CS2030S

Programming Methodology II

Lab 06

Actually

Actually

Types for Actually

```
public abstract <R> Actually<R> transform(Immutator<? extends R, ? super T> f);
public abstract T unwrap() throws Exception;
public abstract <U extends T> T except(Constant<? extends U> c);
public abstract void finish(Action<? super T> a);
public abstract <U extends T> T unless(U u);
public abstract <U> Actually<U> next(Immutator<? extends Actually<? extends U>, ? super T> c);
```

Subclasses

- `Success<T> extends Actually<T>`
- `Failure extends Actually<Object>`

Actually

Success

Transform

```
try {
  return /* ok */;
} catch(Exception e) {
  return /* err */;
}
```

Actually

Success

Next

```
try {
  @SuppressWarnings("unchecked")
  Actually<U> res = ...;
  return res;
} catch(Exception e) {
  return /* err */;
}
```

Actually

Success

Unwrapping

```
public T unwrap() {
  return this.t;
}
public <U extends T> T except(Constant<? extends U> c) {
  return this.t;
}
public <U extends T> T unless(U u) {
  return this.t;
}
```

Actually

Failure

Transform

```
return /* err */; // already error
```

Actually

Failure

Next

```
return /* err */; // already error
```

Actually

Failure

Unwrapping

```
public Object unwrap() throws Exception {
  throw this.e;
}
public <U> U except(Constant<? extends U> c) {
  return c.init;
}
public <U> U unless(U u) {
  return u;
}
```

Documentation

Documentation

Generating Documentation

```
javadoc -quiet -private -d docs cs2030s/fp/Lazy.java
```

Note

- `-quiet`: only errors and warnings are shown
- `-private`: include documentation from all fields/methods
- `-d docs`: put the generated HTML in a subdirectory called `docs`

Lab 6

Lab 6

Goal

Goal of Lab 6

- Extends `cs2030s.fp` with `Lazy<T>` and `Memo<T>`
  - `Lazy<T>`: evaluate when needed
  - `Memo<T>`: never repeat yourself
- Practice using `Actually<T>`
- Practice using lambdas and lazy evaluation

Note

- `Memo<T>` will be the basis for Lab 7 and Lab 8.

Lab 6

Goal
Condition

Boolean Condition