

# 蓝桥杯全国软件和信息技术专业人才大赛组委会

## 第十四届蓝桥杯全国软件和信息技术专业人才大赛个人赛 (软件类) 软件测试组竞赛规则及说明

### 1. 组别

具有正式学籍的在校全日制研究生、本科及高职高专学生（以报名时状态为准）可报名参加，以个人为单位进行比赛，本次比赛设大学组。

### 2. 竞赛赛程

省赛时长：4 小时。

决赛时长：4 小时。

详细赛程安排以组委会公布信息为准。

### 3. 竞赛形式

原则上采用线下比赛的方式，如因疫情等原因无法线下比赛的选手将采取线上比赛的方式。

#### 3.1 线下比赛：

个人赛，一人一机，全程机考。

选手机器通过局域网连接到各个赛场的竞赛服务器。

选手答题过程中无法访问互联网，也不允许使用本机以外的资源（如 USB 连接）。

选手将答案提交到比赛系统中，超过比赛时间将无法提交。

竞赛系统以“服务器-浏览器”方式发放试题、回收选手答案。

#### 3.2 线上比赛：

个人赛，自备机器和自行安装比赛环境。

选手机器通过浏览器输入准考证上的线上比赛网址进行登录比赛。

选手将答案提交到比赛系统中，超过比赛时间将无法提交。

选手须按照线上比赛手册要求准备相关软件、硬件和网络，并保证遵守线上比赛手册上的要求进行比赛，不得违规作弊。

线上比赛手册计划于 2023 年 3 月在大赛官网发布。

### 4. 赛点及参赛选手机器环境

#### 4.1 赛点局域网服务器

(1) Win7、Win8 或 Win10 操作系统（推荐使用），内存不低于 8G，硬盘不小于 120G

(2) 显示器：分辨率 1024\*768 像素或以上

#### 4.2 参赛选手机器环境

(1) Win7、Win8 或 Win10 操作系统（推荐使用），内存不低于 4G，硬盘不小于 60G

(2) 显示器：分辨率 1024\*768 像素或以上

(3) Office 工具套件, 包括: Excel 和 Word

**选择 Java 语言参赛时需要安装:**

JDK 1.8

Eclipse-java-2020-06

Selenium-4.4.0 工具

Google Chrome v90

chromedriver\_win32.zip 90.0.4430.24

Junit-4.10.jar 单元测试工具

**选择 Python 语言参赛时需要安装:**

Python 3.8.6

VS code v1.36

Selenium-4.4.0 工具

Google Chrome v90

chromedriver\_win32.zip 90.0.4430.24

## 5. 赛题类型、数量、分值、评分标准、评分细则

- (1) 试题总分为 150 分, 选手所提交答案的测评结果将会作为评分依据。
- (2) 选手可自行选择 Java 或者 Python 语言中的任意一门编程语言参赛。
- (3) 竞赛题目分为 3 个部分, 功能测试、自动化测试和单元测试。
- (4) 共有 2 个题型, 具体题型及题目数量以正式比赛时赛题为准。

设计题: 功能测试属于主观题, 主要考察测试用例的设计能力以及发现缺陷的能力, 选手基于被测系统, 按照试题要求, 使用大赛提供的固定模板的 Excel 文件填写答案。

编程大题: 自动化测试和单元测试属于编程题, 题目包含明确的问题描述、输入和输出格式, 以及用于解释问题的样例数据。编程大题所涉及的问题一定是有明确客观的标准来判断结果是否正确, 并可以通过程序对结果进行评判。

具体说明如下:

### 5.1 功能测试

表 5-1 功能测试赛程表

功能测试赛程	数量	分值	考察点	评分标准	备注
设计测试用例	1	35	测试用例文档应包括: 用例编号、功能模块、测试点、前提条件、操作步骤、测试数据、预期结果、用例级别等。	1、按照设计的测试用例级别进行判分。用例级别分为 3 级, 标准详见下表 6-1 所列。 (重复项不得分) 2、测试用例编写符合测试用例规范。	
提交缺陷报告	1	35	Bug 提交文档包括: 缺陷编号、模块、摘要、操作步骤、预期结果、实际结果、严重程度、提交人、截图等。	1、按照发现的 bug 级别得分。Bug 级别分为 4 级, 分级标准详见下表 6-2 所列。(重复项不得分) 2、测试 bug 清单编写符合规范, 每一项都需要填写, 否则得 0 分	

### 5.2 自动化测试

表 5-2 自动化测试赛程表

题型	数量	分值	考察点	评分标准	备注
----	----	----	-----	------	----

自动化测试编程	1	50	<ul style="list-style-type: none"> <li>- Selenium 引入 WebDriver</li> <li>- 启动浏览器</li> <li>- 浏览器窗口操作</li> <li>- 浏览器导航操作</li> <li>- 多种元素定位器</li> <li>- 元素的基本操作方法</li> <li>- 时间等待方法</li> <li>- 下拉框操作方法</li> <li>- Alert 弹出框方法</li> <li>- 附件上传操作方法</li> <li>- 下拉滚动条方法</li> <li>- 窗口及 Frame 切换方法</li> <li>- 页面元素属性删除方法</li> <li>- 参数化和数据驱动</li> <li>- 断言</li> <li>- 模拟鼠标键盘 Action 方法</li> <li>- 页面截图操作</li> <li>- 读取 csv 文件</li> <li>- TestNg 自动化测试框架</li> <li>- PO 模式</li> </ul>	<ol style="list-style-type: none"> <li>1、考察运行步骤：脚本设置正确，不影响运行结果。</li> <li>2、运行结果分析：通过运行结果分析，验证场景所设置的参数，是否满足预期目标值。</li> </ol>	<p>主要考察工具基本操作、selenium 与脚本语言的使用以及 Selenium 控件与参数化应用。</p> <p>提前给考生提供 Selenium 开发环境搭建手册。Python 和 Java 语言二选一。</p>
---------	---	----	---	---	--

### 5.3 单元测试

表 5-3 单元测试赛程表

题型	数量	分值	考察点	评分标准	备注
单元测试	1	30	<ul style="list-style-type: none"> <li>- 语句覆盖法</li> <li>- 判定覆盖法</li> <li>- 条件覆盖法</li> <li>- 判定条件覆盖法</li> <li>- 条件组合覆盖法</li> <li>- 路径覆盖法</li> <li>- 基本路径覆盖法</li> <li>- Junit/UnitTest 基础注解</li> <li>- Junit/UnitTest 断言方法</li> <li>- Junit/UnitTest 参数化/数据驱动</li> <li>- Junit/UnitTest 测试套件</li> <li>- Junit/UnitTest 测试优先级顺序</li> <li>- Junit/UnitTest 注释</li> <li>- Junit/UnitTest 异常</li> <li>- Junit/UnitTest TestRunner 内置方法</li> <li>- Junit/UnitTest 编码规范</li> </ul>	<ol style="list-style-type: none"> <li>1、规范检查：单元测试的编码符合规范。</li> <li>2、覆盖率：功能代码的测试充分覆盖。</li> <li>3、框架使用：单元测试框架的掌握程度。</li> <li>4、运行结果分析：通过运行结果分析，验证单元测试的通过率。</li> </ol>	<p>使用 Java 或 Python 语言编写应用程序，设计测试数据，使用 Junit 或者 UnitTest 框架对应用程序编写单元测试代码，使用设计的测试数据完成单元测试。</p> <p>提前给考生提供单元测试工具及环境搭建操作手册。Python 和 Java 语言二选一。</p>

## 6. 用例级别和缺陷级别定义

表 6-1 测试用例级别定义

用例级别	定义描述

高	一组需要优先被执行以确认该软件版本是否可以继续测试的用例，即冒烟测试用例。如果这部分用例测试失败会阻碍其他测试用例的验证。
中	1、是最常被执行、保证功能稳定、目标的行为和能力正常工作的、能发现重要错误的测试用例。 2、可以更全面的验证功能的各个方面的用例，主要指异常测试、如边界、断网、容错和配置测试的测试用例。
低	一组最少被执行的测试用例，也就是在项目的声明周期里不是常常被执行，例如 GUI，错误信息，可用性，压力和性能测试等。

表 6-2 缺陷级别定义	
缺陷级别	定义描述
致命缺陷	指系统任何一个主要功能完全丧失，用户数据受到破坏、系统崩溃、悬挂、死机或者危及人身安全的缺陷。或者系统所提供的功能或服务受到明显的限制，不能执行正常工作流程或实现重要功能。包括： <ul style="list-style-type: none"> <li>- 可能有灾难性的后果，如造成系统崩溃，造成事故的缺陷等；</li> <li>- 数据库错误，如数据丢失、数据毁坏等；</li> <li>- 安全性被破坏等。</li> </ul>
严重缺陷	1、指可能导致系统不稳定，运行时好时坏，严重影响系统要求或基本功能实现的缺陷。如： <ul style="list-style-type: none"> <li>- 造成数据库不稳定的错误；</li> <li>- 在说明中的需求未在最终系统中实现；</li> <li>- 程序无法运行，系统意外退出；</li> <li>- 业务流程不正确等。</li> </ul> 2、次要功能没有完全实现，但不影响用户的正常使用，不会影响系统稳定性的缺陷。
一般缺陷	指使操作者不方便或遇到麻烦，但它不影响功能的操作和执行，如个别不影响产品理解的错别字、文字排列不整齐等一些小问题，重点指系统的 UI 问题。如： <ul style="list-style-type: none"> <li>- 系统的提示语不明确，不简单明了；</li> <li>- 滚动条无效；</li> <li>- 可编辑区域和不可编辑区域不明显；</li> <li>- 光标跳转设置不好，鼠标（光标）定位错误；</li> <li>- 上下翻页，首尾页定位错误；</li> <li>- 界面不一致，或界面不正确；</li> <li>- 日期或时间初始值错误（起止日期、时间没有限定）；</li> <li>- 出现错别字，标点符号错误，拼写错误，以及不正确的大小写等。</li> </ul>
改进建议	系统中值得改良的问题。比如： <ul style="list-style-type: none"> <li>- 容易给用户错误和歧义的提示；</li> <li>- 界面需要改进的；</li> <li>- 某个控件没有对齐等；</li> <li>- 测试人员有疑虑的部分；</li> <li>- 提出的修改建议。</li> </ul>

## 7. 答案提交

- (1) 选手只有在比赛时间内提交的答案是有效的，比赛之后的任何提交均无效。
- (2) 选手须使用比赛指定的方式来提交答案，任何其他方式的提交（如邮件、U 盘）都不作为评测依据。

(3) 每道试题可以重复提交答案，以最后一次提交的答案为准并作为评测的依据。

(4) 比赛过程中，试题分数不会显示给选手，选手应当在没有反馈的情况下自行设计数据调试自己的程序。

(5) 程序中应只包含计算模块，不要包含任何其他的模块，比如图形、系统接口调用、系统中断等。对于系统接口的调用都应通过标准库来进行。

(6) 选手须仔细阅读并严格遵守试题指定的答案文件格式或内容。

## 8. 其他注意事项

(1) 参赛选手应遵守竞赛规则，遵守赛场纪律，如有任何违规，将会被组委会取消报名或竞赛资格。

(2) 竞赛采用机器阅卷+少量人工辅助。选手要特别注意提交答案的形式。必须仔细阅读题目的要求和示例，不要随意添加不需要的内容。

## 9. 样题

### 9.1 功能测试样题：

#### 【题目介绍】

模块名称	功能名称	需求描述
用户中心	注册	1、账号：11 位手机号 2、密码：8-20 位，同时含有数字、大写、小写字母
	登录	输入用户名和密码
	修改密码	不能和原密码相同
	找回密码	短信验证码验证找回
	注销	短信验证码注销
搜索	全站搜索	支持全站搜索
	当前频道搜索	仅支持当前频道搜索

#### 【题目要求】

根据测试用例模板，按照需求描述，使用测试用例常用方法对题目中的功能模块设计测试用例，寻找该模块中存在的缺陷，并提交发现的缺陷。

#### 【答案提交】

#### 【测试用例 Excel 答案】

用例编号	测试对象	测试点	测试环境	操作步骤	测试输入	用例级别	预期结果
LQ-Login-001	登录	测试登录时，输入正确的用户名和正确的密码，系统的响应情况	测试环境已搭建完毕	1.打开 LQ 登录界面;2.在“用户名”输入框输入正确的用户名;3.在“密码”输入框输入正确的密码;4.点击“登录”。	用户名：dongling 密码：123456	BVTs	成功登录系统，并跳转至个人中心
LQ-Login-002	登录	测试登录时，输入正确的用户名、错误的密码（少于标准位数），系统的响应情况	测试环境已搭建完毕	1.打开 LQ 登录界面;2.在“用户名”输入框输入正确的用户名;3.在“密码”输入框输入错误的密码（少于标准位数，取 5 位）;4.点击“登录”。	用户名：董岭 密码：12345	高	登录 LQ 系统失败，并提示“用户名或密码错误。”

#### 【缺陷报告 Excel 答案】

缺陷编号	功能模块	缺陷描述	操作步骤	预期结果	实际结果	附件说明
------	------	------	------	------	------	------

号	块			果		
1	登录	用户登录时, 输入错误的验证码, 依然登录成功	1.主页——>点击【会员登录】, 打开登录页面; 2.输入正确的用户名; 3.输入正确的密码; 4.输入错误的验证码; 5.点击【登录】	登录成功,跳转到用户页面。	登录失败, 页面提示“验证码错误”。	<a href="C:\Users\semmy\Desktop\1.png">C:\Users\semmy\Desktop\1.png</a>
2	登录	用户提交订单, 当收货地址为空时, 订单仍正常生成	1.主页 2.点击【搜索】按钮, 展开图书列表, 任意点击一本图书, 展开该图书信息 3.点击【购买】, 页面跳转到“我的购物车”页面; 4.选择所需要购买的图书, 点击【结算】; 5.收货地址为空 6.点击【提交订单】	跳转到订单生成页面, 订单成功生成	购买失败, “生成订单”页面提示“收货地址不能为空!”	<a href="C:\Users\semmy\Desktop\1.png">C:\Users\semmy\Desktop\1.png</a>

说明：这是一道验证测试人员对功能点规则理解是否准确的题目，答题者只需要按照日常所学的测试用例方法，对题目中出现的功能进行测试用例覆盖即可。本题没有固定答案，答题者提交的答案符合题目中需求描述，即可得分。

根据测试用例模板，必须填写与题目要求相匹配的名称，判分会根据有无错别字，测试用例是否符合对应功能需求描述等方面。针对每一个题目官方会默认一个测试用例数，答案必须符合测试用例范围数内，不符合扣分，符合或者多余官方默认用例数进行冗余扣分。

## 9.2 Java 自动化测试编程样题：

### 【题目介绍】

使用 Java 语言，通过 Selenium 工具来完成《悦购图书商城》注册模块的自动化测试脚本。

- 1、引入 WebDriver 在本地的地址到代码中；
- 2、使用 Edge 浏览器打开以下页面；
- 3、打开悦购图书商城前台页面；
- 4、设置隐式等待，最长等待 10 秒；
- 5、页面全屏；
- 6、切换 frame，点击【会员注册】按钮；
- 7、页面全屏；
- 8、在用户名、密码、确认密码、邮箱输入框中输入内容；
- 9、操作第三方 OCR 来识别验证码图片内容，并填入验证码输入框；
- 10、点击【确认注册】按钮；
- 11、插入断言，检测跳转页面的标题是否正确；
- 12、退出 driver。

要求测试用例文件必须命名为：RegisterTest.java 或 register\_test.py。

### 【参考答案】

```
package com.lanqiao;

import java.io.File;
```

```
import java.io.IOException;
import java.time.Duration;

import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.edge.EdgeDriver;
import org.openqa.selenium.remote.RemoteWebDriver;
import org.testng.Assert;
import org.testng.annotations.Test;

public class RegisterTest {

    //定义一个 RemoteWebDriver 实例
    RemoteWebDriver driver;

    @Test
    public void testRigister() throws IOException, InterruptedException{

        //引入 WebDriver 在本地的地址到代码中
        System.setProperty("webdriver.edge.driver",
"C:/Software/edgedriver_win64/msedgedriver.exe");

        //使用 Edge 浏览器打开以下页面
        driver =new EdgeDriver();

        //打开悦购图书商城前台页面
        driver.get("https://86f2c3d0d142.hz-iframe-svc.simplelab.cn/goods/");

        //设置隐式等待，最长等待 10 秒
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        //页面全屏
        driver.manage().window().fullscreen();

        //切换 frame，点击【会员注册】按钮
        driver.switchTo().frame(0);
        driver.findElement(By.xpath("/html/body/div/div/a[2]")).click();

        //页面全屏
        driver.manage().window().fullscreen();

        //在用户名、密码、确认密码、邮箱输入框中输入内容
```

```
driver.findElement(By.id("loginname")).sendKeys("dongling");
driver.findElement(By.id("loginpass")).sendKeys("123456");
driver.findElement(By.id("reloginpass")).sendKeys("123456");
driver.findElement(By.id("email")).sendKeys("dongling1@163.com");

Thread.sleep(2000);

//点击【确认注册】按钮
driver.findElement(By.id("submitBtn")).click();
Thread.sleep(2000);

//插入断言，如果成功之后的页面标题是“信息板”，就测试通过，否则提示“注册失败”
Assert.assertEquals(driver.getTitle(), "信息板", "注册失败");
Thread.sleep(3000);

driver.quit();

}
}
```

#### 【答案提交】

本题针对日常自动化回归测试中遇到的定位方法、判断等日常操作进行考核，考核者只需要按照题目要求进行录制或者手动修改，保证提交的脚本不报错为前提，脚本覆盖度符合题目要求中的功能点即可得分，漏掉操作步骤脚本响应分值扣除。

说明：答案考核会根据题目中每一步要求点，进行单项考核，并对整体运行结果进行整体考核；单项考核看答案是否根据题目要求录制或者编写，不符合项扣分；整体运行结果主要看运行是否成功，运行效果是否符合题目要求，比如某一操作打开新的窗体有没有成功。备注：Python 语言版本答案，此处略。

### 9.3 Python 单元测试编程样题：

#### 【题目介绍】

Hello, World! 是经典的编程入门问题，一般会是你在学习一门新的语言前，书写的第一个程序。

题目给出了一个函数，返回字符串“Hello, World!”。

源代码题目实现的功能：

1. 函数返回字符串为：Hello, World!
2. 返回字符串中标点类型，空格数量必须一模一样。

#### 【题目要求】

请从 Junit4 (Java) 或 unittest (Python) 中任选其一，编写给出示例函数的单元测试用例，需正确测试题目所列功能点。

测试用例文件必须命名为 GreeterTest.java 或 hello\_world\_test.py

### 【题目源代码】

题目提供的基础代码如下：

```
# 文件名 hello_world.py

# 请勿修改题目提供基础代码的文件名和函数名，基础代码不一定完整和正确；

def hello():

    return 'Goodbye, Mars!'

// 文件名 Greeter.java

// 请勿修改题目提供基础代码的文件名和函数名，基础代码不一定完整和正确；

class Greeter {

    String getGreeting() {

        return "Goodbye, Mars!";

    }

}
```

### 【参考答案】

```
// GreeterTest.java

import org.junit.Test;
import static org.assertj.core.api.Assertions.assertThat;

public class GreeterTest {

    @Test
    public void testThatGreeterReturnsTheCorrectGreeting() {
        assertThat(new Greeter().getGreeting()).isEqualTo("Hello, World!");
    }

}

# hello_world_test.py

import unittest
from hello_world import (
    hello,
```

```
)
```

```
class HelloWorldTest(unittest.TestCase):
```

```
    def test_say_hi(self):
```

```
        self.assertEqual(hello(), "Hello, World!")
```

```
if __name__ == "__main__":
```

```
    unittest.main()
```

备注：Java 语言单元测试版本，此处略。