

**Actor** → Grid<Actor> gr = getGrid();  
 Location loc = getLocation(); // should be an Actor.getLocation() if outside Actor class  
 int dir = getDirection();  
 removeSelfFromGrid();  
 putSelfInGrid(gr, this); // this can be replaced w/ an actor  
 moveTo(newLoc); // newLoc is a Location

**Location** → Location loc2 = loc.getAdjacentLocation(direction); // direction is an int  
 int dirTowards = loc.getDirectionTowards(loc2);

**Grid<Actor>** → boolean valid = gr.isValid(loc); // loc is a Location  
 Actor actor = gr.get(loc);  
 [don't use put or add!]  
 ArrayList<Locations> locs = gr.getLocations(loc);

Valid Adjacent  
 Empty Adjacent  
 Occupied Adjacent

**Critter**

- \* getActors() → chooses which actors will be processed
- processActors(list) → does something with the chosen actors
- \* getMoveLocations() → chooses which locations we could move to
- \* selectMoveLocation(list) → picks one loc from the list (or null)
- makeMove(loc) → performs the move

\* - should not modify the Grid or any actors!

Get Actor at Location

```
Grid<Actor> gr = getGrid();
Location loc = ...;
if (gr.isValid(loc))
{
  Actor a = gr.get(loc);
  if (a != null)
  {
    // do something
  }
}
```

Finishing getActors()

```
public ArrayList<Actor> getActors()
{
  ArrayList<Actor> list = new ArrayList<Actor>();
  // get a bunch of actors and add them to list
  return list;
}
```

Finishing getMoveLocations()

```
public ArrayList<Locations> getMoveLocations()
{
  ArrayList<Location> list = new ArrayList<Location>();
  // add a bunch of Locations
  return list;
}
```