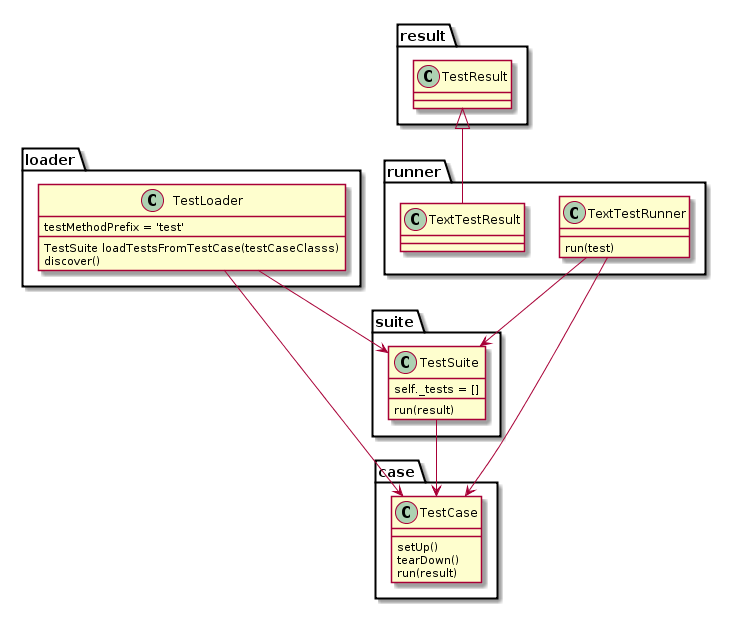
## ****unittest核心工作原理****

unittest中最核心的四个概念是：****test case, test suite, test runner, test fixture****。



上面四个定义的含义：

* 一个TestCase的实例就是一个测试用例。什么是测试用例呢？就是一个完整的测试流程，包括测试前准备环境的搭建(setUp)，执行测试代码(run)，以及测试后环境的还原(tearDown)。元测试(unit test)的本质也就在这里，一个测试用例是一个完整的测试单元，通过运行这个测试单元，可以对某一个问题进行验证。
* 而多个测试用例集合在一起，就是TestSuite，而且TestSuite也可以嵌套TestSuite。
* TestLoader是用来加载TestCase到TestSuite中的，其中有几个loadTestsFrom\_\_()方法，就是从各个地方寻找TestCase，创建它们的实例，然后add到TestSuite中，再返回一个TestSuite实例。
* TextTestRunner是来执行测试用例的，其中的run(test)会执行TestSuite/TestCase中的run(result)方法。   
  测试的结果会保存到TextTestResult实例中，包括运行了多少测试用例，成功了多少，失败了多少等信息。
* 而对一个测试用例环境的搭建和销毁，是一个fixture。

以上整个流程就清楚了：

一个class继承了unittest.TestCase,便是一个测试用例，如果其中有多个以test开头的方法，每有一个这样的方法，在load的时候就会生成一个TestCase实例，如：一个class中有两个test\_xxx方法，最后load到suit中时也有四个测试用例。

写好TestCase，然后由TestLoader加载到TestSuit，然后由TextTestRunner来运行TestSuit，运行的结果保存到TextTestResult中，可以通过unittest.main()执行，这时main会调用TextTestRunner中的run()执行，或者直接通过TextTestRunner执行用例。Runner执行测试用例时，将结果直接输出到控制台，我们可以设置，将结果输出到文件中。

说明：

1、Test\_func.py直接调用mian()函数执行

2、Test\_suit.py（组织TestSuit）为加强版本，控制了用例的执行顺序，这时我们添加到TestSuit中的case会按添加顺序执行。

3、test\_suit1.py将结果输出到文件中。