CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Exercise 3: Functional Testing

Gregory Gay
DIT635 - February 14, 2020

Finish In-Class Activity (Lec 7) First!

Airport Connection Check

- The airport connection check is part of a travel reservation system. It checks the validity of a single connection between two flights in an itinerary.
  - If the arrival airport of Flight A differs from the departure airport of Flight B, the connection is invalid.
  - If the departure time of Flight B is too close to the arrival time of Flight A, the connection is invalid.
  - If an airport doesn’t exist, the connection is invalid…

Airport Connection Check

```
validConnection(Flight arrivingFlight, Flight departingFlight)
                returns ValidityCode
```

A Flight is a data structure consisting of:

- A unique identifying flight code (string, three characters followed by four numbers).
- The originating airport code (three character string).
- The scheduled departure time (in universal time).
- The destination airport code (three character string).
- The scheduled arrival time (in universal time).

There is also a flight database, where each record contains:

- Three-letter airport code (three character string).
- Airport country (two character string).
- Minimum domestic and international connection times (integer, minimum number of minutes that must be allowed for flight connections).

Airport Connection Check

There is also a flight database, where each record contains:

- Three-letter airport code (three character string).
- Airport country (two character string).
- Minimum connection time (integer, minimum number of minutes that must be allowed for flight connections).

`ValidityCode` is an integer with value:

- 0 for OK
- 1 for invalid airport code
- 2 for a connection that is too short
- 3 for flights that do not connect (arrivingFlight does not land in the same location as departingFlight)
- 4 for any other errors (malformed input or any other unexpected errors).

Your Task

- In order to design requirements-based test cases, perform category-partition testing using this specification for the validConnection function.
  - Identify the parameters that can be controlled through testing.
  - Identify testing categories (controllable items) for each parameter.
  - Identify representative input values for each category.

Hints

Process of writing functional tests:

1. Refine requirements so they are testable.
2. Divide the software into independently testable features.
   a. `validConnection(...)` is a testable feature.
3. Identify the explicit and implicit parameters of each feature.
   a. Those passed directly to the function and environmental factors that influence the outcome.

Hints

Process of writing functional tests:

4. For each parameter, identify factors that can influence the test outcome (categories).
5. Divide each category into representative values.
6. Form abstract test cases by choosing a value for each category.

Hints

- You've been given a testable feature.
- Complete steps 3-5 (identify parameters, split into categories of input, partition categories into representative values).
- Function has two explicit parameters (arriving flight, departing flight) and one implicit (airport database).
  - Flight has multiple fields (potential categories)
  - Database records have multiple fields.
  - Remember that parameters and categories can interact. This must be accounted for in the categories.

Hints

- Think about how arrival time, departure time, and the minimum connection time interact.
- Consider that domestic and international connection times can differ in length.
- Consider how the database contents can influence behavior.
- Think about how input can be invalid or malformed (and be explicit - don’t just list “invalid input” but give clear examples).

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY