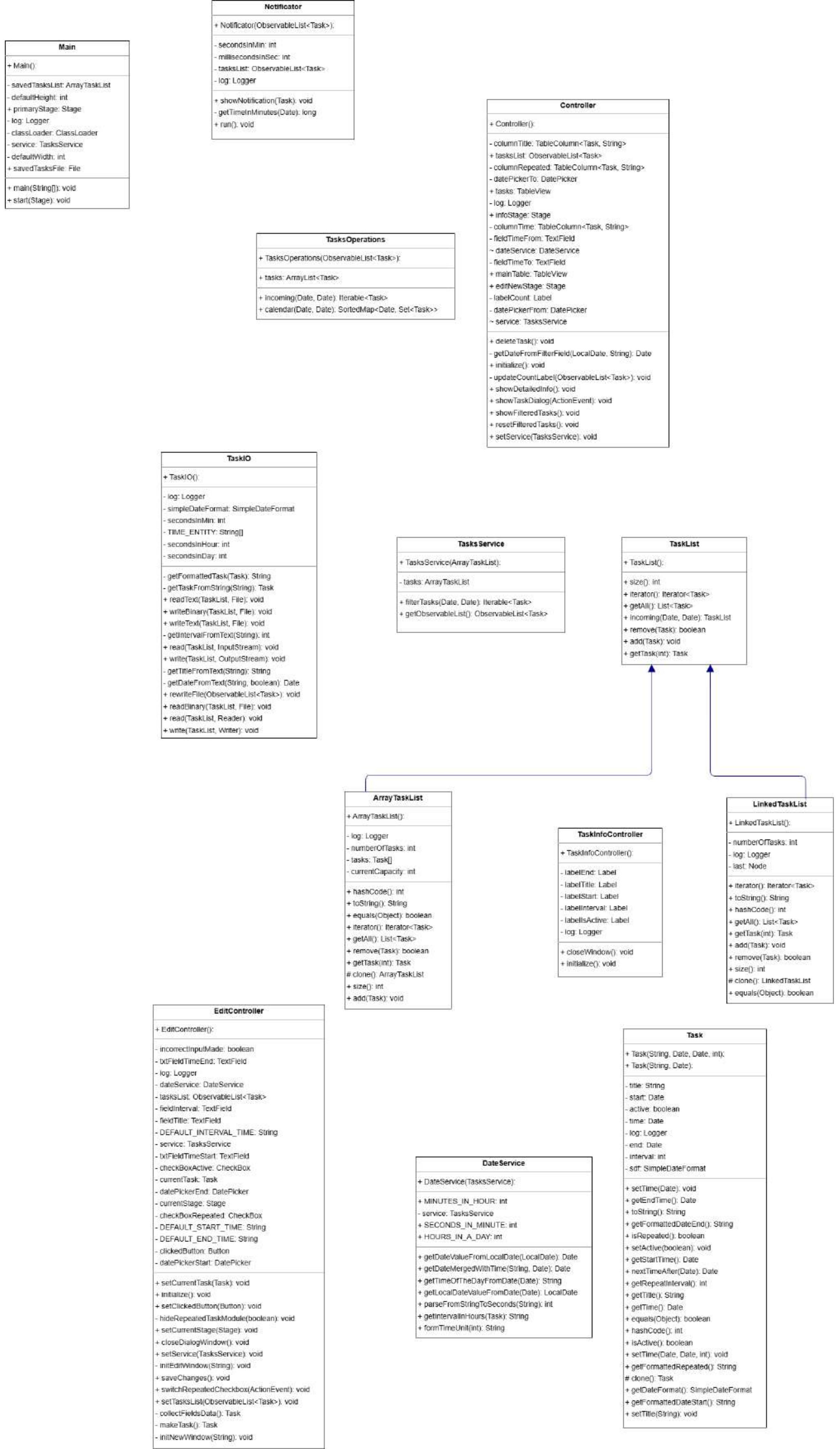
Main
+ Main():
- savedTasksList: ArrayTaskList
- defaultHeight: int
+ primaryStage: Stage
- log: Logger
- classLoader: ClassLoader
- service: TasksService
- defaultWidth: int
+ savedTasksFile: File
+ main(String[]): void
+ start(Stage): void
Notificator
+ Notificator(ObservableList<Task>):
- secondsInMin: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