Class 16: Mutual Exclusion

6.102 — Software Construction
Spring 2024

Exercise: yellkey.com/**year**

Nanoquiz: yellkey.com/**city**

Warmup

Open `Bank1.ts`

Look at the `Bank` ADT, particularly the operations `transfer()` and `audit()`

- what is `audit()` actually doing? (from the point of view of ADT theory)

Look at `main()`, and run the code:
`npm run Bank1` should print three successful bank audits

TODO: change

```
    bank.transfer('Effie', 'Chip', 10);
```

into a transfer that causes the last bank audit to FAIL

- hint: it should be a different kind of illegal transfer

Async methods

In `Bank2.ts`:

- look at `main()`
- `npm run Bank2`
  - should be 1 transfer and 1 successful audit
- look at new private helper `record()` that writes a log on disk
  - notice that it uses `await`

TODO:

- uncomment the calls to `record()` in `transfer()`
- fix the errors so that `transfer()` is now `async`
- `npm run Bank2` — what happens?

Making a promise

Look at `timeout.ts`:

- what is `deferred`, and how is it used?

Making a promise

Look at `timeout.ts`:

- what is `deferred`, and how is it used?

In `Bank3.ts`:

- look at `untilAccountExists` and `main()` and `npm run Bank3` (not working yet)
- notice the new rep field `expectedAccounts`
- fix the TODOs in both `untilAccountExists` and `makeAccount` so that `npm run Bank3` works