

```

# File komega-v01e.py

# Author: G.Doeben-Henisch
# First date: September 4, 2020
# Last change: 11.September 2020

#####
# Execution Environment of my local machine:
# (venv) gerd@gerd-ub2:~/env/komega/tst$ python3 komega-v01d.py
#

#####
# GITHUB
#
# We use a github repository at:
# https://github.com/szmt/komega.git
#
# Im working from a unix-shell using the following github-commands:
# https://git-scm.com/docs/git

#####
# BACKGROUND THEORY
#
# This code is a translation of a theory described in the blog
# https://www.uffmm.org
#
# Last document for the specification of this code:
#
'''
https://www.uffmm.org/2020/09/10/komega-requirements-no-4-version-4-basic-application-
scenario/
'''

#####
# ACTOR STORY
#
# In the specifications an actor story [AS] has been specified. This AS requires # some basic states
which are dedicated for certain tasks to do:

'''
ACTOR STORY

S1: START
S2: EDIT P(roblem description)
S3: EDIT S (actual state) AND X (change rules)
S4: SIMULATION (Applying X to S)
S5: EVALUATION (After the simulation)
S6: STOP
'''

# MAIN IDEA
'''

```

According to the above mentioned actor story the user will be sitting in front of a system interface [SI] which works first only as a console.

In the beginning the user is placed in a start state S1 showing all options available.

The user can select one of these options and can from start state S1 reach all other states S2-S6.

'''

```
#####
```

```
# IMPORTS
```

```
#####
```

```
# SUPPORTING FUNCTION
```

```
#
```

```
# No funtions yet
```

```
# CLASSES
```

```
#
```

```
'''
```

For every state there exists one working class to do the job.

The special class 'Publish' in this code exists only because the interaction of the user with the system will happen with an interactive website which uses HTML and javascript. Here in this experimental environment a simple unix-console is used.

'''

```
import kcv1e as kc
```

```
#####
```

```
# Main Programm
```

```
#
```

```
#####
```

```
# Start main loop
```

```
#
```

```
# The loop will work as long as the value of the variable 'loop' is different to 'N'
```

```
loop='Y'
```

```
while loop!='N':
```

```
#####
```

```
# STATE 1 : START
```

```
# Show available options
```

```
# Get feedback for selection
```

```
# Confirm the selection
```

```
# Distribute to different states
```

```
    kc.ast.menushow()
```

```
# Ask back for selection number
```

```
    opt=input('Enter a Number [1-6] for Menu Option \n')
```

```
# Evaluate the selection
```

```
kc.ast.badoption(opt)
```

```
# Call to a class instance
```

```
if opt=='2':  
    kc.pub.show(kc.ap)
```

```
elif opt=='3':  
    kc.pub.show(kc.asx)
```

```
elif opt=='4':  
    kc.pub.show(kc.asim)
```

```
elif opt=='5':  
    kc.pub.show(kc.aev)
```

```
elif opt=='6':  
    kc.pub.show(kc.astp)
```

```
# Clarify how to continue
```

```
loop=input("STOP = 'N', CONTINUE != 'N' \n")
```

```
'''
```

```
TEST
```

```
(venv) gerd@gerd-ub2:~/env/komega/tst$ python3 komega-v01e.py
```

```
1 is START
```

```
2 is EDIT P
```

```
3 is EDIT S and X
```

```
4 is SIMULATION
```

```
5 is EVALUATION
```

```
6 is STOP
```

```
Enter a Number [1-6] for Menu Option
```

```
0
```

```
!!You have selected a bad option
```

```
STOP = 'N', CONTINUE != 'N'
```

```
a
```

```
1 is START
```

```
2 is EDIT P
```

```
3 is EDIT S and X
```

```
4 is SIMULATION
```

```
5 is EVALUATION
```

```
6 is STOP
```

```
Enter a Number [1-6] for Menu Option
```

```
11
```

```
!!You have selected a bad option
```

```
STOP = 'N', CONTINUE != 'N'
```

```
a
```

```
1 is START
```

```
2 is EDIT P
```

3 is EDIT S and X
4 is SIMULATION
5 is EVALUATION
6 is STOP
Enter a Number [1-6] for Menu Option
1
!!You have selected the state START
STOP = 'N', CONTINUE != 'N'
a
1 is START
2 is EDIT P
3 is EDIT S and X
4 is SIMULATION
5 is EVALUATION
6 is STOP
Enter a Number [1-6] for Menu Option
2
!!You have selected the state EDIT P
Role : "Pedit"
Name : "ap"
STOP = 'N', CONTINUE != 'N'
a
1 is START
2 is EDIT P
3 is EDIT S and X
4 is SIMULATION
5 is EVALUATION
6 is STOP
Enter a Number [1-6] for Menu Option
3
!!You have selected the state EDIT S and X
Role : "SXedit"
Name : "asx"
STOP = 'N', CONTINUE != 'N'
a
1 is START
2 is EDIT P
3 is EDIT S and X
4 is SIMULATION
5 is EVALUATION
6 is STOP
Enter a Number [1-6] for Menu Option
4
!!You have selected the state SIMULATION
Role : "SIM"
Name : "asim"
STOP = 'N', CONTINUE != 'N'
a
1 is START
2 is EDIT P
3 is EDIT S and X
4 is SIMULATION

```
5 is EVALUATION
6 is STOP
Enter a Number [1-6] for Menu Option
5
!!You have selected the state EVALUATION
Role : "EVAL"
Name : "aev"
STOP = 'N', CONTINUE != 'N'
a
1 is START
2 is EDIT P
3 is EDIT S and X
4 is SIMULATION
5 is EVALUATION
6 is STOP
Enter a Number [1-6] for Menu Option
6
!!You have selected the state STOP
Role : "STOP"
Name : "astp"
STOP = 'N', CONTINUE != 'N'
a
1 is START
2 is EDIT P
3 is EDIT S and X
4 is SIMULATION
5 is EVALUATION
6 is STOP
Enter a Number [1-6] for Menu Option
7
!!You have selected a bad option
STOP = 'N', CONTINUE != 'N'
N
(venv) gerd@gerd-ub2:~/env/komega/tst
```

'''

```
#####
```

```
# File kcv1e.py
```

```
# Author: G.Doeben-Henisch
# First date: September 6, 2020
# Last date: September 11, 2020
```

```
#####
```

```
# CLASS DEFINITIONS
```

```
class Start:
```

```

def __init__(self):
    self.menulist = ['START','EDIT P','EDIT S and
X','SIMULATION','EVALUATION','STOP']

def menushow(self):
    i=0 # Counter for menu-loop
    for state in self.menulist:
        i=i+1
        print(i,' is ',state)

def badoption(self,opt):
    if int(opt)<1 or int(opt)>6:
        print('!!You have selected a bad option')

    if int(opt)>0 and int(opt)<7:
        print('!!You have selected the state',self.menulist[int(opt)-1])

class Actor:
    def __init__(self,role,name):
        self.role = role
        self.name = name

class Publish():

    def show(self,other):
        print('Role : "%s"'%other.role)
        print('Name : "%s"'%other.name)

#####
# CLASS INSTANCES

ast=Start()

ap=Actor("Pedit","ap")
asx=Actor('SXedit','asx')
asim=Actor('SIM','asim')
aev=Actor('EVAL','aev')
astp=Actor('STOP','astp')

pub=Publish()

```