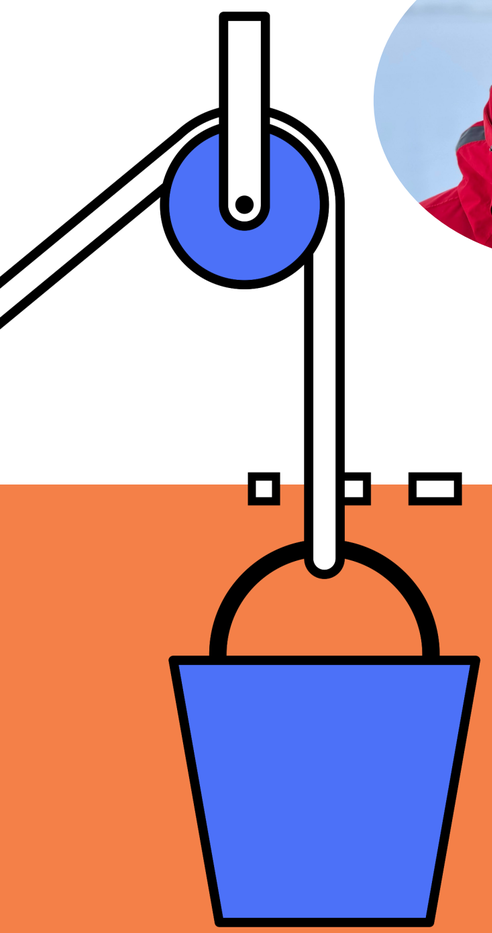
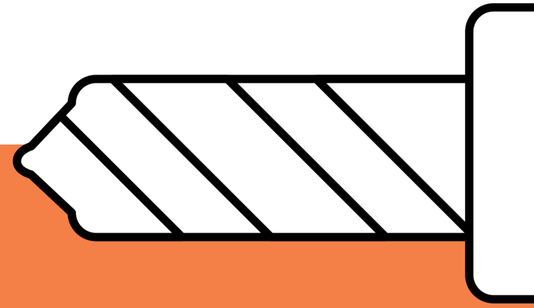
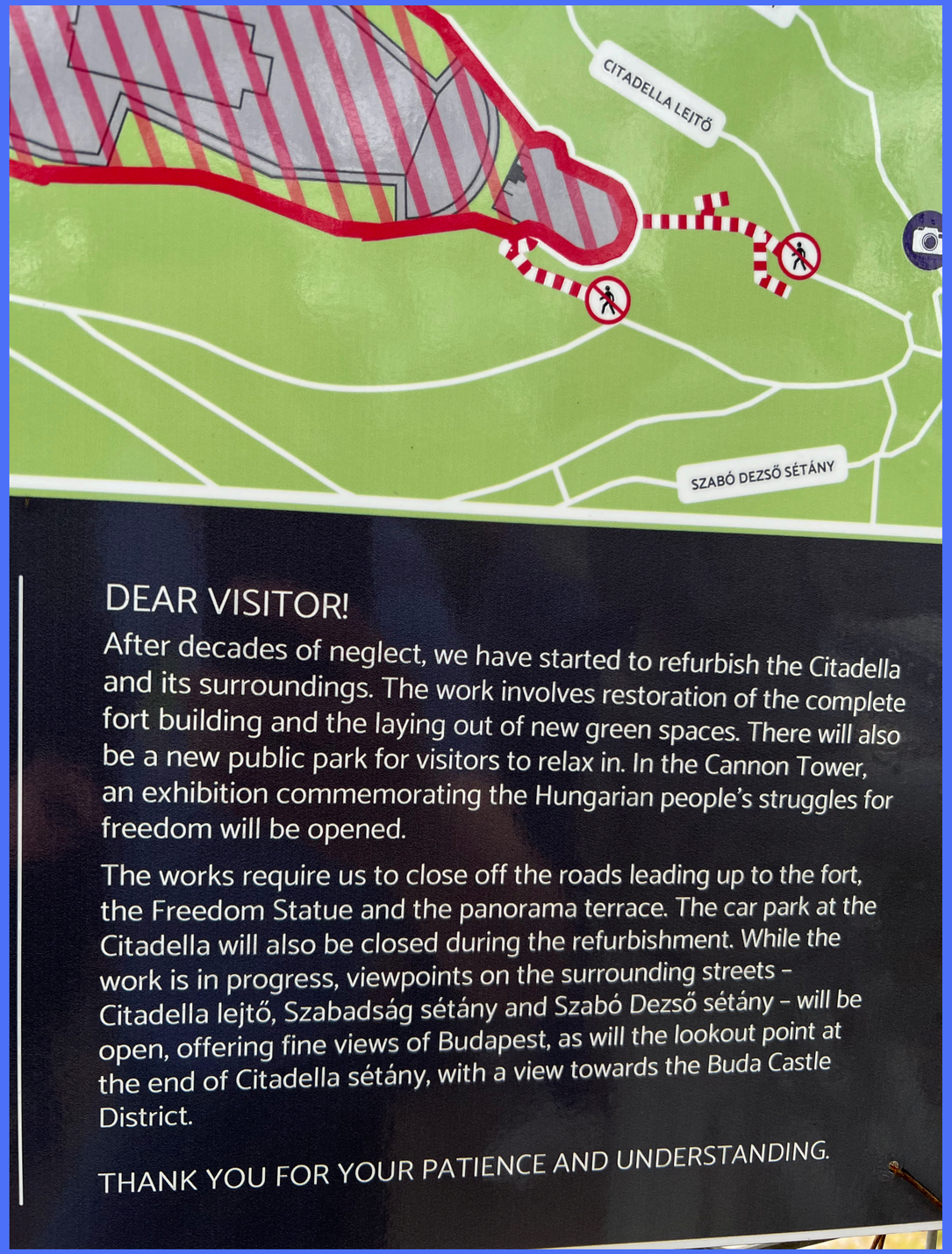
Lars Eckart, Technical Coach, Co-Maintainer ApprovalTests.Java

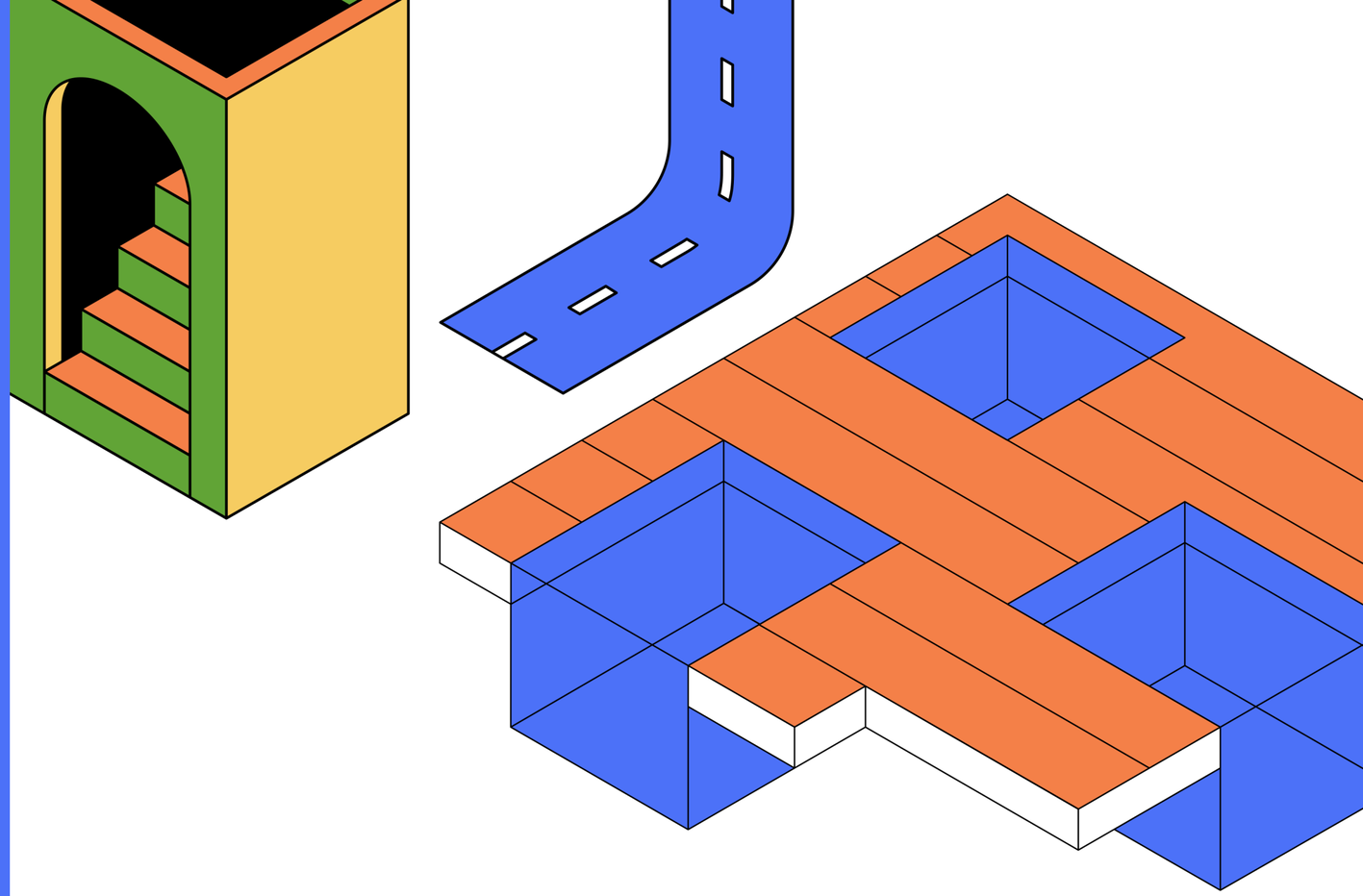




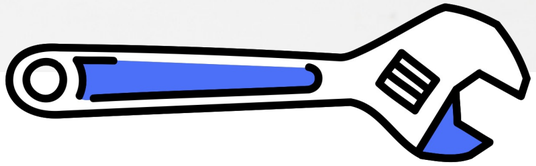
Approval Tests

How I learned about Approval Tests
and why I want you to know about it too

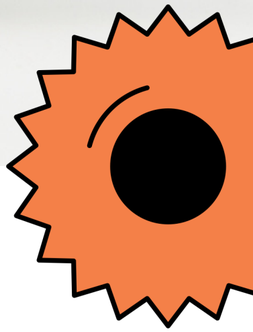




from practice to production



Regular Unit Test



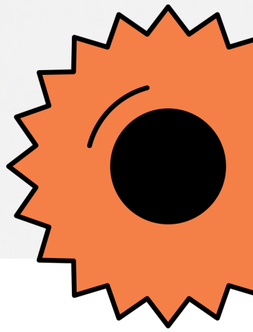
Specific Name

Arrange Act Assert

assertEquals(expected, actual)

```java
class GildedRoseTest {

    @Test
    void aged_brie_increases_in_quality_the_older_it_gets() {
        Item[] items = new Item[]{new Item( name: "Aged Brie", sellIn: 7, quality: 10)};
        GildedRose app = new GildedRose(items);

        app.updateQuality();

        assertEquals(11, items[0].quality);
    }
}
```

```
assertEquals(11, items[0].quality);
```



Approval Test

Name

Arrange Act Assert

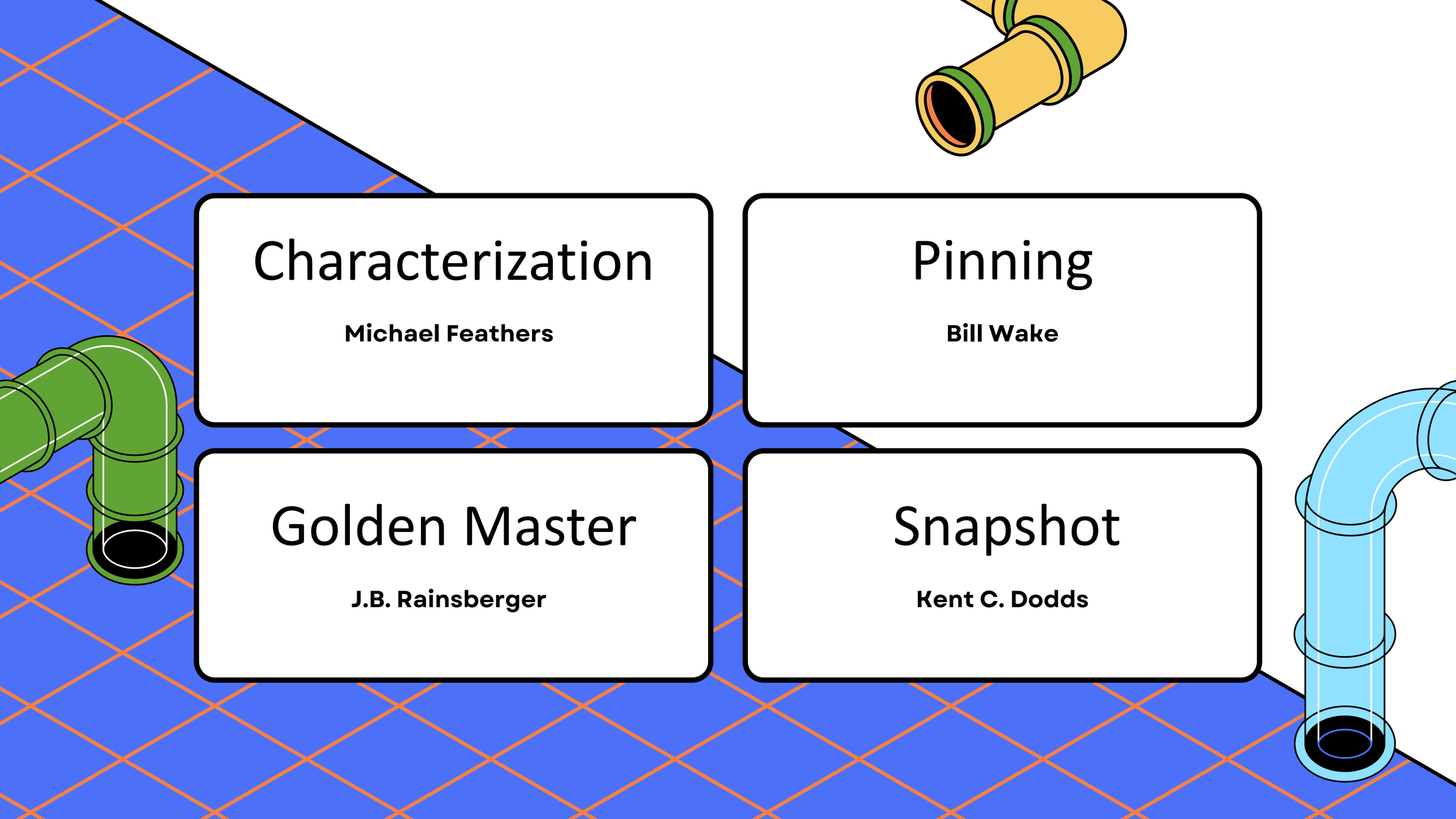
Verify(actual)

```java
@Test
void approval() {
    Item[] items = new Item[]{new Item( name: "Aged Brie", sellIn: 7, quality: 10)};
    GildedRose app = new GildedRose(items);

    app.updateQuality();

    Approvals.verify(items[0]);
}
```

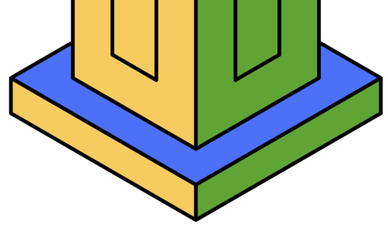
GildedRoseTest.approval.approved.txt

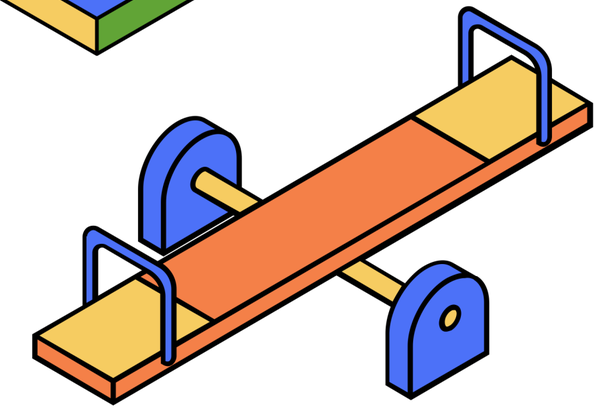
```
1    Aged Brie, 6, 11
```

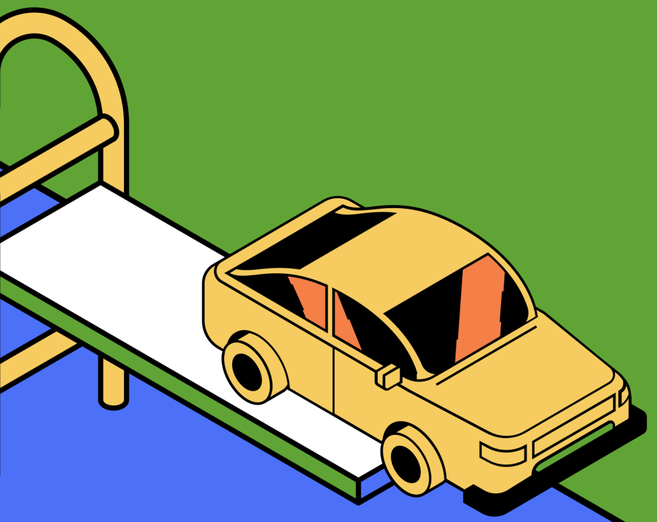
```
Approvals.verify(items[0]);
```


Characterization
Michael Feathers
Pinning
Bill Wake
Golden Master
J.B. Rainsberger
Snapshot
Kent C. Dodds

https://approvaltests.com







Tomorrow, 14 o’clock,
Green Tent
Writing Documentation

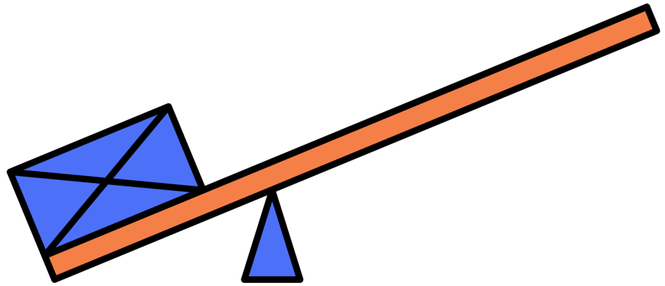
Llewellyn Falco

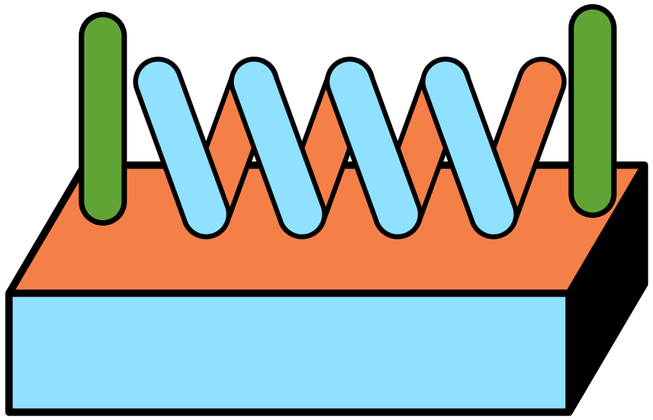
Approvals.verify

String, Object, Map, ResultSet, File

Approvals.verifyAll

Arrays, Iterable





JsonApprovals

AwtApprovals

String, Image, Component, Paintable

CombinationApprovals

verifyAllCombinations

Acceptance Tests

01

Keycloak CLI

```
Keycloak - Open Source Identity and Access Management

Find more information at: https://www.keycloak.org/docs/latest

Usage:

kc.bat [OPTIONS] [COMMAND]

Use this command-line tool to manage your Keycloak cluster.
Make sure the command is available on your "PATH" or prefix it with "./" (e.g.:
"./kc.bat") to execute from the current folder.

Options:

-cf, --config-file <file>
                    Set the path to a configuration file. By default, configuration properties are
                      read from the "keycloak.conf" file in the "conf" directory.
-h, --help          This help message.
-v, --verbose       Print out error details when running this command.
-V, --version       Show version information

Commands:

  build                 Creates a new and optimized server image.
  start                 Start the server.
  start-dev             Start the server in development mode.
  export                Export data from realms to a file or directory.
  import                Import data from a directory or a file.
  show-config           Print out the current configuration.
  tools                 Utilities for use and interaction with the server.
    completion          Generate bash/zsh completion script for kc.bat.

Examples:

  Start the server in development mode for local development or testing:

      $ kc.bat start-dev

  Building an optimized server runtime:

      $ kc.bat build <OPTIONS>
```

02

Betslip sharing

03

MicroService API

```java
@Test
void storeAndRetrieveMovies() throws JsonProcessingException {
    webTestClient.post() RequestBodyUriSpec
            .uri( uri: "/api/movies") RequestBodySpec
            .contentType(MediaType.APPLICATION_JSON)
            .bodyValue("{\"title\": \"The Super Mario Bros. Movie\", \"year\": 2023}") R
            .exchange();

    webTestClient.post() RequestBodyUriSpec
            .uri( uri: "/api/movies") RequestBodySpec
            .contentType(MediaType.APPLICATION_JSON)
            .bodyValue("{\"title\": \"The Matrix\", \"year\": 1999}") RequestHeadersSpec<c
            .exchange();

    String response = webTestClient.get() RequestHeadersUriSpec<capture of ?>
            .uri( uri: "/api/movies") capture of ?
            .exchange() ResponseSpec
            .returnResult(String.class).getResponseBody().blockFirst();

    Object json = mapper.readValue(response, Object.class);
    String indented = mapper.writerWithDefaultPrettyPrinter().writeValueAsString(json);

    Approvals.verify((indented));
}
```

```java
@Test
@Launch({ "--help" })
void testHelp(LaunchResult result) {
    CLIResult cliResult = (CLIResult) result;
    assertHelp(cliResult);
}

@Test
@Launch({ "-h" })
void testHelpShort(LaunchResult result) {
    CLIResult cliResult = (CLIResult) result;
    assertHelp(cliResult);
}
```

```
Keycloak - Open Source Identity and Access Management

Find more information at: https://www.keycloak.org/docs/latest

Usage:

kc.bat [OPTIONS] [COMMAND]

Use this command-line tool to manage your Keycloak cluster.
Make sure the command is available on your "PATH" or prefix it with "./" (e.g.:
"./kc.bat") to execute from the current folder.

Options:

-cf, --config-file <file>
                    Set the path to a configuration file. By default, configuration propert
                      read from the "keycloak.conf" file in the "conf" directory.
-h, --help          This help message.
-v, --verbose       Print out error details when running this command.
-V, --version       Show version information

Commands:

  build                 Creates a new and optimized server image.
  start                 Start the server.
  start-dev             Start the server in development mode.
  export                Export data from realms to a file or directory.
  import                Import data from a directory or a file.
  show-config           Print out the current configuration.
  tools                 Utilities for use and interaction with the server.
    completion          Generate bash/zsh completion script for kc.bat.

Examples:

  Start the server in development mode for local development or testing:

      $ kc.bat start-dev

```

Use cases from production

01

Keycloak CLI

```
Keycloak - Open Source Identity and Access Management

Find more information at: https://www.keycloak.org/docs/latest

Usage:

kc.bat [OPTIONS] [COMMAND]

Use this command-line tool to manage your Keycloak cluster.
Make sure the command is available on your "PATH" or prefix it with "./" (e.g.:
"./kc.bat") to execute from the current folder.

Options:

-cf, --config-file <file>
                    Set the path to a configuration file. By default, configuration properties are
                      read from the "keycloak.conf" file in the "conf" directory.
-h, --help          This help message.
-v, --verbose       Print out error details when running this command.
-V, --version       Show version information

Commands:

  build                 Creates a new and optimized server image.
  start                 Start the server.
  start-dev             Start the server in development mode.
  export                Export data from realms to a file or directory.
  import                Import data from a directory or a file.
  show-config           Print out the current configuration.
  tools                 Utilities for use and interaction with the server.
    completion          Generate bash/zsh completion script for kc.bat.

Examples:

  Start the server in development mode for local development or testing:

      $ kc.bat start-dev

  Building an optimized server runtime:

      $ kc.bat build <OPTIONS>
```

02

Betslip sharing

03

MicroService API

```
@Test
void storeAndRetrieveMovies() throws JsonProcessingException {
    webTestClient.post() RequestBodyUriSpec
            .uri( uri: "/api/movies") RequestBodySpec
            .contentType(MediaType.APPLICATION_JSON)
            .bodyValue("{\"title\": \"The Super Mario Bros. Movie\", \"year\": 2023}") R
            .exchange();

    webTestClient.post() RequestBodyUriSpec
            .uri( uri: "/api/movies") RequestBodySpec
            .contentType(MediaType.APPLICATION_JSON)
            .bodyValue("{\"title\": \"The Matrix\", \"year\": 1999}") RequestHeadersSpec<c
            .exchange();

    String response = webTestClient.get() RequestHeadersUriSpec<capture of ?>
            .uri( uri: "/api/movies") capture of ?
            .exchange() ResponseSpec
            .returnResult(String.class).getResponseBody().blockFirst();

    Object json = mapper.readValue(response, Object.class);
    String indented = mapper.writerWithDefaultPrettyPrinter().writeValueAsString(json);

    Approvals.verify((indented));
}
```

```java
public BufferedImage buildImage(BetSlipImageGenerationRequest request)

    BufferedImage header = buildHeader(request, getWidth(), getHeaderHe
    BufferedImage body = buildBody(request, getWidth(), getBodyHeight()
    BufferedImage footer = buildFooter(request, getWidth(), getFooterHe

    Color bgColor = Platform.SUGARHOUSE.equals(request.platform())
                    ? COLOR_SUGAR_HOUSE_BG
                    : COLOR_BET_RIVERS_BG;

    return new BufferedImageBuilder(getWidth(), getHeight(), BufferedIm
                .addImage(new ImageElement(getClass().getResource( name: IMAG
                .addOverlay(bgColor, AlphaComposite.getInstance(AlphaComposite.SRC_ATOP, alpha: 0.85f))
                .addSection(header, header.getWidth(), header.getHeight(), x: 0, y: 0)
                .addSection(body, body.getWidth(), body.getHeight(), x: 0, header.getHeight())
                .addSection(footer, footer.getWidth(), footer.getHeight(), x: 0, y: header.getHeight()
                .build();
}
```

Use cases from production

01

Keycloak CLI

```
Keycloak - Open Source Identity and Access Management

Find more information at: https://www.keycloak.org/docs/latest

Usage:

kc.bat [OPTIONS] [COMMAND]

Use this command-line tool to manage your Keycloak cluster.
Make sure the command is available on your "PATH" or prefix it with "./" (e.g.:
"./kc.bat") to execute from the current folder.

Options:

-cf, --config-file <file>
                    Set the path to a configuration file. By default, configuration properties are
                      read from the "keycloak.conf" file in the "conf" directory.
-h, --help          This help message.
-v, --verbose       Print out error details when running this command.
-V, --version       Show version information

Commands:

  build                 Creates a new and optimized server image.
  start                 Start the server.
  start-dev             Start the server in development mode.
  export                Export data from realms to a file or directory.
  import                Import data from a directory or a file.
  show-config           Print out the current configuration.
  tools                 Utilities for use and interaction with the server.
    completion          Generate bash/zsh completion script for kc.bat.

Examples:

  Start the server in development mode for local development or testing:

      $ kc.bat start-dev

  Building an optimized server runtime:

      $ kc.bat build <OPTIONS>
```

02

Betslip sharing

03

MicroService API

```
@Test
void storeAndRetrieveMovies() throws JsonProcessingException {
    webTestClient.post() RequestBodyUriSpec
            .uri( uri: "/api/movies") RequestBodySpec
            .contentType(MediaType.APPLICATION_JSON)
            .bodyValue("{\"title\": \"The Super Mario Bros. Movie\", \"year\": 2023}") R
            .exchange();

    webTestClient.post() RequestBodyUriSpec
            .uri( uri: "/api/movies") RequestBodySpec
            .contentType(MediaType.APPLICATION_JSON)
            .bodyValue("{\"title\": \"The Matrix\", \"year\": 1999}") RequestHeadersSpec<c
            .exchange();

    String response = webTestClient.get() RequestHeadersUriSpec<capture of ?>
            .uri( uri: "/api/movies") capture of ?
            .exchange() ResponseSpec
            .returnResult(String.class).getResponseBody().blockFirst();

    Object json = mapper.readValue(response, Object.class);
    String indented = mapper.writerWithDefaultPrettyPrinter().writeValueAsString(json);

    Approvals.verify((indented));
}
```

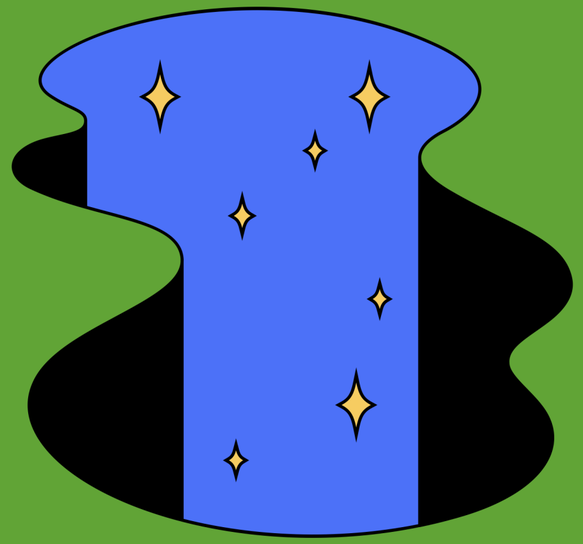
```java
@Test
void storeAndRetrieveMovies() throws JsonProcessingException {
    storeMovie( body: "{\"title\": \"The Super Mario Bros. Movie\
    storeMovie( body: "{\"title\": \"The Matrix\", \"year\": 1999

    String response = webTestClient.get() RequestHeadersUriSpec<cap
            .uri( uri: "/api/movies") capture of ?
            .exchange() ResponseSpec
            .returnResult(String.class).getResponseBody().blockF

    Object json = mapper.readValue(response, Object.class);
    String indented = mapper.writerWithDefaultPrettyPrinter().wr

    Approvals.verify((indented));
}
```

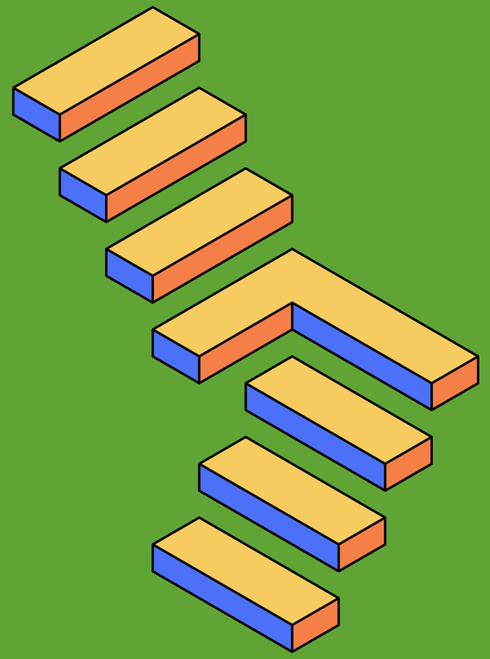
MovieAppApplicationTests.storeAndRetrieveMovies.approved.txt

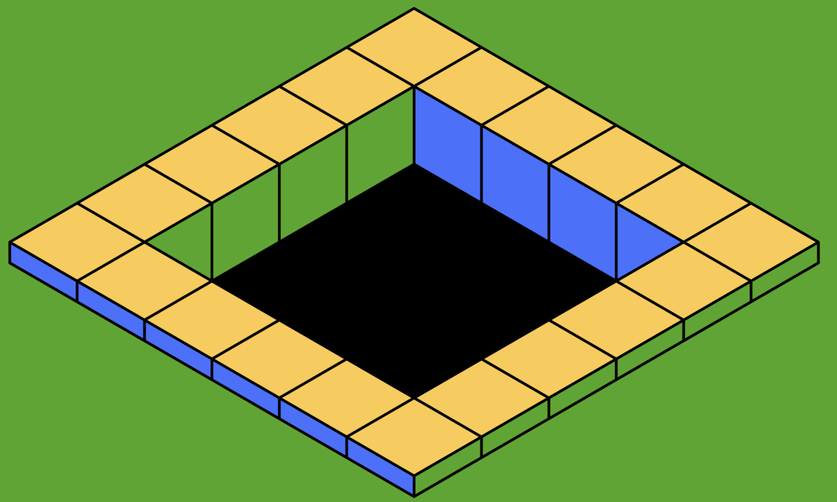
```
1  [ {
2    "title" : "The Super Mario Bros. Movie",
3    "year" : 2023
4  }, {
5    "title" : "The Matrix",
6    "year" : 1999
```

Give it a try!

Thank You!







EMAIL

lars.eckart@hey.com

GitHub

https://github.com/LarsEckart

Web

https://larseckart.com/