Guarder organizes the heap into **128** *Threads*

Each thread is divided in 16 *bags* (or classes) of sizes 2^4 to 2^{19}, that are randomly placed within the thread (i.e., the 1st bag is not necessarily that of size 2^4)

A l'initialisation, tout le heap est initialisé: 128 * (16 * _bibopBagSize)

Class BibopHeap:

```
(classes)
-> BibopObjCache
-> PerThreadMap
-> PerThreadBag
...
(and arrays)
...
-> _threadMap
-> _threadBag
-> _bagCache
-> _freeCache
```

Guarder allocates a proportion of guard pages inside each bag, defined in `xdefines.hh`: `default_rand_guard_prop` The protection frequency is then `1/default_rand_guard_prop`

Function call stack from malloc (methods of the BibopObjCache class):

```
xxmalloc
    |
allocateSmallObject
    |
malloc
    |
getRandomObject
    |
getNext
    |
TryRandomGuardPage
```

In the class BibopHeap::PerThreadBag, we can then compute the pattern as we do in Slimguard by adding a `pattern` field in the PerThreadBag class