Searching Earthquake Data

Relationships Between Classes

The QuakeEntry Class

- QuakeEntry class is essentially a POJO
    - Plain Old Java Object — but here cannot create object without all characteristics

```
public class QuakeEntry {

    private Location myLocation;
    private String title;
    private double depth;
    private double magnitude;

    public QuakeEntry(…){

    }
…
```

The QuakeEntry Class

- QuakeEntry class is essentially a POJO
    - Plain Old Java Object — but here cannot create object without all characteristics
    - No default/parameterless constructor

```
public QuakeEntry(double lat, double lon, double mag,
                  String t, double d) {
   myLocation = new Location(lat,lon);
   magnitude = mag;
   title = t;
   depth = d;
 }
```

The QuakeEntry Class

- QuakeEntry class is essentially a POJO
    - Plain Old Java Object — but here cannot create object without all characteristics
    - No default/parameterless constructor
- Immutable with getters()

```
public class QuakeEntry {
    public Location getLocation(){..
    public double getDepth() {…
    public String getInfo() {…
    …
```

The QuakeEntry Class

- QuakeEntry class is essentially a POJO
  - Plain Old Java Object — but here cannot create object without all characteristics
  - No default/parameterless constructor
- Immutable with getters()

```
public class QuakeEntry {
     public Location getLocation(){..
     public double getDepth() {…
     public String getInfo() {…
     …
```

The QuakeEntry Class

- QuakeEntry class is essentially a POJO
  - Plain Old Java Object — but here cannot create object without all characteristics
  - No default/parameterless constructor
- Immutable with getters()

```
public class QuakeEntry {
    public Location getLocation(){..
    public double getDepth() {…
    public String getInfo() {…
    …
```

The QuakeEntry Class

- QuakeEntry class is essentially a POJO
  - Plain Old Java Object — but here cannot create object without all characteristics
  - No default/parameterless constructor
- Immutable with getters()
- Reasonable `.toString()` method

```
public class QuakeEntry {
    public Location getLocation(){..
    public double getDepth() {…
    public String getInfo() {…
    …
```

The Location Class

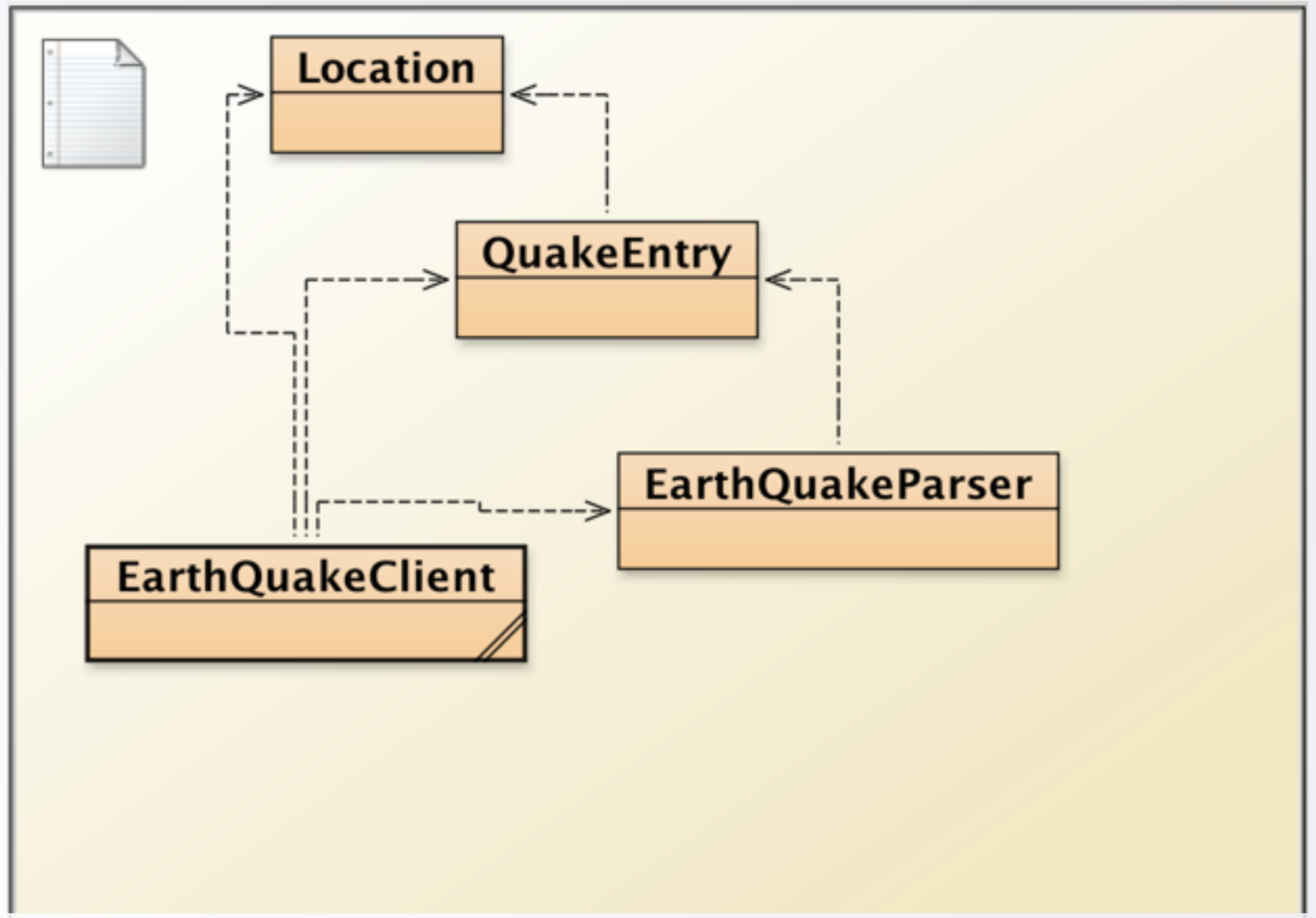
- Many, many contexts: beyond QuakeEntry
    - Use simple, functional design for course?
    - Use industrial strength design for course?
        - We'll adopt Android class

The Location Class

- Many, many contexts: beyond QuakeEntry
  - Use simple, functional design for course?
  - Use industrial strength design for course?
    - We'll adopt Android class
- Latitude and Longitude
  - Initialize from source
  - Distance from A to B?
  - More than state/POJO
    - Behavior!

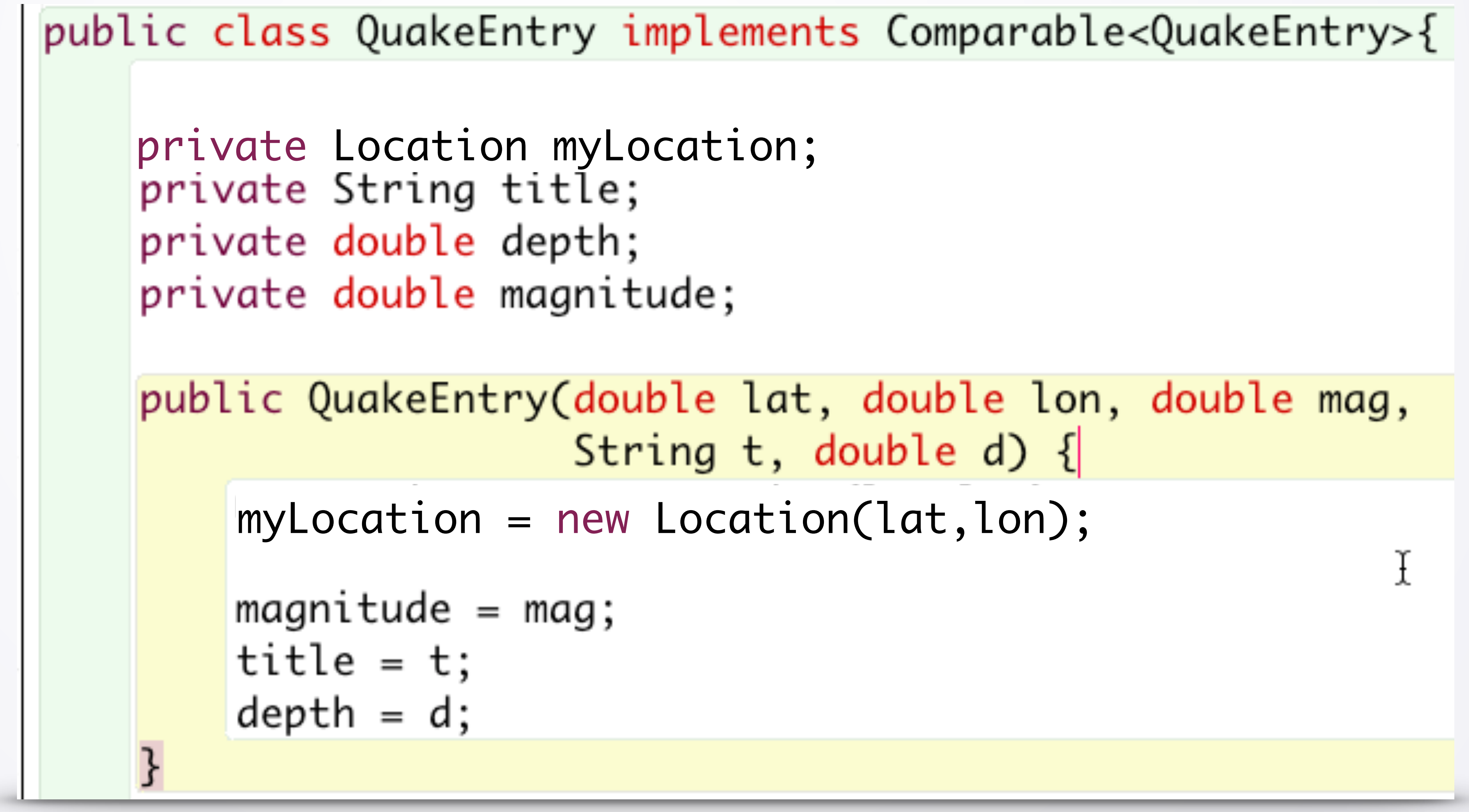
Has-A and Uses-A Relationships

- QuakeEntry object created by Parser



Has-A and Uses-A Relationships

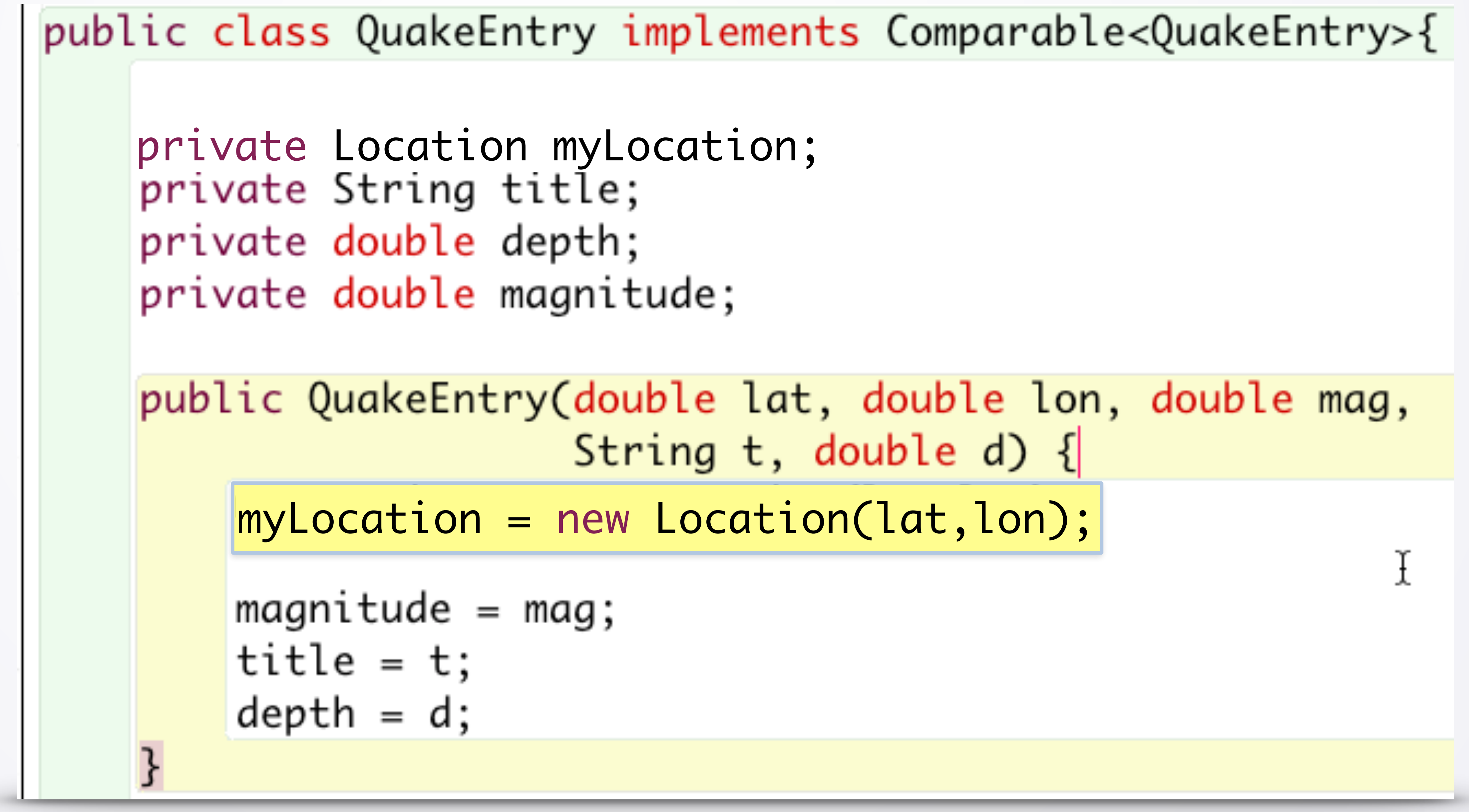
- QuakeEntry object created by Parser
    - Location constructor called from QuakeEntry constructor



```
public class QuakeEntry implements Comparable<QuakeEntry>{

    private Location myLocation;
    private String title;
    private double depth;
    private double magnitude;

    public QuakeEntry(double lat, double lon, double mag,
                      String t, double d) {
        myLocation = new Location(lat,lon);

        magnitude = mag;
        title = t;
        depth = d;
    }
```

Has-A and Uses-A Relationships

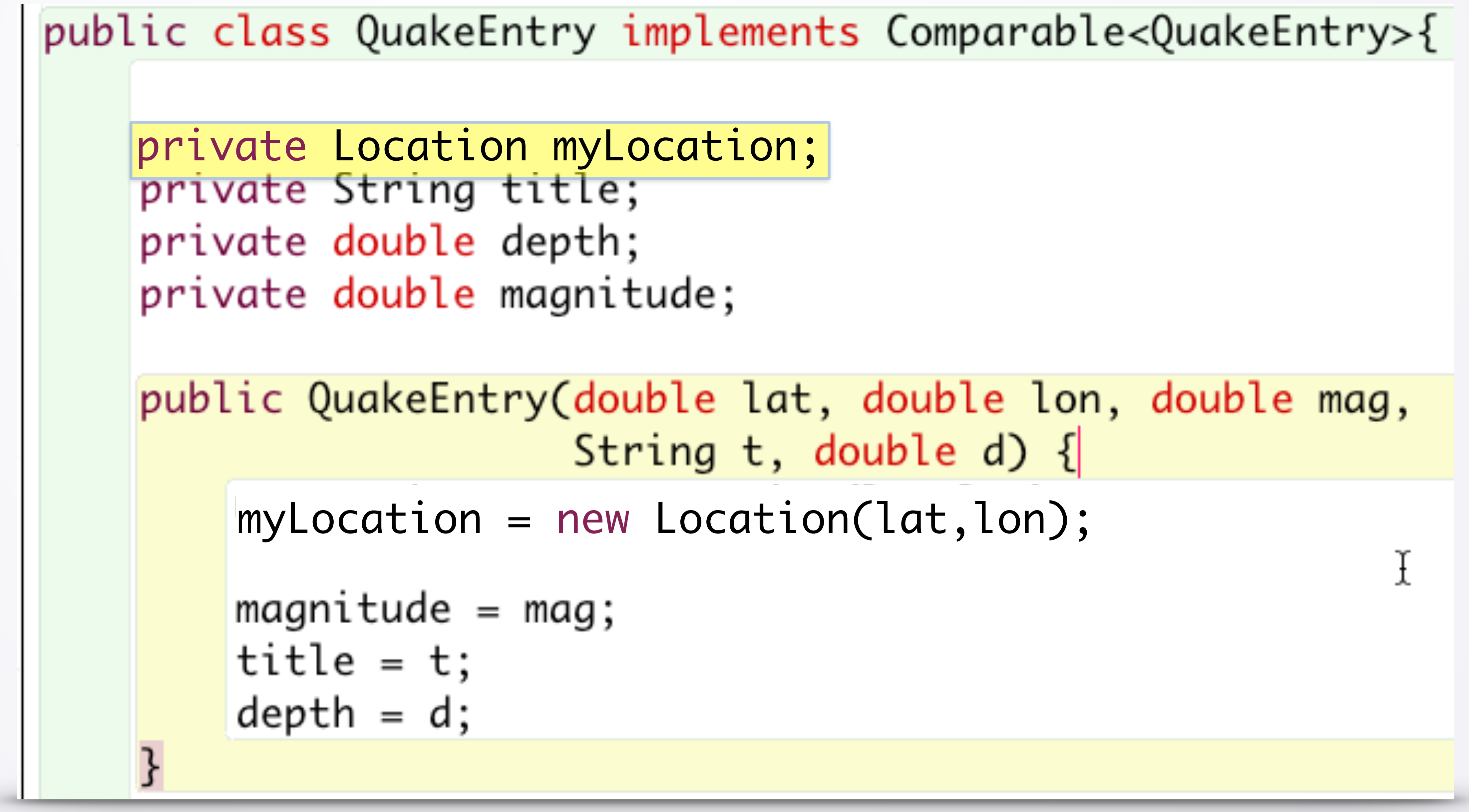
- QuakeEntry object created by Parser
  - Location constructor called from QuakeEntry constructor



```
public class QuakeEntry implements Comparable<QuakeEntry>{

    private Location myLocation;
    private String title;
    private double depth;
    private double magnitude;

    public QuakeEntry(double lat, double lon, double mag,
                          String t, double d) {
        myLocation = new Location(lat,lon);

        magnitude = mag;
        title = t;
        depth = d;
    }
```

Has-A and Uses-A Relationships

- QuakeEntry object created by Parser
    - Location constructor called from QuakeEntry constructor
    - myLocation is instance field in QuakeEntry: Has-A relationship

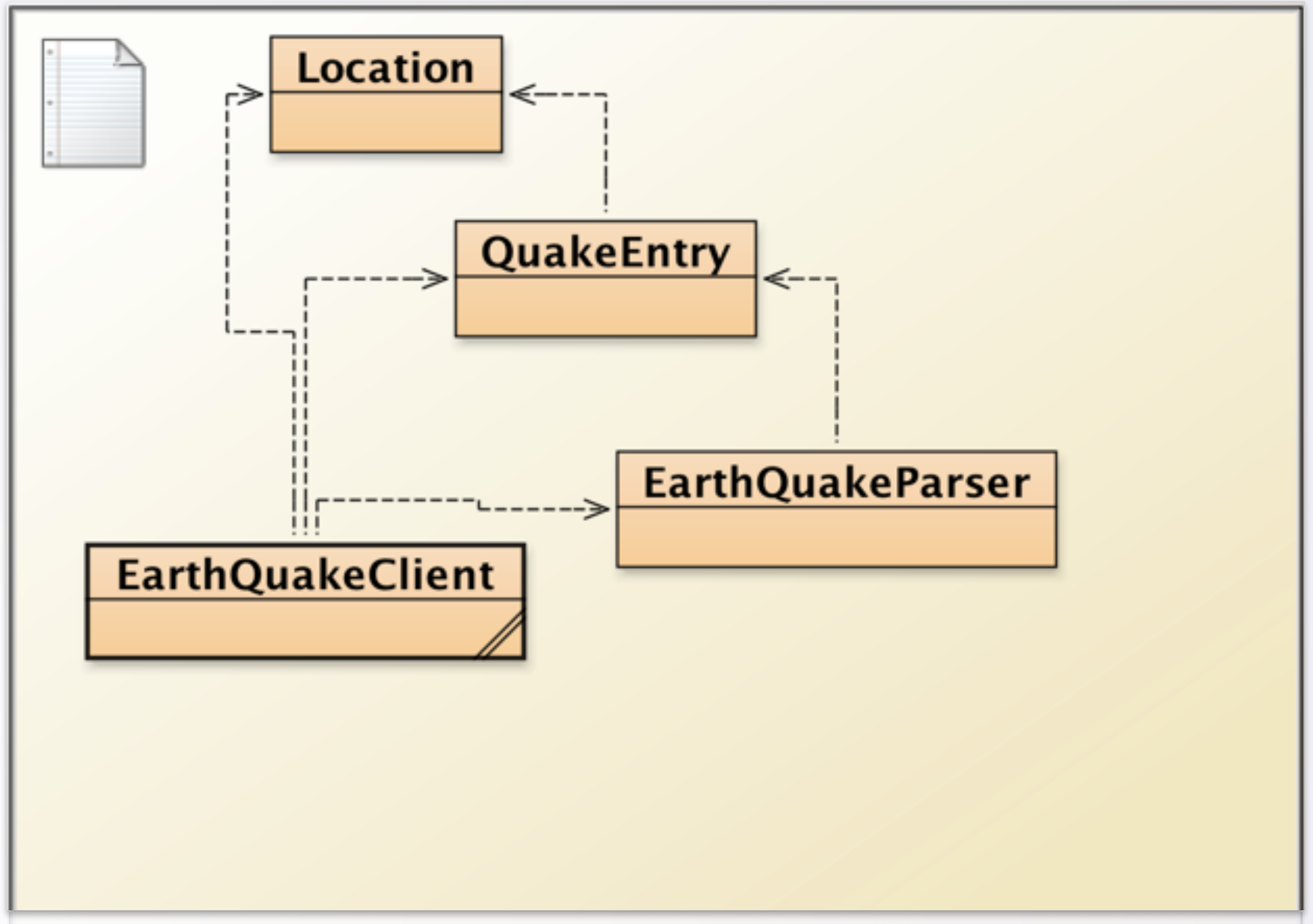


```
public class QuakeEntry implements Comparable<QuakeEntry>{

    private Location myLocation;
    private String title;
    private double depth;
    private double magnitude;

    public QuakeEntry(double lat, double lon, double mag,
                      String t, double d) {
        myLocation = new Location(lat,lon);

        magnitude = mag;
        title = t;
        depth = d;
    }
```

Duke UNIVERSITY

Has-A and Uses-A Relationships

- QuakeEntry object created by Parser
  - Location constructor called from QuakeEntry constructor
  - myLocation is instance field in QuakeEntry: Has-A relationship
- Location used in EarthQuakeClient
  - `.distanceTo`(..)
- QuakeEntry used in EarthQuakeClient too
  - `getMagnitude`(..)


Location
QuakeEntry
EarthQuakeParser
EarthQuakeClient