**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**RAPORT**

**la lucrarea de laborator Nr.3**

**Tema: „Selenium IDE”**

**Disciplina: „Testarea Software”**

**Efectuat:** st. gr. SI-211: Chihai Adrian

**Verificat:** dr. conf. Univ. Andrian Prisăcaru

**Chișinău 2024**

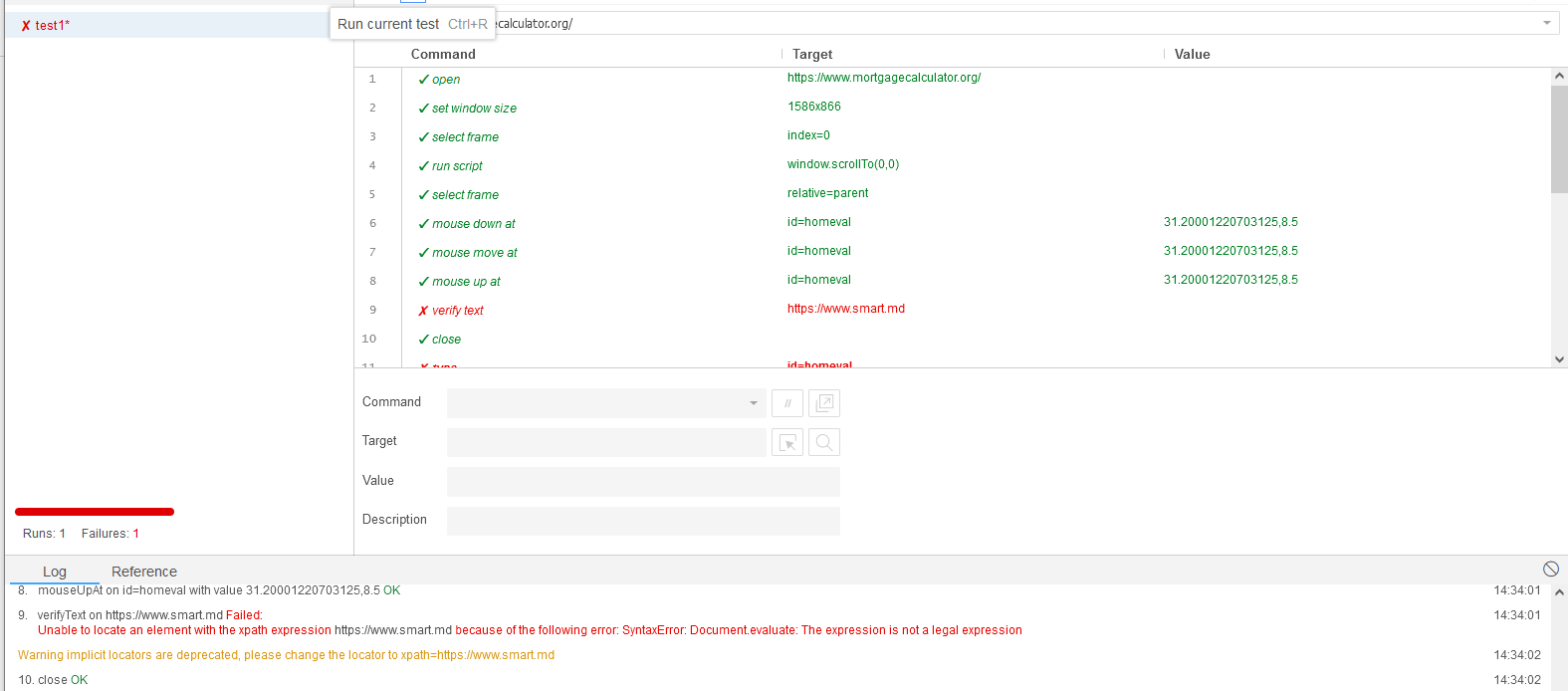
**Test Nr. 1**

Am folosit keyword-ul open care va deschide o pagina noua atunci când instrucțiunea data va fi apelată.



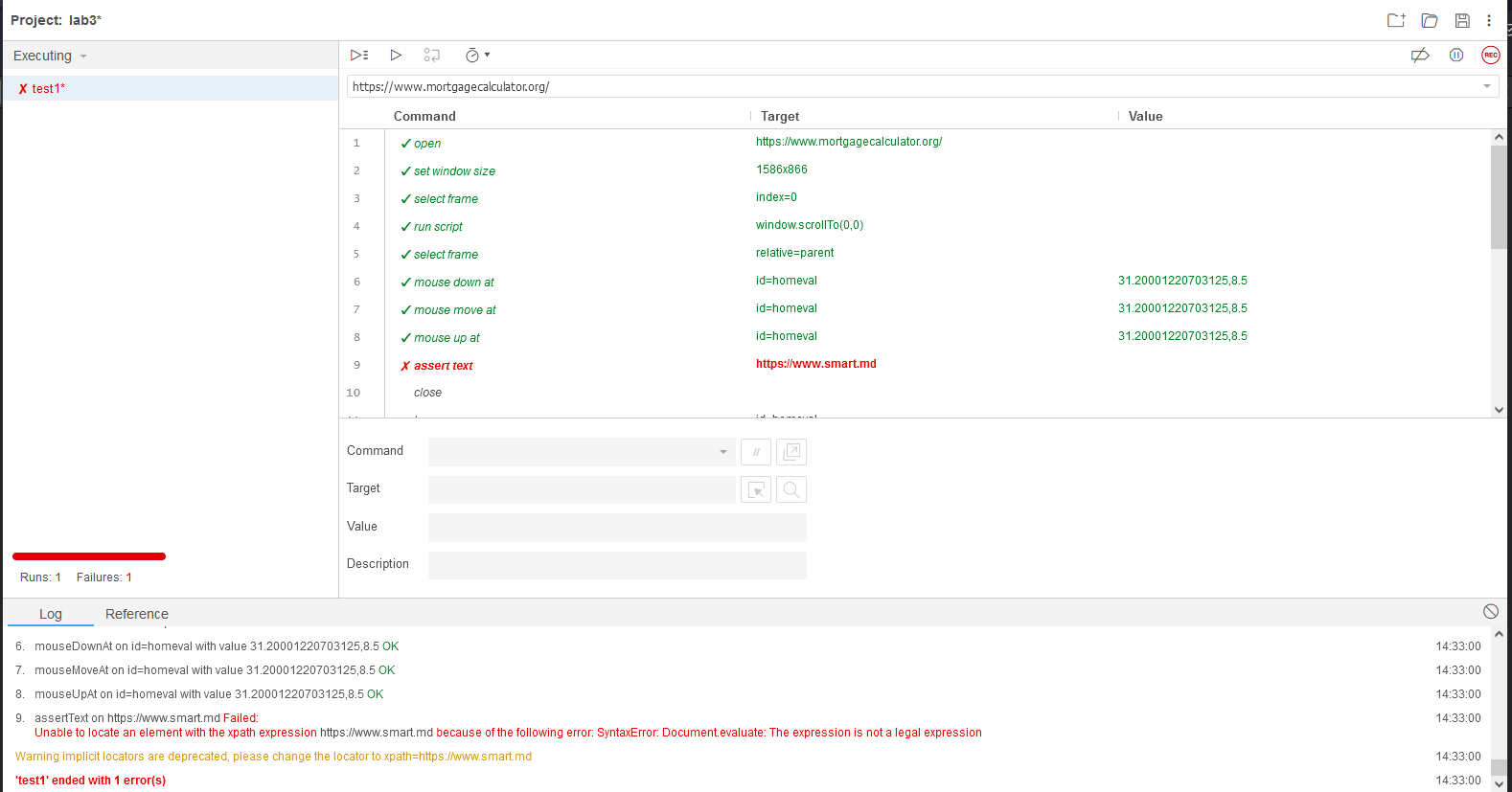
**Figura 1.1** Folosirea comenzii open

Comanda verify text e folosita pentru a verifica dacă elementul dată îndeplinește anumiți parametri, ei aici nefiind specificați noi primim o eroare, dar testele vor continua mai departe.



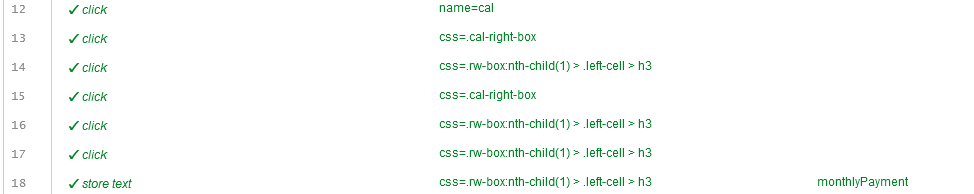
**Figura 1.2** Folosirea comenzii verify text

Assert text are același scop ca și verify, doar ca în cazul în care folosim acest keyword testul se va opri atunci când va avea eroare.



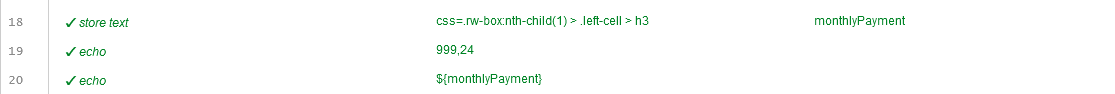
**Figura 1.3** Folosirea comenzii assert text

Cu ajutorul IDE selenium putem selecta store text pentru a salva o variabilă de pe o pagină web, astfel valoarea stocată în ea va putea folosită pe parcurs



**Figura 1.4** Folosirea uneltei store text

Cu ajutorul keywordului echo afișăm la ecran ce dorim, variabile, stringuri... .



**Figura 1.5** Afișarea valorilor la ecran

Mai jos se poate observa output-ul stringului introdus de noi, dar și variabila care a fost salvată cu intrumentul store text.

**Figura 1.6** Output-ul comenzii echo

Acesete teste pe care le efectuăm în IDE le vom putea exporta și sub forma unui limbaj de programare, eu am ales java, iar în el testele sunt create cu ajutorul librării Junit, care ne oferă posibilitatea la accesul anumitor setări de rulare a testelor și e specializat pe asta.

public class Test1Test {  
 private WebDriver driver;  
 private Map<String, Object> vars;  
 JavascriptExecutor js;  
  
 @Before  
 public void setUp() {  
 driver = new FirefoxDriver();  
 js = (JavascriptExecutor) driver;  
 vars = new HashMap<String, Object>();  
 }  
  
 @After  
 public void tearDown() {  
 driver.quit();  
 }  
  
 @Test  
 public void test1() {  
 driver.get("https://www.mortgagecalculator.org/");  
 driver.manage().window().setSize(new Dimension(1586, 866));  
 driver.switchTo().frame(0);  
 js.executeScript("window.scrollTo(0,0)");  
 driver.switchTo().defaultContent();  
 {  
 WebElement element = driver.findElement(By.id("homeval"));  
 Actions builder = new Actions(driver);  
 builder.moveToElement(element).clickAndHold().perform();  
 }  
 {  
 WebElement element = driver.findElement(By.id("homeval"));  
 Actions builder = new Actions(driver);  
 builder.moveToElement(element).perform();  
 }  
 {  
 WebElement element = driver.findElement(By.id("homeval"));  
 Actions builder = new Actions(driver);  
 builder.moveToElement(element).release().perform();  
 }  
 driver.findElement(By.id("homeval")).click();  
 driver.findElement(By.id("homeval")).sendKeys("250000");  
 driver.findElement(By.id("intrstsrate")).click();  
 driver.findElement(By.name("cal")).click();  
 driver.findElement(By.cssSelector(".cal-right-box")).click();  
 driver.findElement(By.cssSelector(".rw-box:nth-child(1) > .left-cell > h3")).click();  
 driver.findElement(By.cssSelector(".cal-right-box")).click();  
 driver.findElement(By.cssSelector(".rw-box:nth-child(1) > .left-cell > h3")).click();  
 driver.findElement(By.cssSelector(".rw-box:nth-child(1) > .left-cell > h3")).click();  
 vars.put("monthlyPayment", driver.findElement(By.cssSelector(".rw-box:nth-child(1) > .left-cell > h3")).getText());  
 System.out.println("999,24");  
 System.out.println(vars.get("monthlyPayment").toString());  
 }  
}

**Figura 1.7** Codul exportat in Java

class TestTest1():  
 def setup\_method(self, method):  
 self.driver = webdriver.Firefox()  
 self.vars = {}  
   
 def teardown\_method(self, method):  
 self.driver.quit()  
   
 def test\_test1(self):  
 self.driver.get("https://www.mortgagecalculator.org/")  
 self.driver.set\_window\_size(1586, 866)  
 self.driver.switch\_to.frame(0)  
 self.driver.execute\_script("window.scrollTo(0,0)")  
 self.driver.switch\_to.default\_content()  
 element = self.driver.find\_element(By.ID, "homeval")  
 actions = ActionChains(self.driver)  
 actions.move\_to\_element(element).click\_and\_hold().perform()  
 element = self.driver.find\_element(By.ID, "homeval")  
 actions = ActionChains(self.driver)  
 actions.move\_to\_element(element).perform()  
 element = self.driver.find\_element(By.ID, "homeval")  
 actions = ActionChains(self.driver)  
 actions.move\_to\_element(element).release().perform()  
 self.driver.find\_element(By.ID, "homeval").click()  
 self.driver.find\_element(By.ID, "homeval").send\_keys("250000")  
 self.driver.find\_element(By.ID, "intrstsrate").click()  
 self.driver.find\_element(By.NAME, "cal").click()  
 self.driver.find\_element(By.CSS\_SELECTOR, ".cal-right-box").click()  
 self.driver.find\_element(By.CSS\_SELECTOR, ".rw-box:nth-child(1) > .left-cell > h3").click()  
 self.driver.find\_element(By.CSS\_SELECTOR, ".cal-right-box").click()  
 self.driver.find\_element(By.CSS\_SELECTOR, ".rw-box:nth-child(1) > .left-cell > h3").click()  
 self.driver.find\_element(By.CSS\_SELECTOR, ".rw-box:nth-child(1) > .left-cell > h3").click()  
 self.vars["monthlyPayment"] = self.driver.find\_element(By.CSS\_SELECTOR, ".rw-box:nth-child(1) > .left-cell > h3").text  
 print("999,24")  
 print("{}".format(self.vars["monthlyPayment"]))

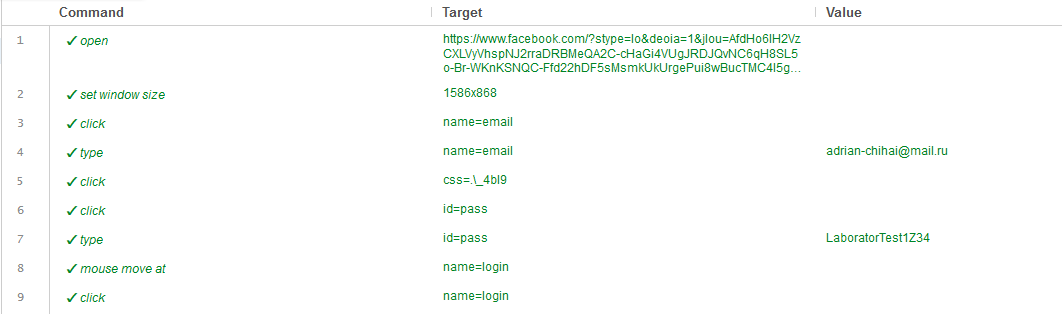
**Figura 1.8** Codul exportat în python

**Test Nr. 2**

În cadrul testului de mai jos se pot vedea pașii pentru conectarea pe contul de facebook.

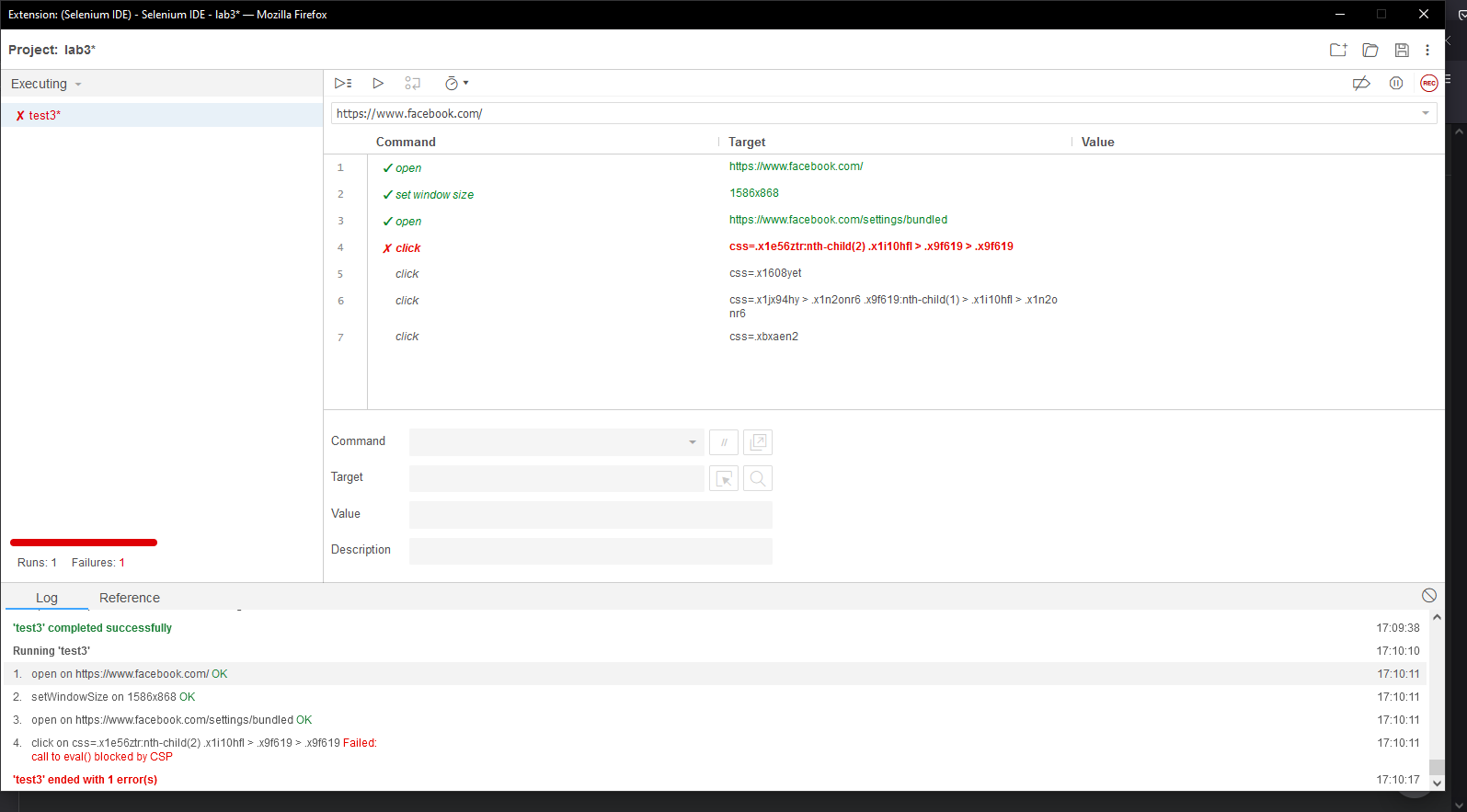


**Figura 2.1** Testul pentru logare pe facebook

Pașii de mai jos sunt la fel, doar ca target-ul este modificat.

**Figura 2.2** Testul pentru logare pe facebook cu target name modificat

În figura 2.3 am încercat să modific profilul de facebook, dar a fost fără succes, deoarece pe facebook securitatea nu ne permite.



**Figura 2.3** Testarea pentru modificarea profilului

**Concluzie**

În cadrul acestui laborator am folosit selenium pentru a testa mai amănunțit site-ul de la laboratul anterior. Am folosit noi unelte și aflat noi funcții, de exemplu open sau cum putem salva o valoare de pe site într-o variabilă. Am văzut și cum aceste teste pot fi exportate într-un limbaj de programare, observând cum aceste teste pot fi efectuate și cu ajutorul unui limbaj.

În ultimul test am folosit facebook ca site pe care urma să efectuez testele, am observat că ne-a permis să ne logăm, dar mai departe a fost imposibil de testat din cauza securității. Observând că dacă alegem un target, la următorul refresh al site-ului acesta îsi va schimba structura în css finder, astfel fiind imposibil de testat.