TDD测试驱动开发实验

* 实验名称： TDD测试驱动开发实验
* 实验目的
  1. 理解TDD测试驱动开发的思想
  2. 掌握TDD测试驱动开发的基本过程。
* 实验内容和要求
  1. 阅读：Kent Beck 《测试驱动开发》
  2. 基于TDD开发一个判断字符串是IP4地址的功能
  3. 使用任何OO语言
  4. 使用xUnit测试框架

基于TDD开发一个判断字符串是IP4地址的功能的思路

编程语言：python 测试框架：unittest

1. 编写测试用例
2. 编写实现代码和测试
3. 运行测试，找出问题
4. 修改代码，使测试通过
5. 运行测试，检查问题
6. 重复以上步骤直到实现功能
7. 重构代码，使其优化

—————————————————————————

一、编写测试用例

考虑一些IPv4地址和非IPv4地址的情况，例如：

* 合法的IPv4地址：**"192.168.1.1"，"0.0.0.0"，"255.255.255.255"**
* 非法的IPv4地址：**"256.256.256.256"** **，"192.168.1.1.1"，"192.168.1""abc.def.ghi.jkl"**

二、编写实现代码和测试

**·** 定义函数**is\_ipv4\_address(address)** 来判断测试用例是否为合法的IPv4地址

* 定义一个名为 **TestIsIPv4Address**的测试类，其中包含了两个测试方法：**test\_valid\_ipv4\_addresses** 和**test\_invalid\_ipv4\_addresses**。

1. **test\_valid\_ipv4\_addresses** 方法用于测试合法的 IPv4 地址是否能被正确识别。通过调用 **is\_ipv4\_address** 函数并使用 **self.assertTrue** 断言来验证函数返回值为 **True**。
2. **test\_invalid\_ipv4\_addresses** 方法用于测试非法的 IPv4 地址是否能被正确识别。通过调用 **is\_ipv4\_address** 函数并使用 **self.assertFalse** 断言来验证函数返回值为 **False**。

通过运行这些测试，可以验证 **is\_ipv4\_address** 函数在实现时是否符合预期的行为。

import unittest  
import re  
  
def is\_ipv4\_address(address):   
 return True  
  
class TestIsIPv4Address(unittest.TestCase):  
  
 def test\_valid\_ipv4\_addresses(self):  
 self.assertTrue(is\_ipv4\_address("192.168.1.1"))  
 self.assertTrue(is\_ipv4\_address("0.0.0.0"))  
 self.assertTrue(is\_ipv4\_address("255.255.255.255"))  
  
 def test\_invalid\_ipv4\_addresses(self):  
 self.assertFalse(is\_ipv4\_address("256.256.256.256"))  
 self.assertFalse(is\_ipv4\_address("192.168.1.1.1"))  
 self.assertFalse(is\_ipv4\_address("192.168.1"))  
 self.assertFalse(is\_ipv4\_address("abc.def.ghi.jkl"))  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

三、运行测试，找出问题

报错如下，说明**test\_invalid\_ipv4\_addresses**发现了问题，即函数**is\_ipv4\_address(address)**没有正确识别不合法的IPv4地址



这个错误的原因在于**is\_ipv4\_address(address)**还没有正式编写内容，只说明了作用。接下来将对其函数进行重构以实现预期功能。

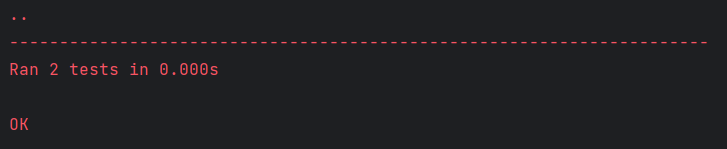
四、修改代码，使测试通过

修改函数**is\_ipv4\_address(address)**内容**，**使用正则表达式验证IPv4地址：

def is\_ipv4\_address(address):  
 pattern = r'^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$'  
 return bool(re.match(pattern, address))

五、运行测试，检查问题

重构代码后的测试结果如下：



说明，运行测试用例，验证代码实现符合预期。

六、重构代码，使其优化

这个版本中，将 IPv4 地址的正则表达式命名为 **ipv4\_pattern**，使其更易于理解。此外，测试用例中使用了更具描述性的断言方法，例如 **assertTrue** 和 **assertFalse**，使测试用例的意图更清晰。此外，只保留一个必要的测试：

import unittest  
import re  
  
def is\_ipv4\_address(address):  
 ipv4\_pattern = r'^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$'  
 return bool(re.match(ipv4\_pattern, address))  
  
class TestIsIPv4Address(unittest.TestCase):  
  
 def test\_valid\_ipv4\_addresses(self):  
 self.assertTrue(is\_ipv4\_address("192.168.1.1"))  
  
 def test\_invalid\_ipv4\_addresses(self):  
 self.assertFalse(is\_ipv4\_address("192.168.1.1.1"))  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

参考文献：[测试驱动开发 ( kent Beck ）](https://github.com/TonnnnnnyLiu/Books/blob/master/%E6%B5%8B%E8%AF%95%E9%A9%B1%E5%8A%A8%E5%BC%80%E5%8F%91.pdf)