# Software Quality Assurance and Testing

# Workbook for Experiment 1

.

**Name: Ali hasan**

Answer the following questions.

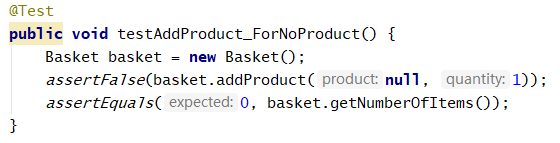
## Question 1

What problems did you find with your tests. List each problem and explain which method you found the problem in and the test that shows the problem.

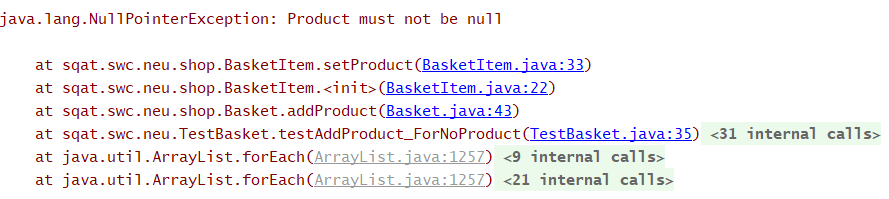
Problem 1

I find a problem in **addProduct()** method. When I pass a **null** product to **addProduct** method, I expect it return a **false** and the number of the items is **0**, but it throws a **NullPointerException.**

It should **cash** the exception otherwise the program will terminate.The test method is as follows:

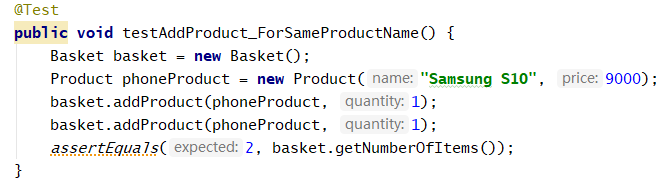


The result is as follows:

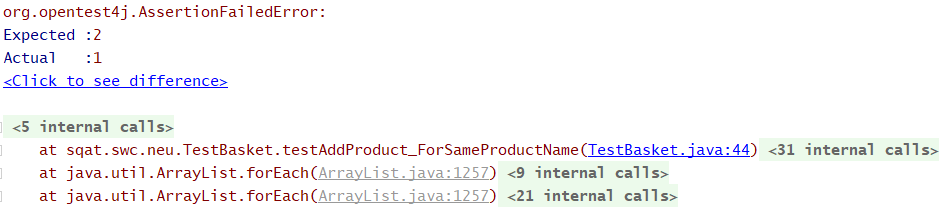


Problem 2

I find a problem in **addProduct()** method. When I add the product with the **same name** twice, I expect the number of the items is **2**, but it returns **1.** The test method is as follows:

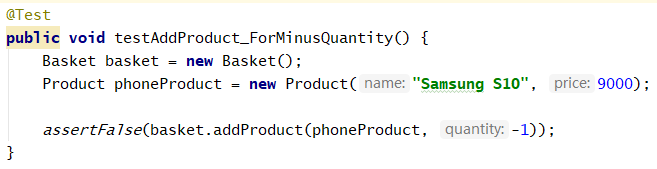


The result is as follows:

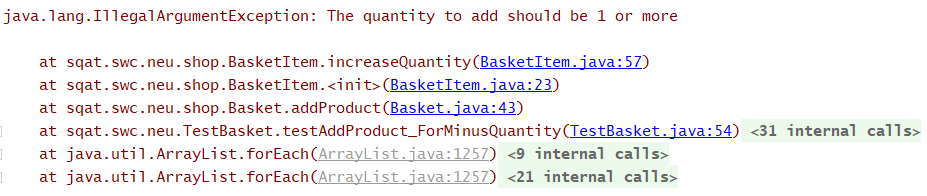


Problem 3

I find a problem in **addProduct()** method. When I add the product with the **minus quantity**, I expect it returns **false** but it throws an **IllegalArgumentException.** It should **cash** the exception otherwise the program will terminate.The test method is as follows:

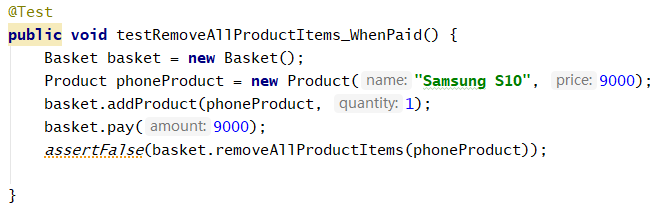


The result is as follows:

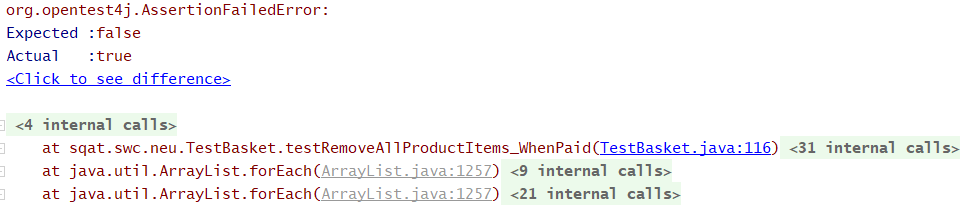


Problem 4

I find a problem in **removeAllProductItems()** method. I call this method when paid, I expect it returns false but it returns **true**. The test method is as follows:

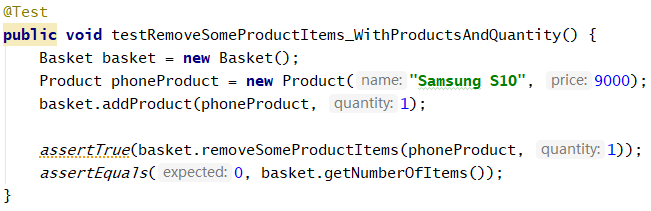


The result is as follows:

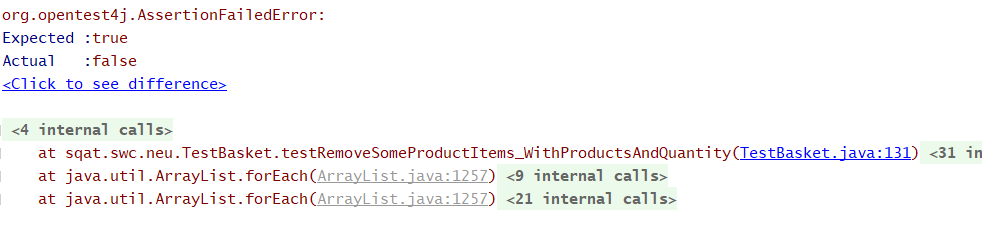


Problem 5

I find a problem in **removeSomeProductItems()** method. When I call this method with a correct product and quantity, I expect it returns **true** but it returns **false.** The test method is as follows:

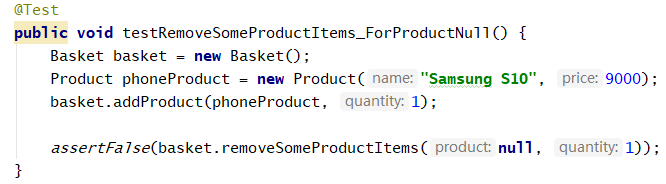


The result is as follows:

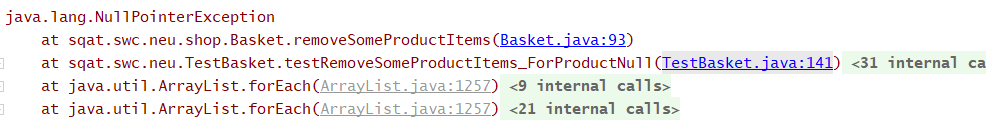


Problem 6

I find a problem in **removeSomeProductItems()** method. When I call the method with the **null product**, I expect it returns **false** but it throws a **NullPointerException.** It should **cash** the exception otherwise the program will terminate.The test method is as follows:

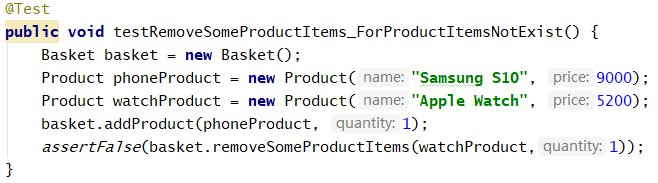


The result is as follows:

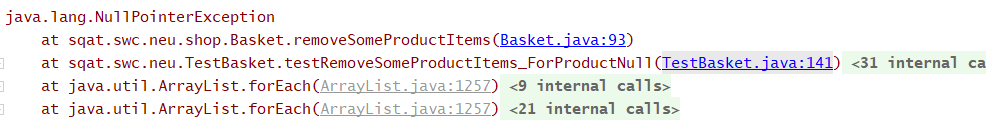


Problem 7

I find a problem in **removeSomeProductItems()** method. When I call the method with the **product that not exists**, I expect it returns **false** but it throws a **NullPointerException.** It should **cash** the exception otherwise the program will terminate.The test method is as follows:

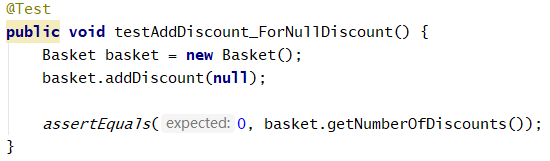


The result is as follows:

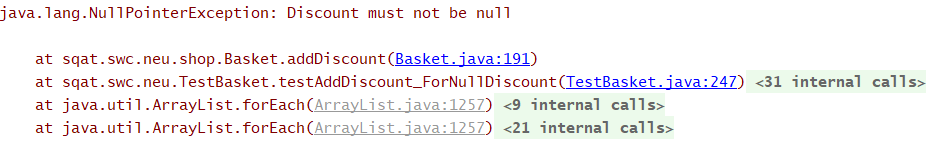


Problem 8

I find a problem in **addDiscount()** method. When I call the method with the **null product**, I expect it returns 0 but it throws a **NullPointerException.** It should **cash** the exception otherwise the program will terminate.The test method is as follows:

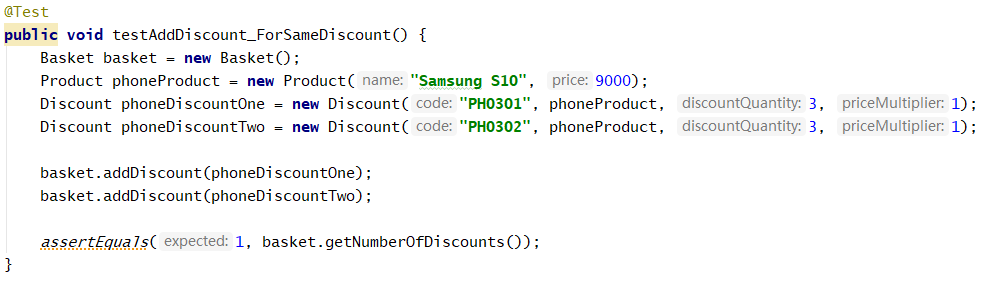


The result is as follows:



Problem 9

I find a problem in **addDiscount()** method. When I call the method with the **same product twice**, I expect it returns **1** but it returns **2.** The test method is as follows:

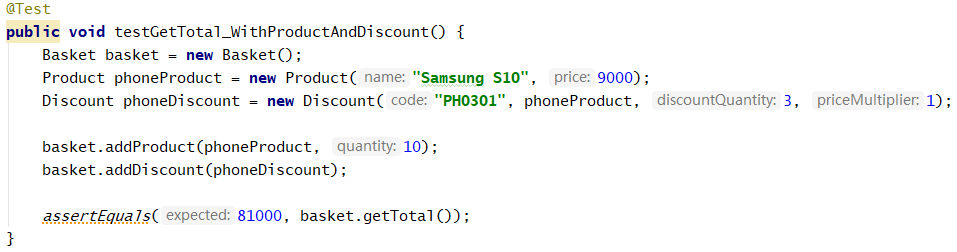


The result is as follows:

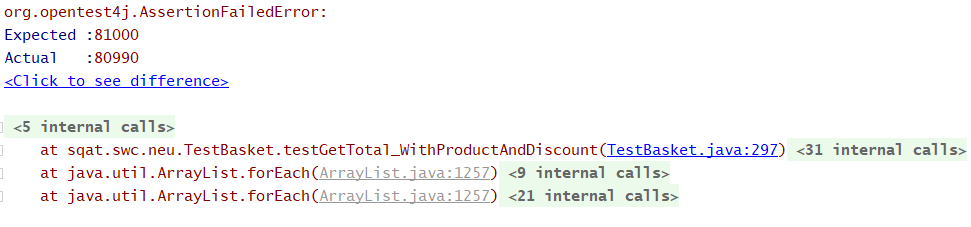


Problem 10

I find a problem in **getTotal()** method. When I call the method, I expect it returns **81000** but it returns **80990.** The test method is as follows:

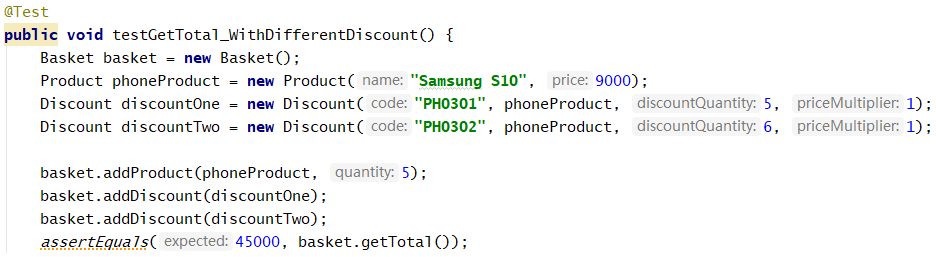


The result is as follows:

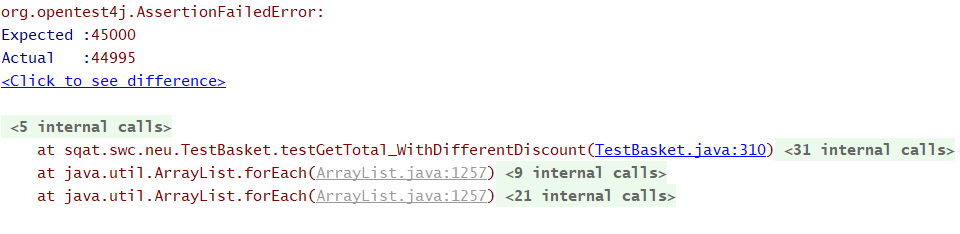


Problem 11

I find a problem in **getTotal()** method. When I call the method, I expect it returns **45000** but it returns **44995.** The test method is as follows:

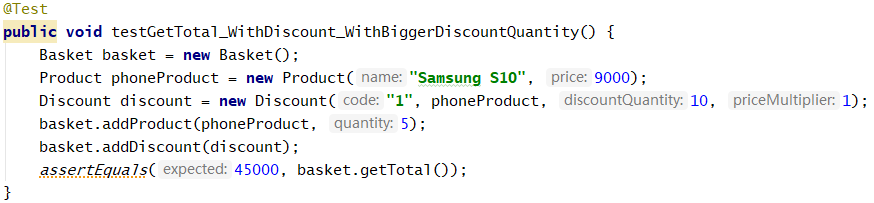


The result is as follows:

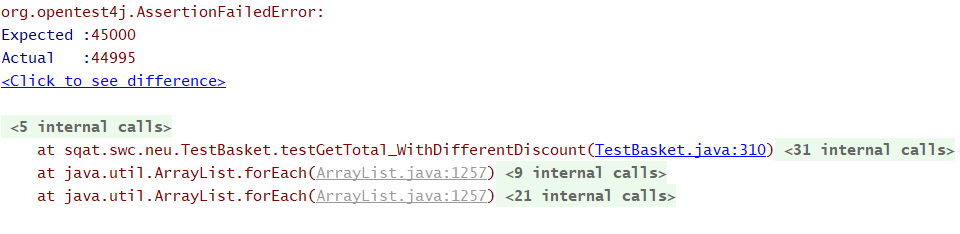


Problem 12

I find a problem in **getTotal()** method. When I add the method with a **bigger discount quantity,** I expect it returns **45000** but it returns **44995.** The test method is as follows:

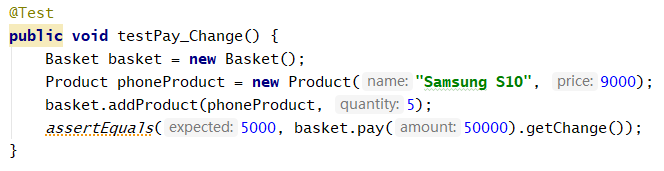


The result is as follows:

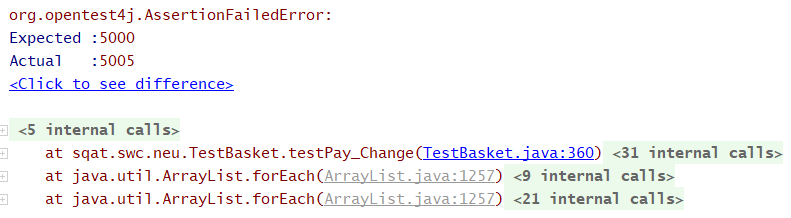


Problem 13

I find a problem in **getTotal()** method. When I add the method and want the change, I expect it returns **5000** but it returns **5005.** The test method is as follows:



The result is as follows:



## Question 2

Does the method containsProduct() in BasketItem contain any conditions that need to be tested? Explain your answer.

Answer

I think we need to test it.

**1.** Test the valid test case. Input the product that already exists in the BasketItem. The return result should be true.

**2.** Test invalid case. Input the product that is the product that not exists in the BasketItem. The return value should be false.

**3.** Test the null input. The return value should be null.

## Question 3

What is the Cyclomatic Complexity for the method getTotal() in the Basket class?

Answer



The Cyclomatic Complexity for the method getTotal() in the Basket class is **7**.

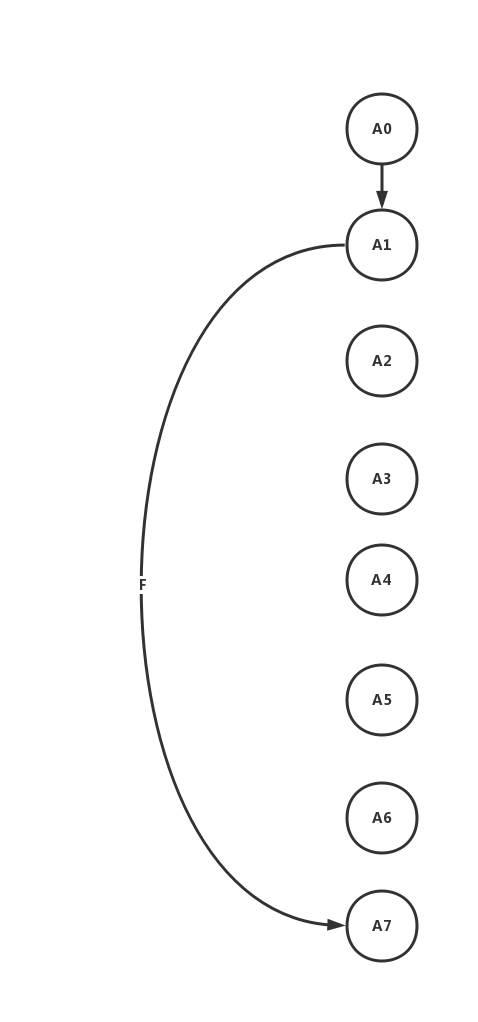
## Question 4

What are the basis paths that you identified for the method getTotal() in the Basket class?

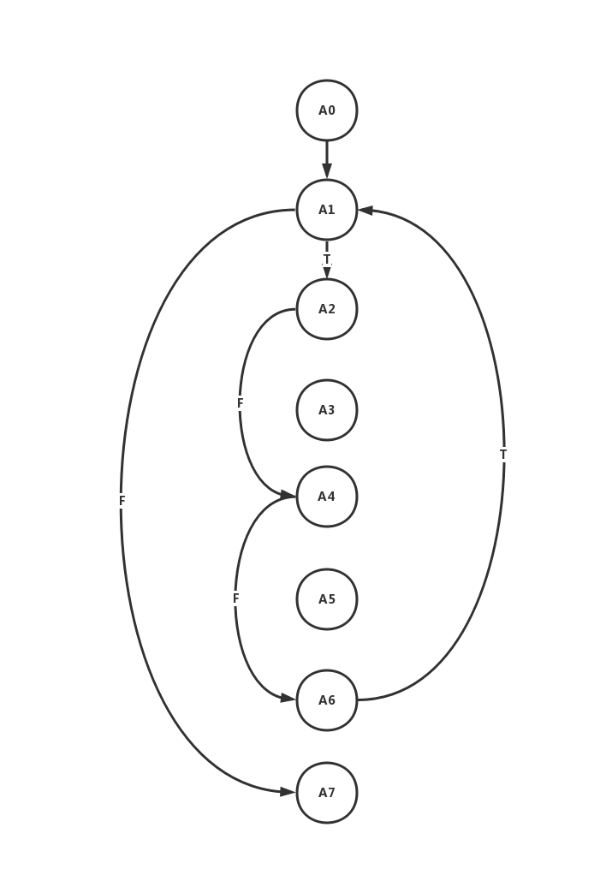
You can draw them with a drawing tool or you can draw them on paper and take a photograph to include in this document.

Answer

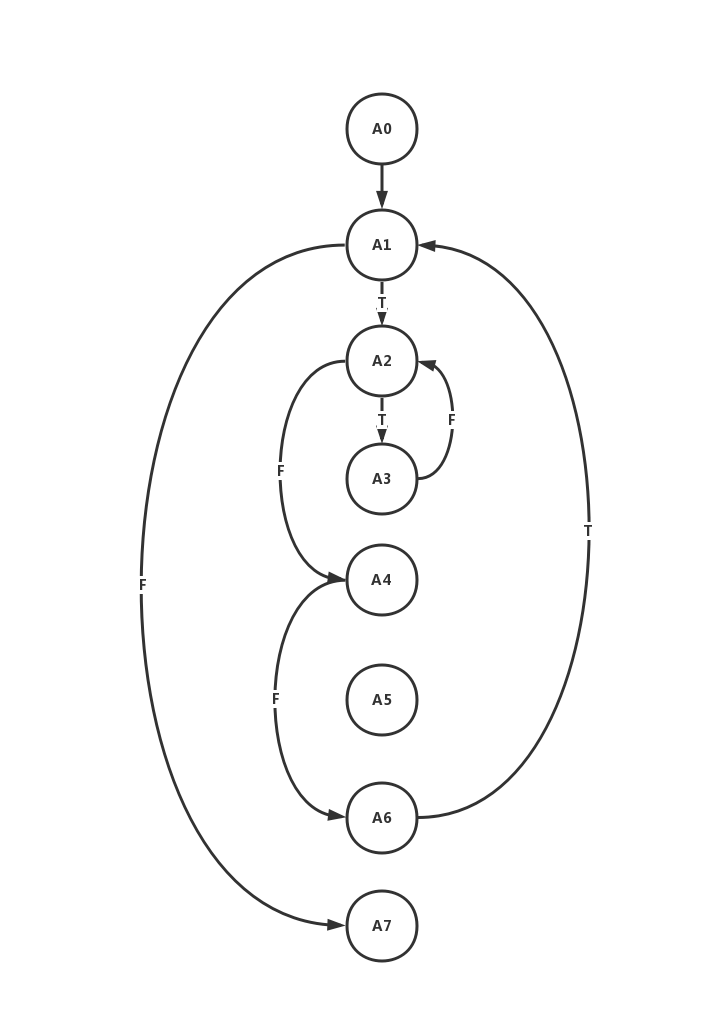
Basis path 1



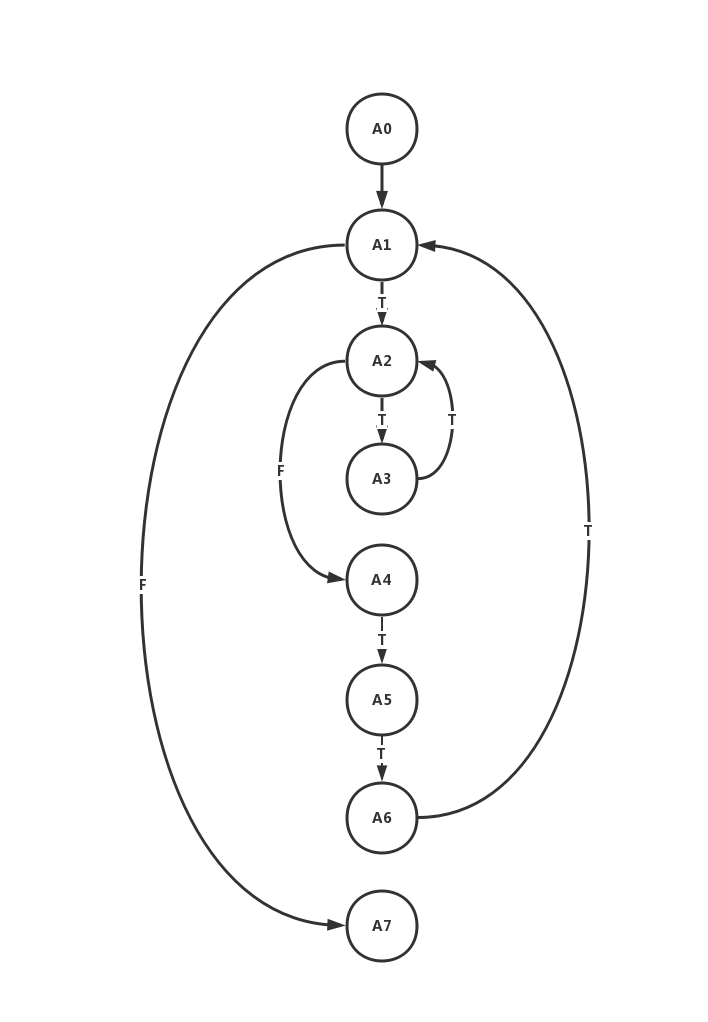
Basis path 2



Basis path 3



Basis path 4



Basis path 5

