

# 软件测试黑盒实验

赵钰 1712990

## Part1：理论题目

1. 某网站在进行新用户的注册时，有如下要求：

- (1) 用户名为 6~18 位数字与字母的组合。
- (2) 密码为 8~18 位数字与字母的组合。

请结合等价类划分法设计出相应的测试用例。说明：用户名和密码需要既有数字也有字母

输入域等价类划分

有效等价类	编号	无效等价类	编号
用户名位数为 6~18	1	用户名小于 6 位	11
		用户名大于 18 位	12
用户名包含数字和字母	2	用户名仅包含数字	13
		用户名仅包含字母	14
		用户名包含数字字母之外的其他字符	15
密码位数为 8~18 位	3	密码位数小于 8 位	16
		密码位数大于 18 位	17
密码包含数字和字母	4	密码仅包含数字	18
		密码仅包含字母	19
		密码包含数字字母之外的其他字符	20

### 输出域等价类划分

有效等价类	编号
用户名和密码格式正确	5

### 覆盖有效等价类的测试用例

输入数据	输出结果	覆盖的等价类
(zy1999, abcd12345)	用户名和密码格式正确	1, 2, 3, 4, 5

### 覆盖无效等价类的测试用例

无效等价类	输入数据	输出结果	覆盖的等价类
用户名小于 6 位	(zy99, abcd12345)	输入错误	11
用户名大于 18 位	(zyzyzy1234567890zyzyzy, abcd12345)	输入错误	12
用户名仅包含数字	(19991234, abcd12345)	输入错误	13
用户名仅包含字母	(nkcszyzy, abcd12345)	输入错误	14
用户名包含数字字母之外的其他字符	(nkcs_zy, abcd12345)	输入错误	15
密码位数小于 8 位	(zy1999, abc123)	输入错误	16
密码位数大于 18 位	(zy1999, abcdefghijklmn123456789)	输入错误	17
密码仅包含数字	(zy1999, 199999995)	输入错误	18
密码仅包含字母	(zy1999, nkcszyzyzy)	输入错误	19
密码包含数字字母之外的其他字符	(zy1999, nkcs_zyzy)	输入错误	20

2. 某工厂的工人管理系统中，有一项是输入工人某月份加工的商品数目，系统返回该工人该月份的等级评定，从而较好地实现按劳分配。具体规则如下：

- (1) 加工的商品数最多不超过 1000 件
- (2) 500 件以下为等级 D
- (3) 500~700 件为等级 C
- (4) 700~900 件为等级 B
- (5) 900 件以上为等级 A

请结合边界值分析法设计出相应的测试用例。

说明：件数一定是为整数，即[0,500)为 D, [500,700)为 C, [700,900)为 B, [ 900,1000 ] 为 A

考虑输入数据的取值范围和类型划分等价类

编号	输入条件	所属类别	编号	输入条件	所属类别
1	0~499 之间整数	有效等价类	7	小数	无效等价类
2	500~699 之间整数	有效等价类	8	非数值(字母)	无效等价类
3	700~899 之间整数	有效等价类	9	非数值(特殊字符)	无效等价类
4	900~1000 之间整数	有效等价类	10	非数值(空格)	无效等价类
5	<0 整数	无效等价类	11	非数值(空白)	无效等价类
6	>1000 整数	无效等价类			

边界值测试用例

测试用例	输入数据	预期输出
Test 1	0	D
Test 2	1	D
Test 3	249	D
Test 4	498	D
Test 5	499	D
Test 6	500	C
Test 7	501	C
Test 8	600	C

Test 9	699	C
Test 10	700	B
Test 11	701	B
Test 12	800	B
Test 13	899	B
Test 14	900	A
Test 15	950	A
Test 16	999	A
Test 17	1000	A
Test 18	-1	提示“请输入 0~1000 间的整数”
Test 19	1001	提示“请输入 0~1000 间的整数”
Test 20	2.5	提示“请输入 0~1000 间的整数”
Test 21	abc	提示“请输入 0~1000 间的整数”
Test 22	#¥%	提示“请输入 0~1000 间的整数”
Test 23		提示“请输入 0~1000 间的整数”
Test 24		提示“请输入 0~1000 间的整数”

3. 一款游戏机的操作规则如下:

(1)玩一次游戏, 用户需要支付 10 元, 使用支付宝扫码支付, 有两种付款结果, 成功支付则进入步骤(2);未成功支付则提示付款失败

(2)屏幕上出现 20 个按键, 选择 9 号和 15 号, 会有礼品弹出, 且弹出的礼品不同;选择其他按键, 则游戏失败, 一无所获

请列出原因和结果, 画出因果图

说明:20 个按键是属于[1,20]区间的整数

原因:

1——支付宝支付 10 元成功

2——选择 9 号按键

3——选择 15 号按键

4——选择其他按键

结果:

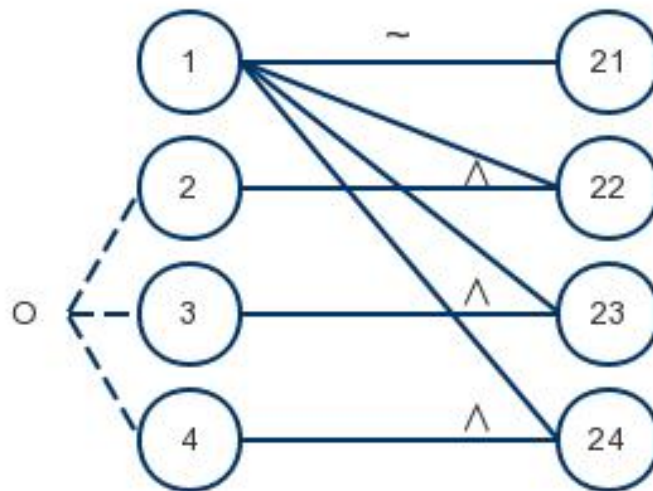
21——提示付款失败

22——弹出 9 号按键对应礼品

23——弹出 15 号按键对应礼品

24——游戏失败

因果图:



4、某厂对一部分职工重新分配工作，分配原则是:

(1)年龄不满 20 岁，文化程度是小学者脱产学习，文化程度是中学者当电工;

(2)年龄满 20 岁但不足 50 岁，文化程度是小学或中学者，男性当钳工，女性 当车工;文化程度是大专者，当技术员。

(3)年龄满 50 岁及 50 岁以上，文化程度是小学或中学者当材料员，文化程度 是大专者当技术员。

请构造决策表并化简。

等价类集合如下:

Y1: 不满 20 岁

Y2: 满 20 岁但不足 50 岁

Y3: 满 50 岁及 50 岁以上

B1: 小学

B2: 中学

B3: 大专

S1: 男

S2: 女

决策表

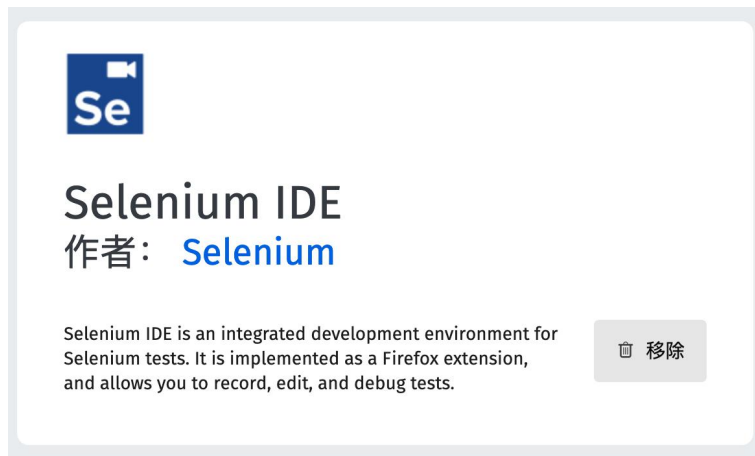
		规则							
选项		1	2	3	4	5	6	7	8
条件	C1: 年龄	Y1	Y1	Y2	Y2	Y2	Y3	Y3	Y3
	C2: 文化程度	B1	B2	B1	B1	B3	B1	B2	B3
	C3: 性别	—	—	S1	S2	—	—	—	—
动作	A1: 脱产学习	✓							
	A2: 电工		✓						
	A3: 钳工			✓					
	A4: 车工				✓				
	A5: 技术员					✓			✓
	A6: 材料员						✓	✓	

化简后决策表

		规则					
选项		1	2	3	4	5	6
条件	C1: 年龄	Y1	Y1	Y2	Y2	Y2~Y <sub>3</sub>	Y3
	C2: 文化程度	B1	B2	B1	B1	B3	B1~B2
	C3: 性别	—	—	S1	S2	—	—
动作	A1: 脱产学习	✓					
	A2: 电工		✓				
	A3: 钳工			✓			
	A4: 车工				✓		
	A5: 技术员					✓	
	A6: 材料员						✓

## Part2: 实践题目

### 1. 安装火狐浏览器以及 Selenium IDE 扩展组件



### 2. 打开 Selenium IDE



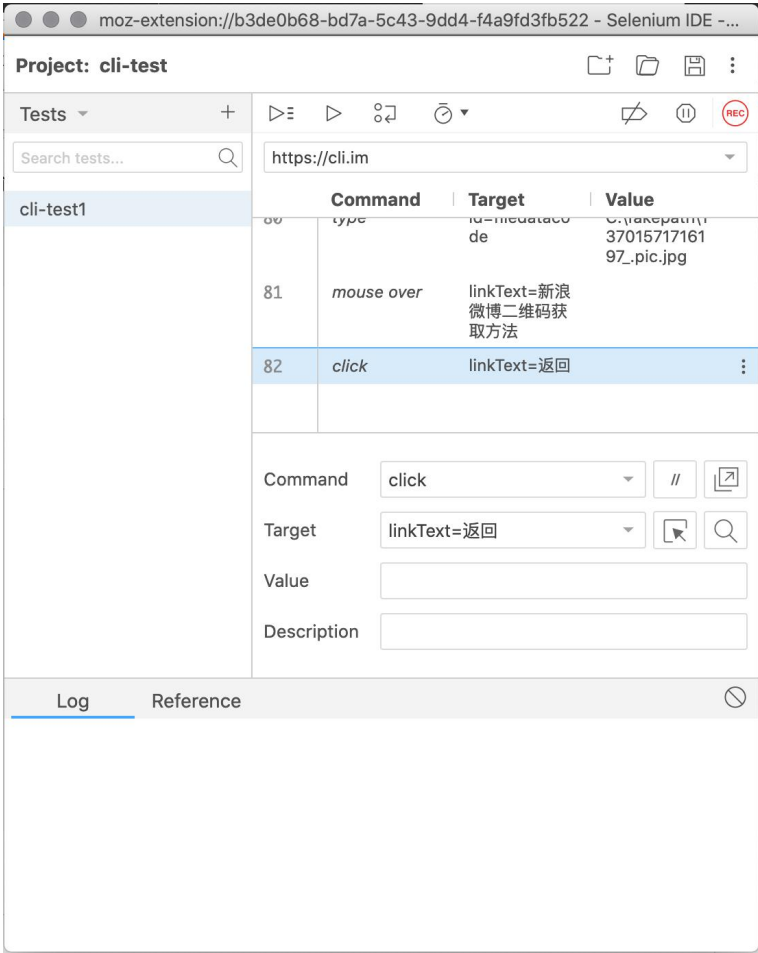
### 3. 选择“Record a new test in a new test”，并新建文件名为“cli-test”，输入 BASE URL 为“https://cli.im/”



3. 在草料网站上依次测试文本、网址、文件、图片、音视频、名片以及更多工具，所有功能正常运行.



停止录制





5. 导出测试脚本为 python 文件如下.

```
# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action_chains import ActionChains

from selenium.webdriver.support import expected_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys


class TestClitest1():

    def setup_method(self, method):

        self.driver = webdriver.Firefox()

        self.vars = {}
```

```
def teardown_method(self, method):

    self.driver.quit()


def wait_for_window(self, timeout = 2):

    time.sleep(round(timeout / 1000))

    wh_now = self.driver.window_handles

    wh_then = self.vars["window_handles"]

    if len(wh_now) > len(wh_then):

        return set(wh_now).difference(set(wh_then)).pop()


def test_clitest1(self):

    # Test name: cli-test1

    # Step # | name | target | value | comment

    # 1 | open | / | |

    self.driver.get("https://cli.im/")

    # 2 | setWindowSize | 550x691 | |

    self.driver.set_window_size(550, 691)
```

# 3 | type | id=text-content | hello selenium ! |

```
self.driver.find_element(By.ID, "text-content").send_keys("hello  
selenium !")
```

# 4 | click | id=click-create | |

```
self.driver.find_element(By.ID, "click-create").click()
```

# 5 | click | linkText=网址 | |

```
self.driver.find_element(By.LINK_TEXT, "网址").click()
```

# 6 | click | id=url\_content | |

```
self.driver.find_element(By.ID, "url_content").click()
```

# 7 | type | id=url\_content | baidu.com |

```
self.driver.find_element(By.ID, "url_content").send_keys("baidu.com")
```

# 8 | click | id=click-create | |

```
self.driver.find_element(By.ID, "click-create").click()
```

# 9 | click | linkText=文件 | |

```
self.driver.find_element(By.LINK_TEXT, "文件").click()
```

# 10 | type | id=filedatacode | C:\fakepath\南开大学校史、校情、校庆活  
动介绍.docx |

```
self.driver.find_element(By.ID,
```

```
"filedatacode").send_keys("C:\\fakepath\\南开大学校史、校情、校庆活动介绍.docx")
```

```
# 11 | click | id=filedatacode | |
```

```
self.driver.find_element(By.ID, "filedatacode").click()
```

```
# 12 | click | id=click-create | |
```

```
self.driver.find_element(By.ID, "click-create").click()
```

```
# 13 | click | linkText=文件 | |
```

```
self.driver.find_element(By.LINK_TEXT, "文件").click()
```

```
# 14 | click | id=filedatacode | |
```

```
self.driver.find_element(By.ID, "filedatacode").click()
```

```
# 15 | mouseOver | linkText=添加更多内容 | |
```

```
element = self.driver.find_element(By.LINK_TEXT, "添加更多内容")
```

```
actions = ActionChains(driver)
```

```
actions.move_to_element(element).perform()
```

```
# 16 | mouseOut | linkText=添加更多内容 | |
```

```
element = self.driver.find_element(By.CSS_SELECTOR, "body")
```

```
actions = ActionChains(driver)
```

```
actions.move_to_element(element, 0, 0).perform()
```

```
# 17 | click | id=click-create | |

self.driver.find_element(By.ID, "click-create").click()

# 18 | click | linkText=图片 | |

self.driver.find_element(By.LINK_TEXT, "图片").click()

# 19 | click | id=filedatacode | |

self.driver.find_element(By.ID, "filedatacode").click()

# 20 | click | id=click-create | |

self.driver.find_element(By.ID, "click-create").click()

# 21 | click | linkText=音视频 | |

self.driver.find_element(By.LINK_TEXT, "音视频").click()

# 22 | click | id=filedatacode | |

self.driver.find_element(By.ID, "filedatacode").click()

# 23 | type | id=filedatacode | C:\fakepath\WeChatSight52.mp4 |

self.driver.find_element(By.ID,
"filedatacode").send_keys("C:\\fakepath\\WeChatSight52.mp4")

# 24 | mouseOver | linkText=添加更多内容 | |

element = self.driver.find_element(By.LINK_TEXT, "添加更多内容")

actions = ActionChains(driver)
```

```
actions.move_to_element(element).perform()

# 25 | mouseOut | linkText=添加更多内容 | |

element = self.driver.find_element(By.CSS_SELECTOR, "body")

actions = ActionChains(driver)

actions.move_to_element(element, 0, 0).perform()

# 26 | click | id=click-create | |

self.driver.find_element(By.ID, "click-create").click()

# 27 | click | linkText=名片 | |

self.driver.find_element(By.LINK_TEXT, "名片").click()

# 28 | click | name=name | |

self.driver.find_element(By.NAME, "name").click()

# 29 | type | name=name | Ruby |

self.driver.find_element(By.NAME, "name").send_keys("Ruby")

# 30 | type | name=mobile | 173000000000 |

self.driver.find_element(By.NAME,
"mobile").send_keys("173000000000")

# 31 | click | id=click-create | |

self.driver.find_element(By.ID, "click-create").click()
```

```
# 32 | click | name=mobile | |
```

```
self.driver.find_element(By.NAME, "mobile").click()
```

```
# 33 | mouseDownAt | id=click-create | 37.5,19 |
```

```
element = self.driver.find_element(By.ID, "click-create")
```

```
actions = ActionChains(driver)
```

```
actions.move_to_element(element).click_and_hold().perform()
```

```
# 34 | mouseMoveAt | id=click-create | 37.5,19 |
```

```
element = self.driver.find_element(By.ID, "click-create")
```

```
actions = ActionChains(driver)
```

```
actions.move_to_element(element).perform()
```

```
# 35 | mouseUpAt | id=click-create | 37.5,19 |
```

```
element = self.driver.find_element(By.ID, "click-create")
```

```
actions = ActionChains(driver)
```

```
actions.move_to_element(element).release().perform()
```

```
# 36 | click | id=click-create | |
```

```
self.driver.find_element(By.ID, "click-create").click()
```

```
# 37 | click | css=.left-info > .p-vcard-info:nth-child(3) > input | |
```

```
self.driver.find_element(By.CSS_SELECTOR,  
".left-info > .p-vcard-info:nth-child(3) > input").click()  
  
# 38 | click | name=mobile | |  
  
self.driver.find_element(By.NAME, "mobile").click()  
  
# 39 | type | name=mobile | 17300000000 |  
  
self.driver.find_element(By.NAME,  
"mobile").send_keys("17300000000")  
  
# 40 | click | id=click-create | |  
  
self.driver.find_element(By.ID, "click-create").click()  
  
# 41 | click | css=#build-code-modal .close > span:nth-child(1) | |  
  
self.driver.find_element(By.CSS_SELECTOR,  
"#build-code-modal .close > span:nth-child(1)").click()  
  
# 42 | mouseOver | css=#build-code-modal .close > span:nth-child(1)  
| |  
  
element = self.driver.find_element(By.CSS_SELECTOR,  
"#build-code-modal .close > span:nth-child(1)")  
  
actions = ActionChains(driver)  
  
actions.move_to_element(element).perform()
```



# 43 | click | linkText=微信 | |

```
self.driver.find_element(By.LINK_TEXT, "微信").click()
```

# 44 | click | id=weixinurl | |

```
self.driver.find_element(By.ID, "weixinurl").click()
```

# 45 | click | linkText=个人帐号 | |

```
self.driver.find_element(By.LINK_TEXT, "个人帐号").click()
```

# 46 | click | css=.pointer > #filedatacode | |

```
self.driver.find_element(By.CSS_SELECTOR, ".pointer >  
#filedatacode").click()
```

# 47 | click | linkText=公众号 | |

```
self.driver.find_element(By.LINK_TEXT, "公众号").click()
```

# 48 | click | id=tab1 | |

```
self.driver.find_element(By.ID, "tab1").click()
```

# 49 | click | id=weixinurl | |

```
self.driver.find_element(By.ID, "weixinurl").click()
```

# 50 | type | id=weixinurl | nankaiuni |

```
self.driver.find_element(By.ID, "weixinurl").send_keys("nankaiuni")
```

# 51 | click | id=click-create | |

```
self.driver.find_element(By.ID, "click-create").click()

# 52 | click | linkText=更多工具 | |

self.driver.find_element(By.LINK_TEXT, "更多工具").click()

# 53 | click | css=.tools-item:nth-child(1) > h3 | |

self.vars["window_handles"] = self.driver.window_handles

# 54 | selectWindow | handle=${win78} | |

self.driver.find_element(By.CSS_SELECTOR, ".tools-item:nth-child(1) >
h3").click()

# 55 | click | linkText=立即使用 | |

self.vars["win78"] = self.wait_for_window(2000)

# 56 | selectWindow | handle=${win2270} | |

self.driver.switch_to.window(self.vars["win78"])

# 57 | click | id=qrgenerate-link | |

self.vars["window_handles"] = self.driver.window_handles

# 58 | click | css=.form-control | |

self.driver.find_element(By.LINK_TEXT, "立即使用").click()

# 59 | type | css=.form-control | 12313AB |

self.vars["win2270"] = self.wait_for_window(2000)
```

```
# 60 | click | id=generate | |

self.driver.switch_to.window(self.vars["win2270"])

# 61 | mouseOver | id=generate | |

self.driver.find_element(By.ID, "qrgenerate-link").click()

# 62 | assertAlert | 原授权错误, 请重新授权 | |

self.driver.find_element(By.CSS_SELECTOR, ".form-control").click()

# 63 | mouseOut | id=generate | |

self.driver.find_element(By.CSS_SELECTOR,
".form-control").send_keys("12313AB")

# 64 | click | css=a > .inline | |

self.driver.find_element(By.ID, "generate").click()

# 65 | click | linkText=示例模板生码 | |

element = self.driver.find_element(By.ID, "generate")

actions = ActionChains(driver)

actions.move_to_element(element).perform()

# 66 | click | css=.form-control | |

assert self.driver.switch_to.alert.text == "原授权错误, 请重新授权"

# 67 | click | css=.main-tab > li:nth-child(2) | |
```

```
element = self.driver.find_element(By.CSS_SELECTOR, "body")

actions = ActionChains(driver)

actions.move_to_element(element, 0, 0).perform()

# 68 | click | css=#step-btn-group > .btn:nth-child(1) | |

self.driver.find_element(By.CSS_SELECTOR, "a > .inline").click()

# 69 | mouseOver | css=#step-btn-group > .btn:nth-child(1) | |

self.driver.find_element(By.LINK_TEXT, "示例模板生码").click()

# 70 | assertAlert | 示例模板不支持批量生码! | |

self.driver.find_element(By.CSS_SELECTOR, ".form-control").click()

# 71 | mouseOut | css=#step-btn-group > .btn:nth-child(1) | |

self.driver.find_element(By.CSS_SELECTOR, ".main-tab >
li:nth-child(2)").click()

# 72 | selectWindow | handle=${win78} | |

self.driver.find_element(By.CSS_SELECTOR,
"#step-btn-group > .btn:nth-child(1)").click()

# 73 | click | linkText=更多工具 | |

element = self.driver.find_element(By.CSS_SELECTOR,
"#step-btn-group > .btn:nth-child(1)")
```

```
actions = ActionChains(driver)

actions.move_to_element(element).perform()

# 74 | click | css=.weibo | |

assert self.driver.switch_to.alert.text == "示例模板不支持批量生码! "

# 75 | selectWindow | handle=${win8230} | |

element = self.driver.find_element(By.CSS_SELECTOR, "body")

actions = ActionChains(driver)

actions.move_to_element(element, 0, 0).perform()

# 76 | click | id=weibourl | |

self.driver.switch_to.window(self.vars["win78"])

# 77 | click | linkText=新浪微博二维码获取方法 | |

self.driver.find_element(By.LINK_TEXT, "更多工具").click()

# 78 | click | id=weibourl | |

self.vars["window_handles"] = self.driver.window_handles

# 79 | click | id=filedatacode | |

self.driver.find_element(By.CSS_SELECTOR, ".weibo").click()

# 80 | type | id=filedatacode | C:\fakepath\13701571716197_.pic.jpg |
```

```
self.vars["win8230"] = self.wait_for_window(2000)

# 81 | mouseOver | linkText=新浪微博二维码获取方法 | |

self.driver.switch_to.window(self.vars["win8230"])

# 82 | click | linkText=返回 | |

self.driver.find_element(By.ID, "weibourl").click()

self.driver.find_element(By.LINK_TEXT, "新浪微博二维码获取方法").click()

self.driver.find_element(By.ID, "weibourl").click()

self.driver.find_element(By.ID, "filedatacode").click()

self.driver.find_element(By.ID, "filedatacode").send_keys("C:\\fakepath\\13701571716197_.pic.jpg")

element = self.driver.find_element(By.LINK_TEXT, "新浪微博二维码获取方法")

actions = ActionChains(driver)

actions.move_to_element(element).perform()

self.driver.find_element(By.LINK_TEXT, "返回").click()
```