University of
South Australia

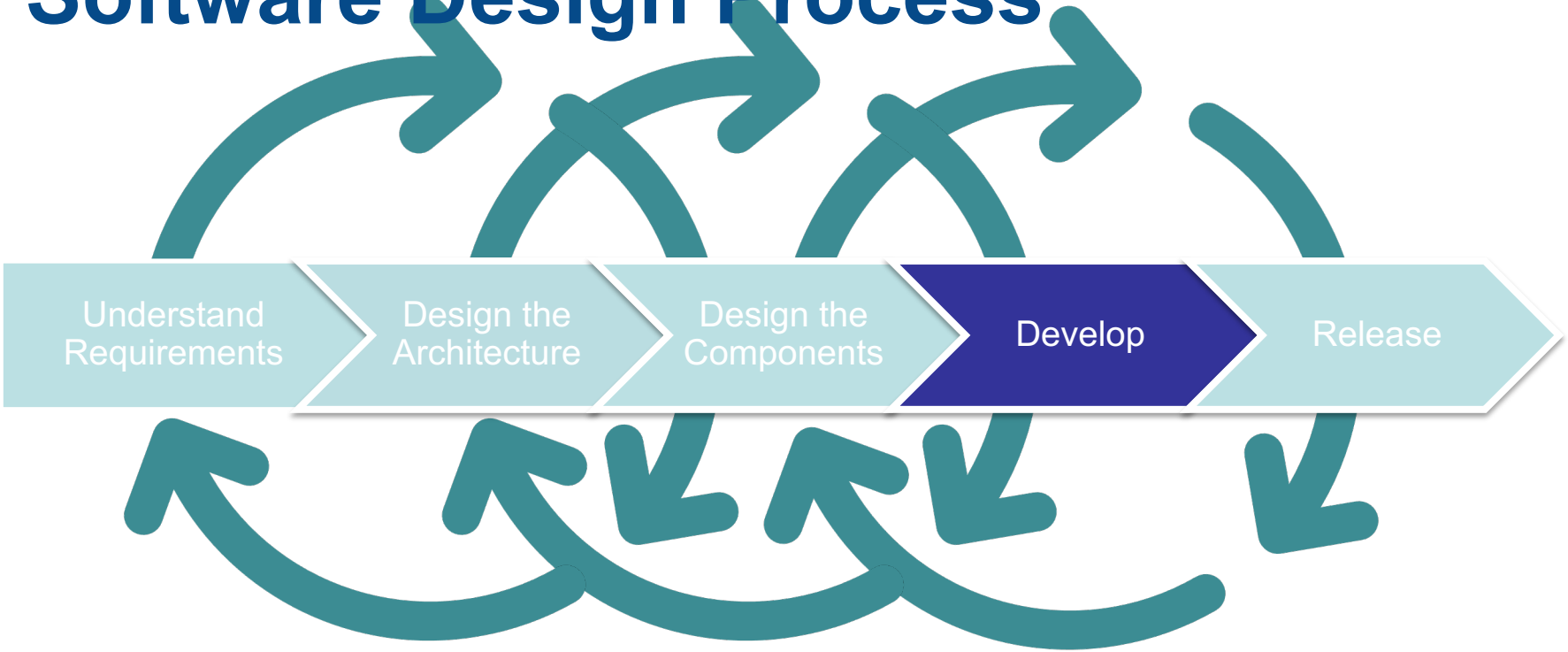
INFS 2044

Practical 2 Answers

Preparation

- Watch Unit Testing and Test Driven Development in Python on LinkedIn Learning
- Read the required readings
- Watch the Week 11 Lecture

Software Design Process



Learning Objectives

- Apply unit testing and test-driven development principles (CO4)

Task 1. Install pytest

- Install pytest

Open a command prompt (or terminal on MacOS/Linux) and run:

```
pip install --user pytest
```

Alternatively, follow the pytest install instructions for your platform/IDE.

Alternative Installation Instructions

To install:
C:\path\to\python.exe –m pip install pytest

To Run:
C:\path\to\python.exe –m pytest <file_containing_tests.py>

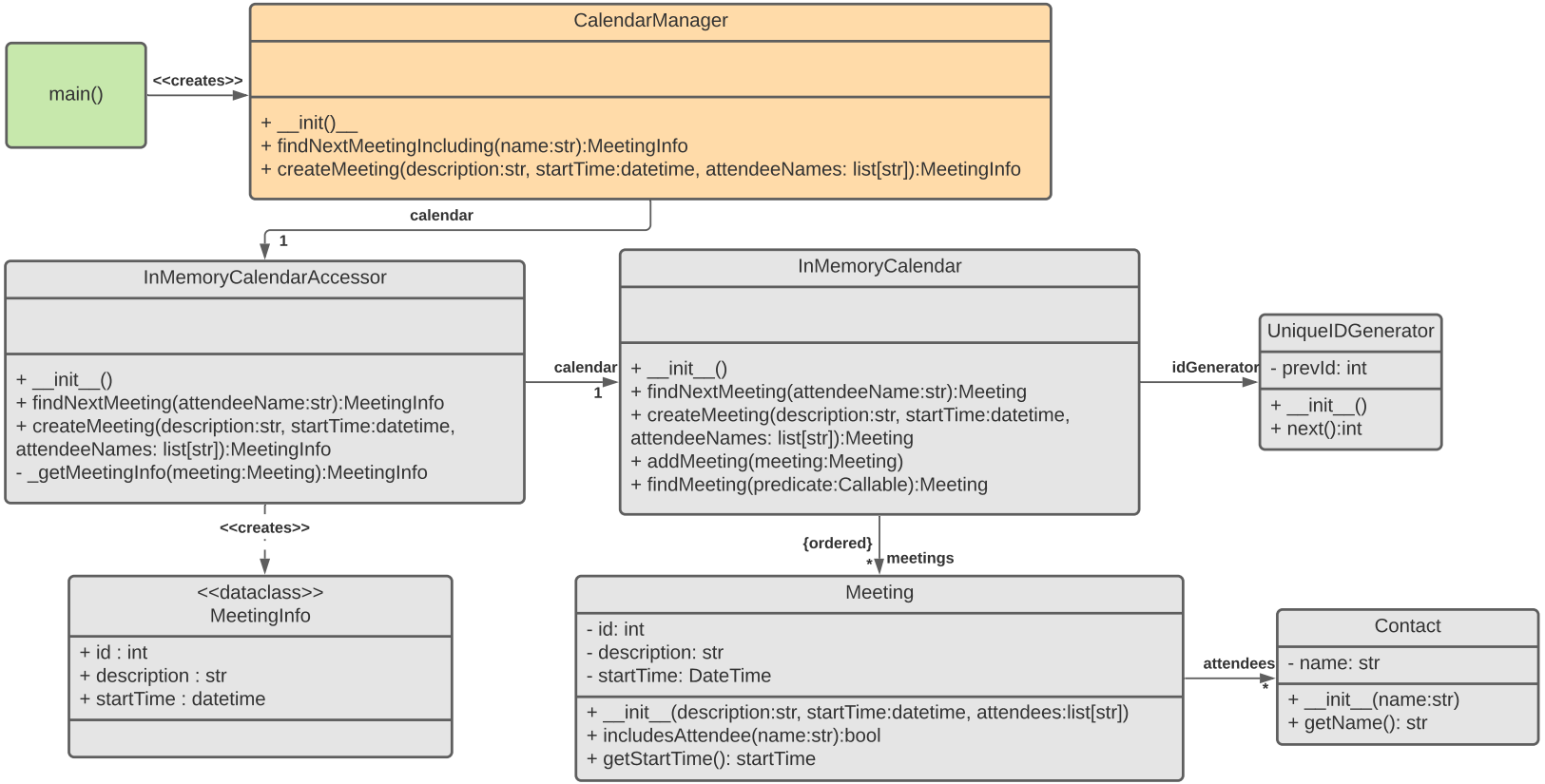
In VS Code terminal, use py instead of the full path to python.exe

Task 2. Test the Calendar Application

- We will revisit the implementation of the Calendar application from Practical 1 to test the classes implemented in Practical 1

- You can work with your code or download the provided code for Practical 2 from the course site.

Practical 1 Implementation Design



Test findNextMeeting() #1

- Rewrite the test performed in the main() method as pytest test.
  - Name the test function *test_findNextMeeting*
  - Put all test functions in a separate source file, not intermingled with the rest of the code.

- Run pytest to verify that the test passes.
- Then delete the main() method from the program and verify again that the test still works.

Test findNextMeeting() #2

- Write a test that verifies that findNextMeeting() returns None if there is no meeting that includes the given attendee's name.
- Use the same three meetings as in the previous test.
- Name the test *test_findNextMeeting_NoneFound*

Test Fixture #1

- Create a test fixture that creates the Manager object.
- Rewrite the previous two tests to use the fixture.
- Verify that the tests pass.

Test Fixture #2

- Create a test fixture that creates the same three meetings as in the previous tests.
- Rewrite the previous two tests to use the fixture (in addition to the Manager fixture created earlier).
- Verify that the tests pass.

Test Meeting Requires Attendees

- Write a test to verify that that creating a meeting without attendees raises a *ValueError* exception.

- Name the test *test_Meeting_Zero_Attendees*

- This test may fail initially.
- Debug the program and verify that the test passes.

Test findNextMeeting() #3

- Create another test fixture that creates the following meetings (in this order):

Description	Start Time	Attendees
Morning meeting	9am on 24 May	Alice, Bob
Afternoon meeting	3pm on 24 May	Bob, Charlie
Lunch meeting	12pm on 24 May	Alice, Bob, Charlie

Test findNextMeeting() #3 (cont.)

- Test that findNextMeeting() returns the *Lunch Meeting* as the next meeting that includes *Charlie*

- If this test fails, debug the program and verify that the bug has been resolved.

Task 3. Pull Out Dependencies

- We wish to prepare our code for the introduction of an alternative calendar backend backed by a database.
- Would the current implementation support this easily?
  - Why/Why not?

Issue

The CalendarManager creates its collaborator, the Accessor:

```
class CalendarManager:

    def __init__(self):
        self.calendar = InMemoryCalendarAccessor()
```

This is undesirable. How to resolve this?

Dependency Injection

- Apply the dependency injection principle so that accessor is supplied as an argument to the constructor of the CalendarManager.
- Refactor the tests to construct the adapter and pass it to the Manager.
- Verify that all tests pass.

Task 4. Test Stub

- Suppose we want to test the implementation of the CalendarManager class, but we have not yet implemented the Accessor class.

- Use a Mock object as a double for the Accessor object.
- Test that findNextMeeting() in the Accessor object is indeed called by findNextMeeting() in the Manager object.

You Should Know

- Write Unit Tests for your code
- Create stub/mock objects
- Apply Test-Driven Development practices

Activities this Week

- Complete Quiz 7
- Start working on Assignment 2

University of
South Australia