Industrial Software Development (ISDe) – January 18, 2021

Name: ..... Surname: ..... Student ID: .....

Exercise 4 (10 points) - 30 mins

Write a machine **m** that prints **HAM - SPAM - JAM** if it encounters the sequence 'HAM', 'SPAM', 'JAM', as shown in the example below. If the input sequence does not match the correct one, the machine has to return to its ZERO state and print **'wrong sequence!'**

```
#  MAIN

sequence1 = ['HAM', 'SPAM', 'JAM']

sequence2 = ['X',
             'HAM', 'SPAM', 'JAM',
             'X']

sequence3 = ['HAM', 'SPAM', 'JAM',
             'HAM', 'SPAM', 'X', 'JAM',
             'HAM', 'SPAM', 'JAM']

print('\nSEQUENCE 1')
m = Machine()
for s in sequence1:
    m.process_input(s)

print('\nSEQUENCE 2')
m = Machine()
for s in sequence2:
    m.process_input(s)

print('\nSEQUENCE 3')
m = Machine()
for s in sequence3:
    m.process_input(s)
```

The output is:

```
SEQUENCE 1
HAM - SPAM - JAM

SEQUENCE 2
  wrong sequence!
HAM - SPAM - JAM
  wrong sequence!

SEQUENCE 3
HAM - SPAM - JAM
  wrong sequence!
  wrong sequence!
HAM - SPAM - JAM
```

Use the State design pattern to implement the classes **Machine** and **State.** The above 'main' code must run correctly, producing the showed output.