

# TP 5

L'objectif de ce projet est d'ajouter des exceptions aux classes `FileDePriorite` et `CodeBinaire` des précédents TP.

Au préalable vous créez les propriétés `est_vide` et `element` pour la classe `FileDePriorite`.

## 1 Classe FileDePriorité

- Cette classe est susceptible de lever deux types d'exception (définies dans le module `file_de_priorite`):
- `FileDePrioriteVideErreur` qui est levée lorsqu'on essaye d'obtenir ou défiler l'élément en tête d'une file vide;
  - `ElementNonComparableErreur` qui est levée lorsqu'on essaye d'ajouter un élément :
    - dont la classe de possède pas les opérations de comparaison;
    - qui ne peut pas être comparé avec les éléments déjà présents dans la file.

Voici un exemple d'utilisation de la classe `FileDePriorite`:

```
In [1]: from file_de_priorite import FileDePriorite
```

```
In [2]: f=FileDePriorite()
```

```
In [3]: f.enfiler(complex(1,2))
```

```
-----  
ElementNonComparableErreur          Traceback (most recent call last)  
<ipython-input-3-abb9c06e0d34> in <module>()  
----> 1 f.enfiler(complex(1,2))
```

...

```
ElementNonComparableErreur: La classe de (1+2j) ne possède pas les  
méthodes de comparaison
```

```
In [4]: f.enfiler(1)
```

```
In [5]: f.enfiler(1.0)
```

```
In [6]: f.enfiler("aa")
```

```
-----  
ElementNonComparableErreur          Traceback (most recent call last)  
<ipython-input-6-abb9c06e0d34> in <module>()  
----> 1 f.enfiler("aa")
```

...

ElementNonComparableErreur: aa ne peut être comparé à 1.0

## 2 Classe CodeBinaire

Cette classe est susceptible de lever deux types d'exception (définis dans le module `code_binaire`):

- `TypeError` qui est levée lorsque l'on essaye :
    - de créer un code binaire
    - d'ajouter à un code binaire, autre chose qu'un bit;
    - de concaténer en élément qui n'est pas un code binaire;
  - `AuMoinsUnBitErreur` qui est levée lorsque l'on supprime un nombre de bit qui aboutirait à avoir un code binaire sans bit : un code binaire doit toujours être composé d'au moins un bit.
- Voici un exemple d'utilisation de la classe `CodeBinaire` :

```
In [1]: from code_binaire import CodeBinaire
```

```
In [2]: c=CodeBinaire(1)
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-2-7384a8cd4683> in <module>()  
----> 1 c=CodeBinaire(1)
```

...

TypeError: bit doit construit à partir de Bit

```
In [3]: from code_binaire import Bit
```

```
In [4]: c=CodeBinaire(Bit.BIT_0)
```

```
In [5]: del(c[0])
```

```
-----  
AuMoinsUnBitErreur                       Traceback (most recent call last)  
<ipython-input-5-d99311800d6a> in <module>()  
----> 1 del(c[0])
```

...

AuMoinsUnBitErreur: Un code binaire doit posséder au moins un bit