GARBAGE COLLECTION!

= MARK AND SWEEP =

@theJennLee

Even with garbage collection, programs can still "leak", and eventually run out of memory!

The program would allocate new objects, but doesn't stop referencing old objects, so the collector can't recycle memory.

THE END.

Whenever a computer program does something, it uses memory! Memory holds the data your computer is working with. Because we live in a world where infinite memory is not real, the computer has to clean the memory to free up space. This is called "garbage collection".

One type of garbage collection is called "Mark and Sweep", which is a kind of "tracing garbage collection". This type cleans memory in two phases:

* Mark: The collector determines which objects are still in use.

* Sweep: The collector deallocates the unused objects, and frees the space they used to occupy.

The collector unmarks the objects so it can repeat mark and sweep later.

The freed space can now be used to store other objects!

... and frees the space the unused objects used to occupy!

SWEEP PHASE

During the sweep phase, the collector cleans up the unmarked objects...

MARK PHASE

This is a blank memory.

As a program runs, it creates objects...

... and the objects will take up space in the memory.

An object may reference other objects!

An example:

`my_object.buddy = new Buddy()`

Once an object is no longer referenced, it is no longer used.

During the mark phase, the collector starts from the root set, and marks the objects it can reach through references!

At the end of the mark phase, all reachable objects will be marked.

reachable and marked

not reachable and not marked

Not marked objects will be not reachable, which also means they are not used.