Software quality – assignment 2

* **Task** **1**
* does not work on firefox browser (shows message „IE not supported“) – should be specified
* parameters can be adjusted through URL
* output shows up even when there is one empty parameter – should be only when both are not empty
* output differs from the one in assignment – shows weight also
* input accepts symbols and numbers – since this is first and last name it should be only alphabet letters
* I used specification/requirements-based testing approach, there are more approaches that can be used – ACC method, Risk based testing
* Using Risk based testing approach might verify this application better and find more bugs
* Specification is buggy, there are missing elements, product manager should specify what exactly can be used as input (Upper case, symbols, etc), specify error messages, specify browsers it will run on, security (URL changing)
* **Task 2**
* Attributes: secure, reliable, fast, user friendly, testable, cross-platform, adjustable, supported
* Components: login, registration, email confirmations and notifications, orders, bank transfers, support, legal proclamations, permissions, administration ,advertisements, information storage, dealership communication, shipping, browser compatibility, display of webpage, display of application, error logging, items, comments
* Capabilities: payments in different ways (card, bank transfer cash,..), option to choose transport, return of product, choose currency, choose language, choose pick up place, find customer position – suggest closest pick up places, prefill order fields from customer profile information, cancel order, registration over HTTPS, special attributes for password, SMS check for registration/login/order confirmation/cancellation confirmation, email new discounts, change personal information, delete account, write a review
* Test cases: go to <http://eshop.com/us> change language to czech and check you are redirected to <http://eshop.com/cz>, create an account with given id – check the account exists in database, delete account – check the account does not exist in database, create account with some mail address – try to create another one with the same mail address – fail, create account – recieve confirmation email/SMS, create order – check order in database, cancel order – find order in database as cancelled
* Clusters of impact: not getting money from customer due to transfer issues, not getting money from customer due to bad security, customer not getting what he ordered, customer not getting order in time, customer not getting order at all, customer unavailable to cancel order, customer unavailable to reach customer support, customer unavailable to login into his account
* Attributes and the clusters of impact we chose are of the highest importance because when it comes to online shopping the customer is on the first place always, he must be satisfied and feel comfortable all the way through from registration to making and order until he recieves it, every little mistake can drive him out of our site or make him leave a nasty report or a review, we minimized the chances by focusing on the customer experience the most, which makes the most out of our site when it comes to maximizing profit and human advertisement

import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.chrome.ChromeDriver;  
  
import java.util.List;  
  
import static org.junit.Assert.*assertTrue*;  
  
public class testClass {  
 private WebDriver webDriver;  
 private final static String *WEB\_DRIVER* = "webdriver.chrome.driver";  
 private final static String *WEB\_DRIVER\_PATH* = "c:\\Users\\mokro\\Downloads\\chromedriver.exe";  
  
 @Before  
 public void setup() {  
 System.*setProperty*(*WEB\_DRIVER*, *WEB\_DRIVER\_PATH*);  
 webDriver = new ChromeDriver();  
 webDriver.get("http://en.wikipedia.org");  
 }  
  
 @After  
 public void close() {  
 webDriver.close();  
 }  
  
 @Test  
 public void testSeleniumNegativeWikipediaLogin() {  
 webDriver.findElement(By.*id*("pt-login")).click();  
 WebElement usernameField = webDriver.findElement(By.*name*("wpName"));  
 WebElement passwordField = webDriver.findElement(By.*name*("wpPassword"));  
 usernameField.sendKeys("\*user");  
 passwordField.sendKeys("1234");  
 *assertTrue*(webDriver.findElements(By.*className*("error")).isEmpty());  
 webDriver.findElement(By.*id*("wpLoginAttempt")).click();  
 *assertTrue*(webDriver.findElement(By.*className*("error")).isDisplayed());  
 }  
  
 @Test  
 public void testSeleniumPositiveWikipediaHelpPage() {  
 WebElement helpField = webDriver.findElement(By.*id*("n-help"));  
 helpField.findElement(By.*tagName*("a")).click();  
 *assertTrue*(webDriver.getTitle().startsWith("Help:Contents"));  
 }  
  
 @Test  
 public void testSeleniumPositiveWikipediaLanguageChange() {  
 List<WebElement> languageFields = webDriver.findElements(By.*className*("interlanguage-link-target"));  
  
 for (WebElement field : languageFields) {  
 if (field.getAttribute("lang").equals("cs")) {  
 field.click();  
 break;  
 }  
 }  
  
 *assertTrue*(webDriver.getCurrentUrl().equals("https://cs.wikipedia.org/wiki/Hlavn%C3%AD\_strana"));  
 }  
}