UNIVERSITY OF TWENTE.

The Object Lifecycle

Topic of Software Systems (TCS module 2)

Lecturer: Tom van Dijk

OBJECT LIFECYCLE

1. Loading the class
2. Constructing the object
3. …
4. Garbage collection

CLASS LOADING

- Load the compiled class: a `.class` file containing bytecode
- Do some checks
- Run static initializers

CONSTRUCTING THE OBJECT

- Allocate memory for the object
- Run the instance initializers
- Run the specified constructor (with the `new` keyword)

GARBAGE COLLECTION

- Computers have limited memory
- Many errors come from managing resources: memory leaks
- Java has garbage collection
- Garbage collection is automatic, you don’t think about it

GARBAGE COLLECTION

- If no reference exists to a object, it may be deleted
- Java deletes objects that are no longer reachable

WHEN ARE OBJECTS REACHABLE?

Mark objects are reachable:

1. Mark all objects that are referenced by static variables
2. Mark all objects that are referenced in the program stack
    - A list keeping track of method calls and local variables
    - Every method call adds a new block of information on top of the stack
    - Every finished execution, removes the last block from the stack
3. Mark all objects that are referenced by reachable objects (recursive)

MEMORY LEAKS IN JAVA

- Even Java can leak memory!
  - Caching (storing computation results to reuse later)
  - If a server maintains a list of connection handlers

- Explicitly dereference by setting a variable to `null`
- Wrap references in a `WeakReference` object
- No guarantee that/when objects will be collected