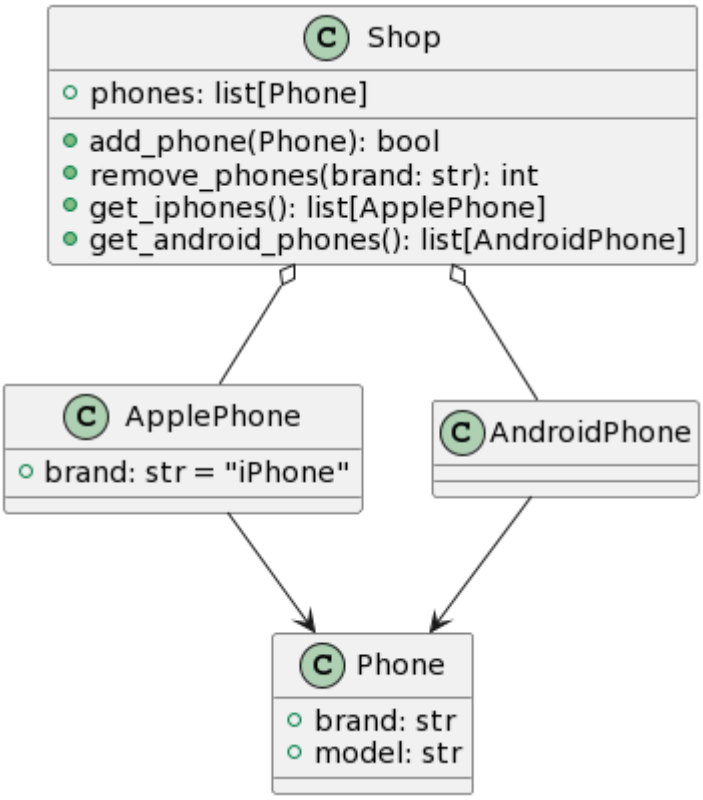
Midterm exam -- ACIT2515

Object oriented programming

In this assignment, you will implement classes to represent a shop selling phones. The shop has two kinds of phones: Apple phones and Android phones.

Class diagram



Part 1: Build the `Phone` class

All phones have a brand and a model (strings). You can use the `test_phone.py` unit test file.

3 marks (1 mark per test)

Part 2: Apple and Android Phones

- Apple phones always have the `iPhone` brand.
- Android phones behave like regular phones.

You can use the `test_all_phones.py` unit test file.

4 marks: 1 mark per test (3) + 1 mark for correct use of inheritance

Part 3: the shop

A shop has a `phones` attribute, containing a list of instances of phones (= the inventory).

- When a shop is created, it has no phones in its inventory.
- A phone can be added to the shop with the `add_phone` method.
    - This method raises a `TypeError` exception if the argument is not an instance of Phone or one of its subclasses.
- Phone(s) can be removed with the `remove_phone` method.
    - This method takes a **string** argument: the brand to remove from the inventory.
    - All phones with that brand are removed from the inventory.
    - The method returns the number of phones that were removed.
- `get_iphones` returns a list of all Apple phones in the shop
- `get_android_phones` returns a list of all Android phones in the shop
- There are two dunder methods that you need to implement. Read the tests!

You can use the `test_shop.py` unit test file.

> Make sure `get_iphones` and `get_android_phones` use the type to match the phones, and not their brands / attributes!

9 marks: 1 mark per test (7) + code quality and efficiency (2)

Possible bonus marks

- Clean code, docstrings, comments and PEP8
- Overall code quality and efficiency

Submission

Submit the following files to D2L:

- `phone.py`
- `apple_phone.py`
- `android_phone.py`
- `shop.py`

Do NOT use a ZIP file!