CalculatorController

+ doDisplayFirstNumber()
+ doSaveOpertion()
+ doDisplaySecondNumber()
+ doDisplayResult()
+ doDisplayError()
+ doTurnOff()

- controller

Calculator

+ create(controller : int)
+ pressNumberButton()
+ pressDotButton()
+ pressOperationButton()
+ pressOnButton()
+ pressEqualsButton()
+ timeout()
~ setState()
~ displayFirstNumber()
~ saveOperation()
~ displaySeconNumber()
~ displayResult()
~ displayError()
~ turnOff()

- state

CalculatorState

+ *numberEntered(calculator : Calculator)*
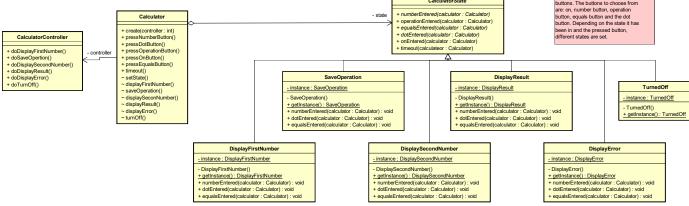+ operationEntered(calculator : Calculator)
+ *equalsEntered(calculator : Calculator)*
+ *dotEntered(calculator : Calculator)*
+ onEntered(calculator : Calculator)
+ timeout(calculateor : Calculator)

The diagram showes the system of a calculator. To use it the user presses buttons. The buttons to choose from are: on, number button, operation button, equals button and the dot button. Depending on the state it has been in and the pressed button, different states are set.

SaveOperation

- instance : SaveOperation

- SaveOperation()
+ getInstance() : SaveOperation
+ numberEntered(calculator : Calculator) : void
+ dotEntered(calculator : Calculator) : void
+ equalsEntered(calculator : Calculator) : void

DisplayResult

- instance : DisplayResult

- DisplayResult()
+ getInstance() : DisplayResult
+ numberEntered(calculator : Calculator) : void
+ dotEntered(calculator : Calculator) : void
+ equalsEntered(calculator : Calculator) : void

TurnedOff

- instance : TurnedOff

- TurnedOff()
+ getInstance() : TurnedOff

DisplayFirstNumber

- instance : DisplayFirstNumber

- DisplayFirstNumber()
+ getInstance() : DisplayFirstNumber
+ numberEntered(calculator : Calculator) : void
+ dotEntered(calculator : Calculator) : void
+ equalsEntered(calculator : Calculator) : void

DisplaySecondNumber

- instance : DisplaySecondNumber

- DisplaySecondNumber()
+ getInstance() : DisplaySecondNumber
+ numberEntered(calculator : Calculator) : void
+ dotEntered(calculator : Calculator) : void
+ equalsEntered(calculator : Calculator) : void

DisplayError

- instance : DisplayError

- DisplayError()
+ getInstance() : DisplayError
+ numberEntered(calculator : Calculator) : void
+ dotEntered(calculator : Calculator) : void
+ equalsEntered(calculator : Calculator) : void

The sequence diagram presents the sunny scenario of how a calculator works. It is the simpliest case when the user after turning the calculator on, presses one number, next an operation, another number and the equal button. Having done that, the user waits five minutes until the calculator turns off again.

| : CalculatorController | : Calculator | : DisplayFirstNumber | : SaveOperation | : DisplaySecondNumber | : DisplayResult | : TurnedOff |

1: onEntered(calculator:Calculator)

1.1: displayFirstNumber()

1.1.1: doDisplayFirstNumber()

1.2: setState(DisplayFirstNumber.getInstance())

2: numberEntered(calculator:Calculator)

2.1: displayFirstNumber()

2.1.1: doDisplayFirstNumber()

3: operationEntered(calculator:Calculator)

3.1: saveOperation()

3.1.1: doSaveOpertion()

3.2: setState(SaveOperation.getInstance())

4: numberEntered(calculator:Calculator)

4.1: displaySeconNumber()

4.1.1: doDisplaySecondNumber()

4.2: setState(DisplaySecondNumber.getInstance())

5: equalsEntered(calculator:Calculator)

5.1: displayResult()

5.1.1: doDisplayResult()

5.2: setState(DisplayResult.getInstance())

6: timeout(calculateor:Calculator)

6.1: turnOff()

6.1.1: doTurnOff()

6.2: setState(TurnedOff.getInstance())

State5

The diagram showes
the system of a
calculator. To use it the
user presses buttons.
The buttons to choose
from are: on, number
button, operation button,
equals button and the
dot button.

Press number or equal
[There are less than 9
numbers]

Press operation number /
Overwrite the operation

Press dot button [no
dot in the number] /
add the dot

Display first number

Press operation number

Display first number &
have the operation saved

Timeout (5 minutes)

/ set first number as 0

Press the dot / Set the
first number as "0."

Press equals button

Press number button / set first
number as the new number

Press number button

Press the dot / Display "0."

Press operation button

Display second number

Press equals button [The operation is
not division and the second number is
not 0]

Press dot button [no dot in
the number] / add the dot

Press on button / set first
number as 0

Display max 9 numbers of
the result

Press number button [There are
less than 9 numbers]

Press equals button [The operation is
division and the second number is 0]

Press equals button

Display Error

Press ON button