Clock_Observer_Application::cClockObserverApplicationBuilder

- - mClock: cClock
- - mptrAnalogClock: cAnalogClock*
- - mptrDigitalClock: cDigitalClock*
- - mptrAnalogDigitalClock: cAnalogDigitalClock*

- + cClockObserverApplicationBuilder()
- + ~cClockObserverApplicationBuilder()
- + execute(): void
- - buildRelations(): void

{leaf}

Clock::cClock

- - mHour: cLimitCounter
- - mMinute: cOverflowCounter
- - mSecond: cOverflowCounter

- + cClock(Hour: uint32_t, Minute: uint32_t, Second: uint32_t)
- + ~cClock()
- + setHour(Hour: uint32_t): void
- + setMinute(Minute: uint32_t): void
- + setSecond(Second: uint32_t): void
- + reset(): void
- + getHour(): uint32_t {query}
- + getMinute(): uint32_t {query}
- + getSecond(): uint32_t {query}
- + *update(): void*
- + *show(): void {query}*

Counter::cLimitCounter

- # mCountValue: uint32_t
- # mCountLimit: uint32_t {readOnly}

- + cLimitCounter(CountValue: uint32_t, CountLimit: uint32_t)
- + *~cLimitCounter()*
- + setCountValue(CountValue: uint32_t): void
- + getCountValue(): uint32_t {query}
- + *resetCountValue(): void*
- + *count(): void*

-mHour

-mrefNextCounter

-mMinute

Counter::cOverflowCounter   {leaf}

- - mrefNextCounter: cLimitCounter&

- + cOverflowCounter(CountValue: uint32_t, CountLimit: uint32_t, refNextCounter: cLimitCounter&)
- + ~cOverflowCounter()
- + *resetCountValue(): void*
- + *count(): void*

-mSecond

-mClock

-mptrDigitalClock

Clock::cDigitalClock

- + cDigitalClock(Hour: uint32_t, Minute: uint32_t, Second: uint32_t)
- + ~cDigitalClock()
- + *show(): void {query}*

-mptrAnalogClock

Clock::cAnalogClock

- + cAnalogClock(Hour: uint32_t, Minute: uint32_t, Second: uint32_t)
- + ~cAnalogClock()
- + *show(): void {query}*

-mptrAnalogDigitalClock

Clock::cAnalogDigitalClock   {leaf}

- + cAnalogDigitalClock(HourAnalog: uint32_t, MinuteAnalog: uint32_t, SecondAnalog: uint32_t, HourDigital: uint32_t, MinuteDigital: uint32_t, SecondDigital: uint32_t)
- + ~cAnalogDigitalClock()
- + *update(): void*
- + *show(): void {query}*

std::exception

User_Exception::excRangeException

- + excRangeException()
- + *~excRangeException()*
- + *what(): char* {query}*