Australian
National
University

02

Classes and Objects 2

Locals, globals, heap

Garbage collection

Initializers, access control

enum types

Locals (stack), Globals (statics), and Heap (objects)

Local variables are declared within the scope of a method and hold temporary state. They disappear once the method returns.

Global variables (aka '*class variables*') are declared within the scope of a class (with a `static` qualifier), and exist as long as the class is loaded (which is usually for the duration of the program).

Heap variables (aka *'instance variables*') are declared within the scope of a class (without a `static` qualifier), and exist as long as the containing instance is reachable.

Garbage Collection

In some object oriented languages, the programmer must keep track of objects and delete them when they are no longer used.

This is error-prone.

Java uses a garbage collector to automatically collect objects that can no longer be used. Garbage collection approximates liveness by reachability (the collector conservatively assumes that any reachable object is live).

The `this` keyword

Within instance methods and constructors, the `this` keyword refers to the object whose method or constructor is being called.

- Disambiguating field names from parameters
  - Parameters and instance field names may clash. The `this` keyword explicitly refers to the instance.
- Calling other constructors
  - When there are multiple constructors, they may call each other using `this` as if it were the method name.

Access Control

Access modifiers determine whether fields and methods may be accessed by other classes

- Top level: `public` or *package-private*
- Member level: `public`, `protected`, *package-private*, or `private`

Modifier	Class	Package	Subclass	World
`public`	✓	✓	✓	✓
`protected`	✓	✓	✓	✗
no modifier	✓	✓	✗	✗
`private`	✓	✗	✗	✗

Class and Instance Members

The `static` keyword identifies class variables, class methods and constants.

- A **class variable** is common to all objects (there is only one version)
- A **class method** is invoked using a class name (not an object reference) and executes independently of any particular object.
- A **constant** can be declared by combining the `final` modifier with the `static` keyword.

Initializers

Fields may be initialized when they are declared. They can also be initialized by **initializer blocks**, which can initialize fields using arbitrarily complex code (error handling, loops, etc.).

- A static initializer block is consists of code enclosed by braces ‘{}’and preceded by the `static` keyword. It runs when the class is first accessed.
- A instance initializer block does not have the `static` keyword, and runs before the constructor body of the class.

Enum Types

An **enumerated type** is defined with the `enum` keyword.
A variable of enum type must be one of a set of predefined values.
This is useful for defining non-numerical sets such as `NORTH`, `SOUTH`, `EAST`, `WEST`, or `HD`, `D`, `CR`, `P`, `N`, etc.

- May have other **fields**
- May have **methods**
- May use **constructors**
- Can be used as argument to **iterators**