**黑白盒測試**

**黑盒測試**

軟體的黑盒測試意思指的是在軟體的接口處進行測試，這種測試方法實際是把測試對象看成是一個拆不開的物件，測試人員不清楚物件裏程序內部邏輯結構和內部特性，只能依據程序的說明書，檢查程序的功能是否符合說明書所述的輸入條件和輸出結果。因此，黑盒測試我們又可以稱為**[功能測試或數據驅動測試](https://zh.wikipedia.org/wiki/%E9%BB%91%E7%9B%92%E6%B5%8B%E8%AF%95)**。

**例子說明：**

家裏有一個全自動製麵包機，只需準備好材料和放入機器內，便可以等待製作，但等待期間不知道內部製作過程，只知道正確的結果應該是完成可以食用的麵包。

**白盒測試**

軟體的白盒測試意思指的是對軟體的過程細節做詳細細緻的檢查，這種測試方法實際是把測試對象看成是一個拆散了的物件，測試人員把整個物件完全拆開，可以看到裏面程序內部邏輯結構和所有信息，針對內部結構進行設計或選擇測試用例，對程序所有邏輯進行測試，通過在各種不同的測試用例檢查程序狀態，確定實際狀態是否與預期的狀態一致。因此，白盒測試我們又可以稱為[**結構測試或邏輯驅動測試**](https://zh.wikipedia.org/wiki/%E7%99%BD%E7%9B%92%E6%B5%8B%E8%AF%95)。

**例子說明：**

家裏有一個製麵包機，在製作的過程中，每一個步驟都要人手去監控與操作，製作員需要注意每一個步驟的細節，了解每一個步驟在機器內部是如何完成的，檢查步驟的正確性，到最後製作出一個完成可以食用的麵包。

**灰盒測試**

軟體的灰盒測試是介於白盒與黑盒測試之間的一種方式，灰盒測試除了關注輸出對於輸入的正確性外，同時也關注內部表現，但關注的程度不像白盒測試那樣詳細和完整，只是通過一些表征性的現象和事件來判斷內部的運行狀態，比如說測試的結果有時候是正確的，但內部處理邏輯其實已經出現錯誤。這種情況發生的情況非常多，但如果我們每次都使用白盒測試，其實很浪費時間和效率很低，所以很多時候我們都會選擇灰盒測試。

|  |  |  |
| --- | --- | --- |
|  | 優點 | 缺點 |
| 白盒測試 | 1.能仔細考慮軟體的實現 2.可檢測代碼中的每條分支和路徑 3.揭示隱藏在代碼中的錯誤 4.對代碼的測試比較徹底 | 1.測試時間長 2.無法檢測代碼中遺漏的路徑和敏感性錯誤 3.昂貴 4.不驗証規格的正確性 |
| 黑盒測試 | 1.對於子系統甚至系統，效率比較高 2.測試人員不需要了解實現細節，包括特定的程式語言 3.測試人員和編程人員彼此獨立 4.以用戶的角度測試，比較容易理解和接受 5.有助於暴露規格的不一致或有歧義的問題 6.測試用例可以在規格完成後馬上進行 | 1.只有一小部分輸入被測試到，要測試每個可能的輸入幾乎不可能 2.沒有清晰、簡明的規格，測試用例很難設計 3.如果測試人員不被告知開發人員已經執行過的用例，在測試數據上會存在不必要的重複 4.有很多程序路徑沒有被測試到 5.不能直接針對特定程序段測試，而這些程序段可能很複雜，有可能隱藏更多的問題 6.大部分和研究相關的測試都是直接針對白盒測試的 |

**三種測試方式之比較**

**灰盒測試與黑盒測試的分別**

如果測試對象軟體包含多個模塊，當使用黑盒測試時，只需要關注整個軟體系統的執行結果是否乎合預期，無需關心軟體內部各個模塊之間如何溝通。而當使用灰盒測試時，就需要關注各模塊之間的交互，各模塊的輸入和輸出是否達到預期，這就是灰盒與黑盒測試的分別。

**灰盒測試與白盒測試的分別**

上面講述了灰盒測試需要關注模塊的輸入條件和輸出結果，但還是不需要關心模塊內部的實現細節。對於軟體系統的內部模塊來說，灰盒測試對內部模塊當成一個黑盒來看待。而白盒測試不同的是，還需要深入了解模塊內部的細節，這就是灰盒與白盒測試的分別。

**黑盒測試與白盒測試的分別**

如何選擇測試方法呢?小篇建議要先想想測試需要達到甚麼目的，而再需要項目成員根據實際所需情況去制定。

**接口測試Postman**

1. 接口由來：連接前後端
2. 接口測試意義
3. 縮短項目時間、節約時間
4. 提高工作效率
5. 更利於系統健壯姓
6. 接口種類
7. 外部接口：如銀行
8. 內部接口：模塊與模塊間，web端和app端的對接
9. 接口有那些類型
10. get(主要)：直接通過地址獲得
11. post(主要)：提交資料，較常用於查詢資料使用，需要將客戶端資訊傳給伺服器進行數據修改、增刪
12. put
13. delete
14. 常用接口工具
15. postman
16. jmeter(性能測試同時可以做接口測試)
17. soapui
18. fiddier
19. 接口測試流程(和黑盒測試相同)：黑盒測試是指功能測試，不管內部程序，只管web的呈現是否正常；
20. 需求討論
21. 需求評審
22. 用例設計
23. 功能
24. 功能是否正常
25. 功能是否按照接口文檔實現
26. 邏輯業務
27. 是否依賴業務：例如：下單需考慮客戶是否登入了
28. 異常
29. 參數異常：

關鍵字參數：參數為空、參數多或者少、錯誤參數

1. 數據異常

關鍵字數據：長度不一、數據為空、數據錯誤

1. 安全
2. Header
3. cookie
4. 唯一識別碼
5. 數據準備
6. 執行
7. 用力設計考慮那些方面
8. 功能
9. 邏輯業務：關聯功能，例如下單功能，具有依賴業務，需登入才可以下單，此時需考慮：headers
10. 異常
11. 參數異常：
12. 關鍵字異常：如phone輸入成pyth
13. 參數為空
14. 參數多或者少
15. 數據異常：
16. 關鍵字

數值輸入到關鍵字，如null

1. 長度不一致

長度與規範的長度不一樣

1. 數據為空

數據輸入為空值

1. 數據錯誤

輸入的值內容不正確

1. Postman工具使用
2. 各類型接口測試
3. get

透過地址後?輸入數據與關鍵字：如https://org/get?phone=xxx&key=xxx

1. post

透過body設置請求數據，需要將body傳送給伺服器端

1. History、collections的使用
2. History：

會記錄每次的請求，可以匯出檔案，可以在collection看到

1. Collections：

可以讀取值前儲存之前的請求紀錄，可以選擇批量運行

1. 環境變量設置：同一測試數據在不同環境中進行測試
2. 點擊右上角齒輪，即可添加環境
3. 變量需要用”{{}}”，如：{{url}}
4. 在collections的批量執行中可以選擇設定好的環境
5. 斷言：判斷測試結果的對錯
6. 在Tests中設定，
7. 如：選擇”Restponse body:Contains str”：判斷body內容中是否有包含目標值，ex：測試成功、success等
8. 在Collections批量運行時即會顯示：pass 或 fail
9. Postman持久化運行Collecion Runner

**UnitTest**

1. UnitTest介紹

在python中，有一個已存好的框架值接調用，名子叫PyUnit，類似於Junit，同屬於Unit家族中的一員。PyUnit就是UnitTest。Unit家族主要用於單元測試。

現在Java應用TestNG來想辦法替代Junit，Pyton中，應用Pytest，想要替代PyUnit。

1. 核心介紹
2. Test\_Case

測試用例，基於unittes.testcase，所有的用例都可以基於Test\_Case來實現。

1. Test\_FixTurse

setUp，tearDown，test\_case集合起來稱之

1. Test\_Suites

測試套件，用來獨立包裝溪的測試套件，重新複製測試用例

1. Test\_Runner

運行器

1. UnitTest基本規則

在編寫代碼時，一定要遵守的規則

1. setUp(前置條件)和tearDown(後置條件)兩個函數，命名必須依照這個格式和名稱
2. Test\_Case在編寫時，函數命名必須以test開頭
3. Unittest在運行時，需要通過調用unittest.main()進行運行

**自動化測試之郵件自動化**

1. SMTP

通過SMTP來連繫其他郵件

在python中，有一個資料庫儲存：smtplib專門用來處理郵件的函式庫，mail庫，用來發送郵件的函式庫。

1. 自動化郵件的時測案例
2. 要處理的主題
3. 郵件人
4. 收件人
5. 正文
6. 附件
7. 郵件正文
8. 文本訊息
9. HTML訊息
10. 圖片內容
11. 附件內容
12. Txt.
13. Jpg
14. Pdf
15. Xls
16. 發送郵件流程
17. 讀取附件
18. 添加附件
19. 與內容一起發送
20. 自動化框架設計思路
21. 正常自動化

編寫自動化相關內容，進行維護更新，進行執行自動化測試

1. 原碼：
2. 關鍵字驅動，
3. 進行文件讀取，
4. 實現數據驅動，
5. 通過可是化界面，上傳文件，
6. 發送運行指令，讓服務端運行以上傳聞件，
7. 進行腳本自動化腳本的執行，運行過程中，紀錄用力的成功與失敗狀態，產生測試結果，
8. 生成測試報告，基於郵件的行式，將報告發送置相關團隊成員
9. 關聯代碼庫
10. 發布新的版本時自動化部屬，運行常規測試
11. 產出遊件報告，發送郵件

**CSS元素定位**

1. 自動化測試元素定位的放式
2. ID
3. Name
4. Classname
5. Tagename
6. Linktext
7. Partial-linktext
8. Xpath
9. css
10. css定位元素方式
11. 絕對路徑
12. 絕對路徑：從根標籤(html)一級一級找到目標標籤
13. 上下及之間使用>或者空格連接

drive.find\_element\_by\_css\_selector("html>body > div.L3eUgb > div.o3j99.ikrT4e.om7nvf > form > div:nth-child(1) > div.A8SBwf > div.RNNXgb > div.SDkEP > div.a4bIc > input").send\_keys("小貓咪")

1. Id定位
2. Id選擇器符號：#

drive.find\_element\_by\_css\_selector("#kw").send\_keys("小貓咪")

1. 透過class定位

class選擇器符號：.

drive.find\_element\_by\_css\_selector(".s\_ipt").send\_keys("小貓咪")

1. 其他屬性
2. [屬性名=’屬性名稱’]

drive.find\_element\_by\_css\_selector("[autocomplete='off']").send\_keys("小貓咪")

1. 通過多屬性定位

[屬性名=’屬性名稱’] [屬性名=’屬性名稱’]

drive.find\_element\_by\_css\_selector("[autocomplete='off'][name='wd']").send\_keys("小貓咪")

1. 通過部分屬性值進行定位

匹配符號：

1. \* ：包含某個字幅
2. ^：首字母包含，以甚麼開頭
3. $：以甚麼結尾

drive.find\_element\_by\_css\_selector("[autocomplete\*='f']").send\_keys("小貓咪")

drive.find\_element\_by\_css\_selector("[autocomplete^='of']").send\_keys("小貓咪")

drive.find\_element\_by\_css\_selector("[autocomplete$='ff']").send\_keys("小貓咪")

1. 通過層級定位：一般情況下會跟id/calss/屬性一起組合進行定位

上下級符號：>或者空格

通過層級定位

drive.find\_element\_by\_css\_selector("form>span>input").send\_keys("小貓咪")

drive.find\_element\_by\_css\_selector("form#form>span>input").send\_keys("小貓咪")#層級和id組合定位

drive.find\_element\_by\_css\_selector("form.fm>span>input").send\_keys("小貓咪")#層級和class組合定位

1. 通過兄弟節點定位：
2. 兄弟節點：同一父級標籤，下面有多個相同標籤元素，那麼這些元素稱為兄弟節點
3. 定位第一個節點：first-child
4. 定位第2/3/4…n個節點：nth-child(n) n元素序號
5. 定位最後一個節點：last-child

#通過兄弟節點

drive.find\_element\_by\_css\_selector("div#s-top-left a:first-child").click()

drive.find\_element\_by\_css\_selector("div#s-top-left>a:nth-child(2)").click()

drive.find\_element\_by\_css\_selector("div#s-top-left>a:last-child").click()

**HtmlTestRunner自動生成測試報告**

1. HtmlTestRunner

是Python 標主準庫的unittest模組的一個擴展。他生成易於使用的HTML測試報告。

**RobotFrameWork**

1. RobotFrameWork介紹

Python結合selenium實現的一個開源的自動化測試框架，叫做RobotFrameWork。在WebUI自動化中，是結合selenium來實現，在App端視結合Appium來實現，在街口是結合request來實現。

目前主流的自動化設計框架：RF、AirTest

1. RF環境的搭建要如何實現
2. Python安裝
3. Python中RobotFrameWork的環境
4. pip install robotframework
5. pip install robotframework-ride
6. 基於要實現的操作，導入對應的模組，例如webUI，導入selenium

pip install robotframework-seleniumlibrary

1. 導入瀏覽器驅動，將WebDriver文件放置在python的安裝跟路徑下
2. RF的實際操作
3. 新建工程，type(directory)，format(txt)
4. 新增套件(suite)：type(file)，format(txt)
5. 新建測試用例，Newtestcase
6. 導入library，例如seleniumlibrary，導入有分大小寫，錯誤是紅色
7. 在RF中，絕對講究關鍵字驅動和數據驅動，通過表格編程，調用以封裝的關鍵字來實現內容
8. 快捷鍵
9. F5，打開查找關鍵字頁面
10. Ctrl+alt+space,補全
11. 問題
12. 做為測試，要站在用戶的角度來看待問題，所以要盡可能模擬用戶的場景來操作，比如使用360瀏覽器。
13. 在瀏覽器的世界中，總計五大內核：IE、chrome、FireFox、Opera、Safari
14. QQ、360、獵豹……，基本上都是調用IE和chrome瀏覽器內核

例如360全兼容模式是調用IE內核，急速模式是調用chrome內核

1. 自動化說明
   1. 自動化表示：自動化功能測試和自動化性能測試，一般自動化指的是自動化功能測試。

2

3