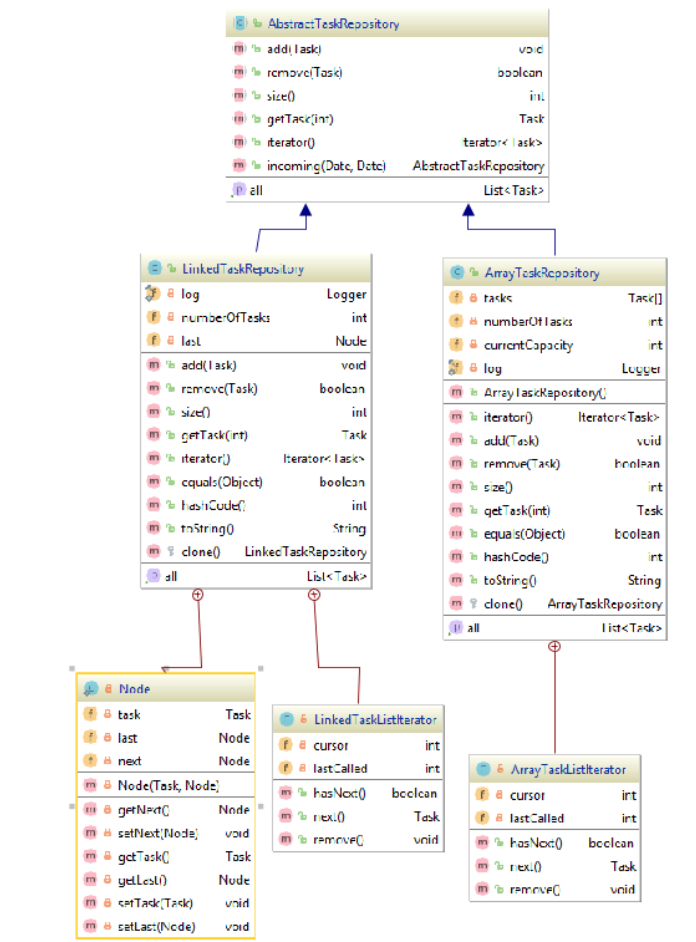


Task		
f	time	Date
f	interval	int
f	log	Logger
f	sdf	SimpleDateFormat
m	Task(String, Date)	
m	Task(String, Date, Date, int)	
m	getDateFormat()	SimpleDateFormat
m	setTime(Date, Date, int)	void
m	nextTimeAfter(Date)	Date
m	equals(Object)	boolean
m	hashCode()	int
m	toString()	String
m	clone()	Task
p	repeated	boolean
p	title	String
p	startTime	Date
p	time	Date
p	formattedDateStart	String
p	formattedDateEnd	String
p	repeat interval	int
p	active	boolean
p	endTime	Date
p	formattedRepeated	String

TaskIO		
f	simpleDateFormat	SimpleDateFormat
f	TIME_ENTITY	String[]
f	secondsInDay	int
f	secondsInHour	int
f	secondsInMin	int
f	log	Logger
m	write(AbstractTaskRepository, OutputStream)	void
m	read(AbstractTaskRepository, InputStream)	void
m	writeBinary(AbstractTaskRepository, File)	void
m	readBinary(AbstractTaskRepository, File)	void
m	write(AbstractTaskRepository, Writer)	void
m	read(AbstractTaskRepository, Reader)	void
m	writeText(AbstractTaskRepository, File)	void
m	readText(AbstractTaskRepository, File)	void
m	getTaskFromString(String)	Task
m	getIntervalFromText(String)	int
m	getDateFromText(String, boolean)	Date
m	getTitleFromText(String)	String
m	getFormattedTask(Task)	String
m	getFormattedInterval(int)	String
m	rewriteFile(ObservableList<Task>)	void

Notificator		
f	millisecondsInSec	int
f	secondsInMin	int
f	log	Logger
f	tasksList	ObservableList<Task>
m	Notificator(ObservableList<Task>)	
m	run()	void
m	showNotification(Task)	void
m	getTimeInMinutes(Date)	long



DateService	
SECONDS_IN_MINUTE	int
MINUTES_IN_HOUR	int
HOURS_IN_A_DAY	int
service	TasksService
DateService(TasksService)	
getLocalDateValueFromDate(Date)	LocalDate
getDateValueFromLocalDate(LocalDate)	Date
getDateMergedWithTime(String, Date)	Date
getTimeOfDayFromDate(Date)	String

TasksService	
savedTasksFile	File
TasksService(File)	
getIntervalInHours(Task)	String
formTimeUnit(int)	String
parseFromStringToSeconds(String)	int
filterTasks(Date, Date)	Iterable<Task>
observableList	ObservableList<Task>

TasksOperations	
tasks	ArrayList<Task>
TasksOperations(ObservableList<Task>)	
incoming(Date, Date)	Iterable<Task>
calendar(Date, Date)	SortedMap<Date, Set<Task>>

NewEditController		
clickedButton	Button	
log	Logger	
currentStage	Stage	
currentTask	Task	
service	TaskService	
dateService	DateService	
incorrectInputMade	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	
setClickedButton(Button)	void	
setCurrentStage(Stage)	void	
initialize()	void	
initNewWindow(String)	void	
initEditWindow(String)	void	
switchRepeatedCheckbox(ActionEvent)	void	
hideRepeatedTaskModule(boolean)	void	
saveChanges()	void	
closeDialogWindow()	void	
collectFieldsData()	Task	
makeTask()	Task	
service	TaskService	
currentTask	Task	
tasksList	ObservableList<Task>	

TaskController		
log	Logger	
tasksList	ObservableList<Task>	
service	TaskService	
dateService	DateService	
editNewStage	Stage	
infoStage	Stage	
mainTable	TableView	
tasks	TableView	
columnName	TableColumn<Task, String>	
columnTime	TableColumn<Task, String>	
columnRepeated	TableColumn<Task, String>	
labelCount	Label	
datePickerFrom	DatePicker	
fieldTimeFrom	TextField	
datePickerTo	DatePicker	
fieldTimeTo	TextField	
initialize()	void	
updateCountLabel(ObservableList<Task>)	void	
showTaskDialog(ActionEvent)	void	
deleteTask()	void	
showDetailedInfo()	void	
showFilteredTasks()	void	
getDateFromFilterField(LocalDate, String)	Date	
resetFilteredTasks()	void	
service	TaskService	

TaskInfoController		
log	Logger	
labelTitle	Label	
labelStart	Label	
labelEnd	Label	
labelInterval	Label	
labelsActive	Label	
initialize()	void	
closeWindow()	void	

Main		
primaryStage	Stage	
defaultWidth	int	
defaultHeight	int	
log	Logger	
classLoader	ClassLoader	
savedTasksFile	File	
service	TaskService	
start(Stage)	void	
main(String[])	void	