IRON HACK

Stubs & Fakes

Profanity filter

https://www.flickr.
com/photos/thatrileygirl/18639377

Profanity filter

Implement a plot profanity filter. Look for a movie in IMDB by title and substitute the word fuck in the plot for f**k

What’s the smallest test?

How do we test that we filter the word ‘dasijaosidjaosidjaosid’?

Sorry guys, I’ve unplugged the router

Stubs

Stubs provide canned answers to calls made during the test, usually not responding at all to anything outside what's programmed in for the test.

Stubs

```
class MovieProfanityFilter
  def initialize(movie_searcher)
    @movie_searcher = movie_searcher
  end

  def filter_synopsis(movie_title)
    synopsis = @movie_searcher.search_by_title(movie_title)[0][:
synopsis]
    synopsis.gsub("fuck", "f**k")
  end
end

class IMDBSearcher
  def search_by_title(movie_title)
    Imdb::Search.new(movie_title).movies.map do |movie|
      {
        title: movie.title,
        synopsis: movie.plot_synopsis
      }
    end
  end
end
```

```
let (:swear_word_searcher) do
  instance_double("IMDBSearcher",
              :search_by_title => [{synopsis: "The best fucking movie ever"}])
end

let (:nice_word_searcher) do
  instance_double("IMDBSearcher",
                :search_by_title => [{synopsis: "The best movie ever"}])
end

it "filters the title if it contains the f word" do
      expect(MovieProfanityFilter.new(swear_word_searcher).filter_synopsis
("best"))
      .to eq("The best f**king movie ever")
end

it "does not filter the title if it does not contain a swear word" do
                  expect(MovieProfanityFilter.new(nice_word_searcher).
filter_synopsis("best"))
      .to eq("The best movie ever")
end
```

Fake

Fake objects actually have working implementations, but usually take some shortcut which makes them not suitable for production (an InMemoryTestDatabase is a good example).

```ruby
class MovieSearcherFake
  def initialize(synopsis)
    @synopsis = synopsis
  end

  def search_by_title(title)
    [{ synopsis: @synopsis }]
  end
end
```

```ruby
let (:swear_word_searcher) {MovieSearcherFake.new("The best fucking movie ever")}
let (:nice_word_searcher) {MovieSearcherFake.new("The best movie ever")}

it "filters the title if it contains the f word" do
    expect(MovieProfanityFilter.new(swear_word_searcher).filter_synopsis("best"))
      .to eq("The best f**king movie ever")
end

it "does not filter the title if it does not contain a swear word" do
    expect(MovieProfanityFilter.new(nice_word_searcher).filter_synopsis("best"))
      .to eq("The best movie ever")
end
```

Exercise

We are provided a class that is called EmailProvider and has a method get_subjects that returns all subjects of a given email account.

TDD a class that has a method that receives a list of words and checks that all words are at least in one subject

Always wrap dependencies

https://www.flickr.
com/photos/ramsd/7228178876