Assignment Two

Brae & Emily & Tom

May 9, 2018

CSSE2002: Programming in the Large

Getting Started

1. Download `supplied.zip` and `doc.zip` from blackboard
2. Load the extracted contents of `supplied.zip` into IntelliJ - refer to week two slides

Getting Started

1. Download supplied.zip and doc.zip from blackboard
2. Load the extracted contents of supplied.zip into IntelliJ - refer to week two slides

The supplied code **will not compile** when initially loaded. Some code needs to be written (by you) before it will compile. This is stated in the task sheet.

Where To Start?

Start with classes that don't have dependencies (e.g. Pair)

MapIO is hard   (IO has not yet been covered)

Start with classes top of the hierarchy (e.g. Thing)

The order specified in the task sheet is a good starting point

Where To Start?

Start with classes that don't have dependencies (e.g. Pair)

MapIO is hard (IO has not yet been covered)

Start with classes top of the hierarchy (e.g. Thing)

The order specified in the task sheet is a good starting point

Where To Start?

Start with classes that don't have dependencies (e.g. Pair)

MapIO is hard (IO has not yet been covered)

Start with classes top of the hierarchy (e.g. Thing)

The order specified in the task sheet is a good starting point
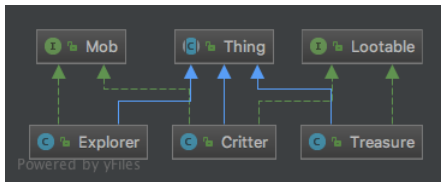
Where To Start?

Start with classes that don't have dependencies (e.g. Pair)

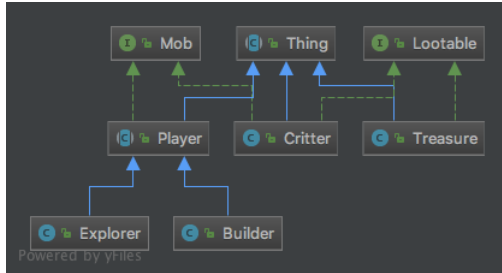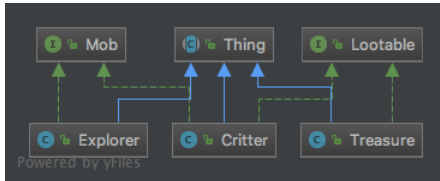**MapIO is hard** (IO has not yet been covered)

Start with classes top of the hierarchy (e.g. Thing)

The order specified in the task sheet is a good starting point

What Has Changed?

What Has Changed?

Joel's Solution

In Joel's solution he has used `Collections.unmodifiableList` and `Collections.unmodifiableMap` when returning a instance variables. Do I need to know this?

Joel's Solution

In Joel's solution he has used `Collections.unmodifiableList` and `Collections.unmodifiableMap` when returning a instance variables. Do I need to know this?

No. You were not expected to know these methods.

Joel's Solution

In Joel's solution he has used `Collections.unmodifiableList`
and `Collections.unmodifiableMap` when returning a instance
variables. Do I need to know this?

No. You were not expected to know these methods.

```
return Collections.unmodifiableList(this.contents);

return new ArrayList<Thing>(this.contents);

List<Thing> results = new ArrayList<Thing>();
for (Thing thing : this.contents) {
        results.add(thing);
}
return results;
```

Exceptions

Exceptions do not need to be complicated.

```
public class CrawlException extends Exception {}

public class NullRoomException extends CrawlException
    {}
```

There is no need to add a constructor or any extra methods to exceptions.