

HomeActivity
+ onCreate(savedInstanceState : Bundle?): void + goFlashMind(view View): void + goAnalysis(view View): void + goPlanning(view View): void + goToDoList(view View): void + goSettings(view View): void + goFocus(view View): void

FlashMindActivity
- tags: ArrayList<Tag>
+ onCreate(savedInstanceState : Bundle?): void + swapCard (view View): void + swapExercise (view View): void

ToDoActivity
- toDoController: ToDoController
+ onCreate(savedInstanceState : Bundle?): void + clickAddToDoItem(): void + clickDoneToDoItem(): void + clickDeleteToDoItem(): void

ToDoController
- toDoItems: ArrayList<ToDoItem>
+ addToDoItem(item: ToDoItem): void + deleteToDoItem(item: ToDoItem): void - findItem(item: ToDoItem): ToDoItem + done(item: ToDoItem): void

ToDoItem
- infoStr: String
- isDone: boolean
+ ToDoItem():

PlaningActivity
- planningController : PlanningController
+ onCreate(savedInstanceState : Bundle?): void + clickDay(view: View): void + clickAddEvent(view: View): void + clickDeleteEvent(view: View): void

PlanningController
-eventItems: ArrayList<EventItem>
+ addEvent(item: EventItem): void + deleteEventItem (item: EventItem): void - findItem(item: EventItem): EventItem - getDayItems(date:DateTime): ArrayList<EventItem>

FocusActivity
- pomodoroItem: PomodoroItem
+ onCreate(savedInstanceState : Bundle?): void + clickStart(view View): void + clickPause (view View) :void + clickCancel (view View) :void - reduceTime (): void + clickAddTag (view View) : void

PomodoroItem
- startTime: DateTime
- endTime: DateTime
- pomodoroTime: int
- breakTime: int
- isPaused: bool
- tag: Tag
+ Pomodoro() + end(): void

TagActivity
- field: time

TagController
- tags: ArrayList<Tag>

Tag
- name: String

d

Card
- question: CardData - answer: CardData - tag: Tag - answers : ArrayList<AnswerItem> - isMarked: bool - title: String
+ Card(tag: Tag, data: CardData): + addAnswer(answer: AnswerItem): bool + checkMark(lastAnswerItem: AnswerItem, currentDate: date) : void + doneTrue(): void + doneFalse(): void

<<enumeration>> DataType
IMAGE TEXT SOUND

CardData
+ dataType: DataType
+ CardData(dataType: DataType)

AnswerItem
- date: dateTime - status: bool
+ method(type): type

ImageData
+ imagePath: str

TextData
+ text: str

EventItem
- date : DateTime - infoStr: String
+ EventItem():

CardCreateActivity
+ cardCreateController: CardCreateController
+ onCreate(savedInstanceState : Bundle?): void + clickImageOrTxt(view:View):void + clickAddTag (view View) : void + clickCreate(view View): void

CardCreateController
+ card: Card
+ addCreate():void

CardActivity
+ cards: ArrayList<Card>
+ onCreate(savedInstanceState : Bundle?): void + clickAddCard(view View): void + clickDetail(view View): void + clickDelete(view View): void

CardDetailActivity
+ card: Card
+ onCreate(savedInstanceState : Bundle?): void + clickFlip(view View): void

--

Color
selectName:Str: String

ExerciseActivity
+ exercisesController: ExerciseController

+ field: type
+ onCreate(savedInstanceState : Bundle?): void
+ clickAddTag (view View): void
+ clickDelete (view View): void

- tags: ArrayList<Tag>
+ addTag(item: Tag): void
+ deleteTag(item: Tag): void
- findItem(item: Tag): ToDoItem

- name: String
- color: Color
- nofMarkedCard: int
+ Tag():
+ getter/setter methods

AnalysisActivity
- analyzer : Analyzer
+ onCreate(savedInstanceState : Bundle?): void
+ clickAnalysisDropbox(view View) : void
+ selectAnalysis(view View) : void

AnalysisController
- totalTime: int
- totalNumberOfPomodoro: int
- AnalysisData: ArrayList <AnalysisData>
+ Analysis()
- findAnalysisData(tag:Tag) : analysisData
+ analyzePomodoro(pomodoro: Pomodoro): void
+ calculateLineGraph(): Dictionary

	- colorNameStr: Sting
	- rgbCode: String
	+ Color(rgbCode:String):

+ exercisesController: ExerciseController
+ onCreate(savedInstanceState : Bundle?): void
+ clickShowAnswer(view View): void
+ clickTrue(view View): void
+ clickFalse(view View): void
+ clickSwapOtherCard(view View): void

	AnalysisData
	- tag : Tag
	- pomodoroCount: int
	- minuteCount:int
	+ addNumberOfPomodoro: void
	+ addMinute(count:int): void
	+ isEqual(tag: Tag): boolean

ExerciseController
+ card: Card
+ getMarkedCard():void
+ doneTrue(): void
+ doneFalse(): void