

```
from time import sleep
class espaceDeTuples():
    def OUT(self, element):
        self.listeTuples.append(element)
    def IN(self, element, tab):
        resTemp = self.existe(element, tab)
        res = list()
        for index in tab:
            res.append(resTemp[index])
        self.listeTuples.remove(resTemp)
        return res

    def RD(self, element, tab):
        resTemp = self.existe(element, tab)
        res = list()
        for index in tab:
            res.append(resTemp[index])
        return res

    def ADD(self, element, tab):
        resTemp = self.existe(element, tab)
        index = self.listeTuples.index(resTemp)
        listTuple = list(self.listeTuples[index])
        for i in tab:
            listTuple[i] = element[i]
        self.listeTuples[index] = tuple(listTuple)

    def INUNBLOCKED(self, element, tab):
        resTemp = self.existeNonBloquant(element, tab)
        if resTemp is None:
            return None
        res = list()
        for index in tab:
            res.append(resTemp[index])
        self.listeTuples.remove(resTemp)
        return res

    def existe(self, template, tab):
        tuplePossible = []
        while (True):
            sleep(0.6)
            for tupleI in self.listeTuples:
                flag = True
                if len(tupleI) == len(template):
                    for i in range(len(template)):
                        if type(template[i]) != type(tupleI[i]):
                            flag = False
                    if flag:
                        tuplePossible.append(tupleI)
            for tupleI in tuplePossible:
                allIndex = list(range(len(tupleI)))
                for i in tab:
                    for j in allIndex:
                        if i == j:
                            allIndex.remove(j)
                flag = True
                for i in allIndex:
                    if template[i] != tupleI[i]:
                        flag = False
                if flag:
```

```
61         return tupleI
62
63     def existeNonBloquant(self, template, tab):
64         tuplePossible = []
65         for tupleI in self.listeTuples:
66             flag = True
67             if len(tupleI) == len(template):
68                 for i in range(len(template)):
69                     if type(template[i]) != type(tupleI[i]):
70                         flag = False
71             if flag:
72                 tuplePossible.append(tupleI)
73         for tupleI in tuplePossible:
74             allIndex = list(range(len(tupleI)))
75             for i in tab:
76                 for j in allIndex:
77                     if i == j:
78                         allIndex.remove(j)
79             flag = True
80             for i in allIndex:
81                 if template[i] != tupleI[i]:
82                     flag = False
83             if flag:
84                 return tupleI
85
86     def __init__(self):
87         self.listeTuples = list()
88
```