Time.java

- hour : int - minute : int - second : int

- isMilitary : boolean - military Hour : int - amPm : boolean isMidnight: boolean

+ Time(): Constructor

+ Time(boolean isMilitary, int hour, int minute, int second): Constructor

+ getAmPm(): boolean + getIsMidnight(): boolean

+ setIsMidnight(boolean midnight):

+ setAmPm(boolean hourMode) : void

+ getIsMilitary(): boolean

+ setIsMIlitary(boolean militaryMode): void

+ getHour(): int

+ setHour(int hour) : void

+ getMinute(): int

+ setMilitaryHour(int milHour): void

+ getMilitaryHour(): int

+ setMinute(int minute): void

+ getSecond(): int

+ setSecond(int second): void

+ updateSeconds(): void

+ updateSecondsTimer(): void

+ updateMinutes(): void

+ updateMinutesTimer(): void

+ updateHours(): void

+ updateHoursTimer(): void

Clock.iava

-f:JFrame

- panelCont : JPanel

- p : JPanel

stopWatchPanel : JPanel

- timerPanel : JPanel

- clearScreenPanel : JPanel

- timeF : JTextField

- timerF : JTextField

- stopWatchF : JTextField

- dateF : JTextField

- changeTime : JButton

- stopWatch : JButton

- timer : JButton

- clearScreen : JButton

changeDate: JButton

- zoomln : JButton

- zoomOut : JButton

- switchToClock1 : JButton

switchToClock2 : JButton

- redisplayScreen : JButton

- switchHourMode : JButton resetStopWatch: JButton

- pauseStopWathc: JButton

- pauseTimer : JButton

- resetTimer : JButton

- m zoomCounter:int

- m month:int

- m day:int

- m_dayOfWeek : String

- isEnable : boolean

- timerSet : boolean

- isStopWatchPaused : boolean

- stopWatchRunning : boolean

- goodTimeInput: boolean

- goodTimerInput: boolean

- goodCalendarInput: boolean

- isTimerPaused : boolean

+ timerClock : Time

+ stopwatch : Time

+ timerClock : Time

+ week : DayOfWeek

+ Clock(): Constructor

+ getZoomCounter(): int

+ setZoomCounter(int counter): void

+ getClockDayOfWeek(): String

+ qui(): void

+ main(String[] args): void

DavOfWeek.iava

- jan : int

- feb : int

- mar : int

- apr : int

- may: int

- jun : int

- jul : int

- aug : int sep : int

- oct : int

- nov : int

- dec : int

- m month:int

- m day:int

- m dayOfWeek : String

+ DayOfWeek(): Constructor

+ DayOfWeek(int month, int day, String

dayOfWeek): Constructor

+ calculateDayOfWeek(): void

+ calculatePosition(int positionInWeek):

+ getDayOfWeek(): String

+ getMonth(): int

+ setMonth(int month): void

+ setDay(int day): void

+ getDay(): int

+ incrementDayOfWeek(): String