```
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:
```

```
                    return tupleI

    def existeNonBloquant(self, template, tab):
        tuplePossible = []
        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:
                    return tupleI

    def __init__(self):
        self.listeTuples = list()

```