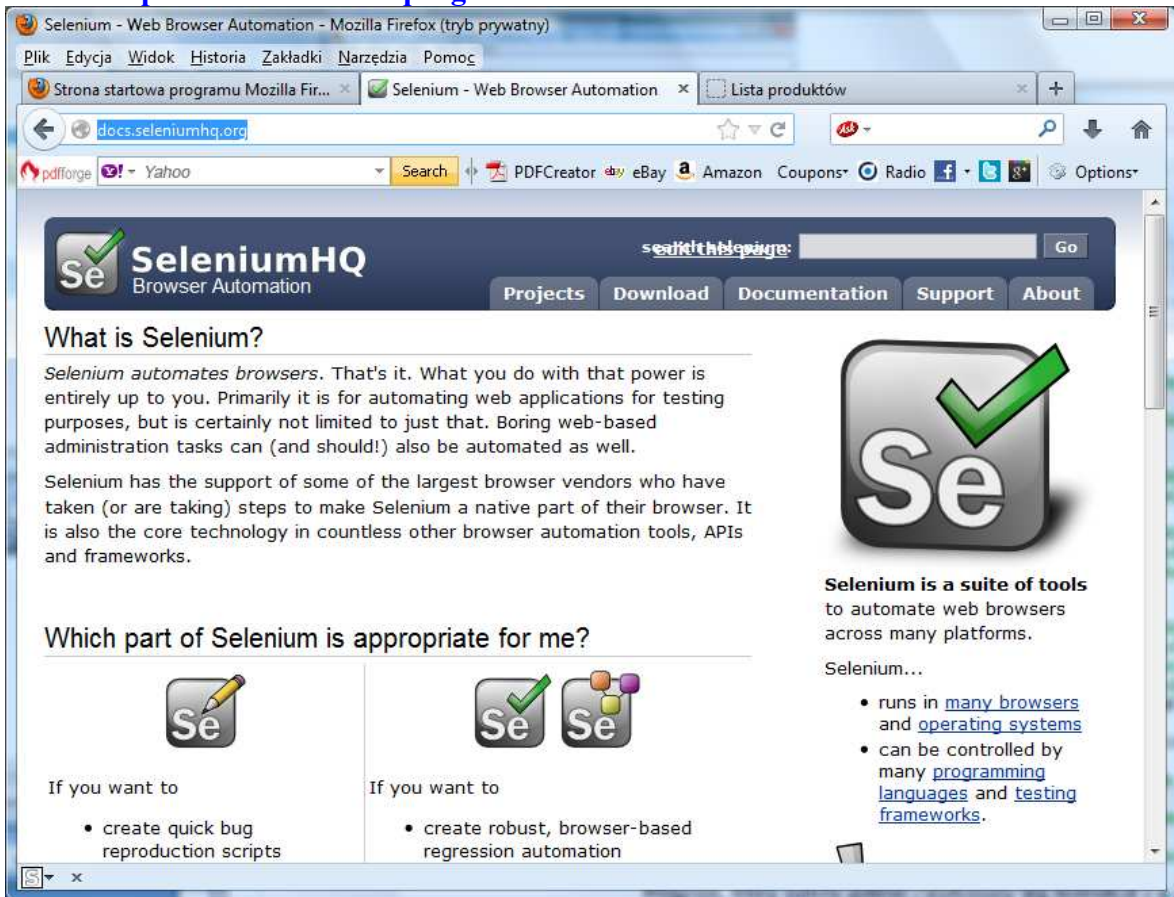
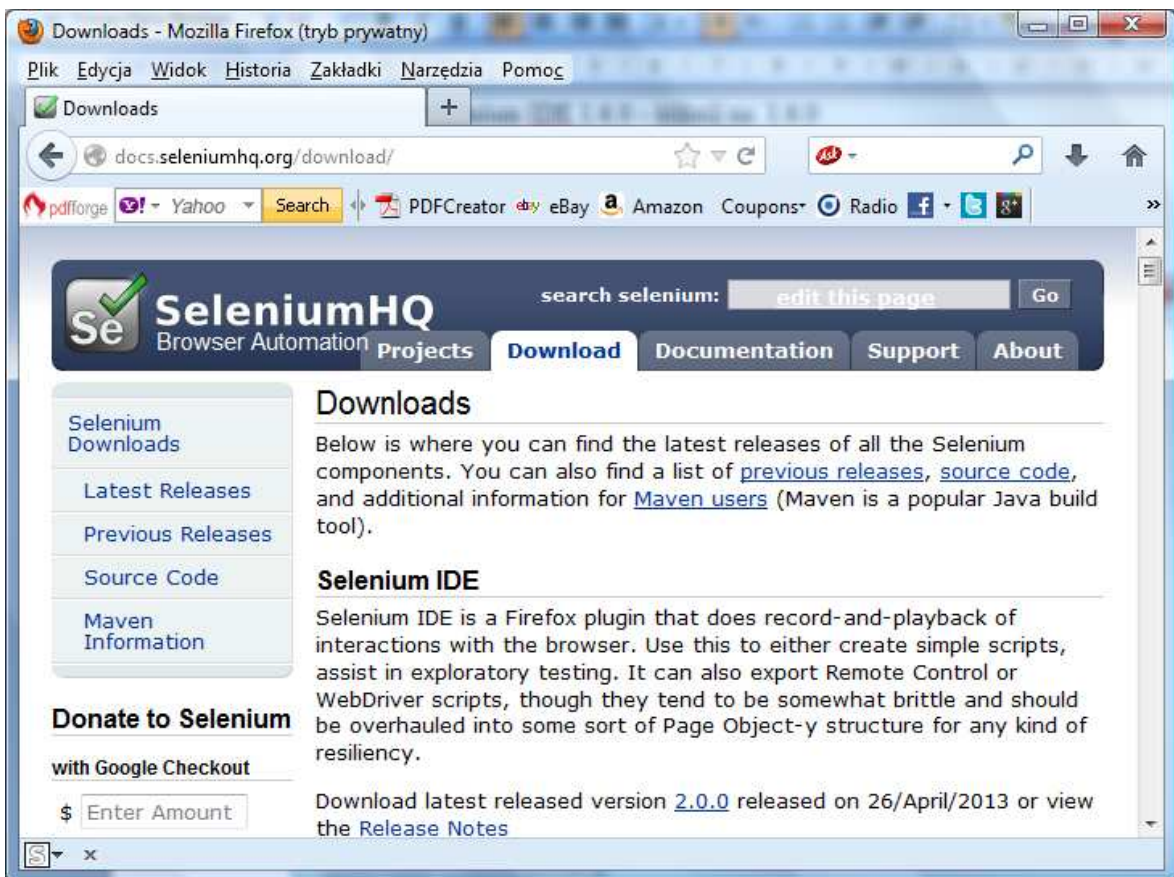


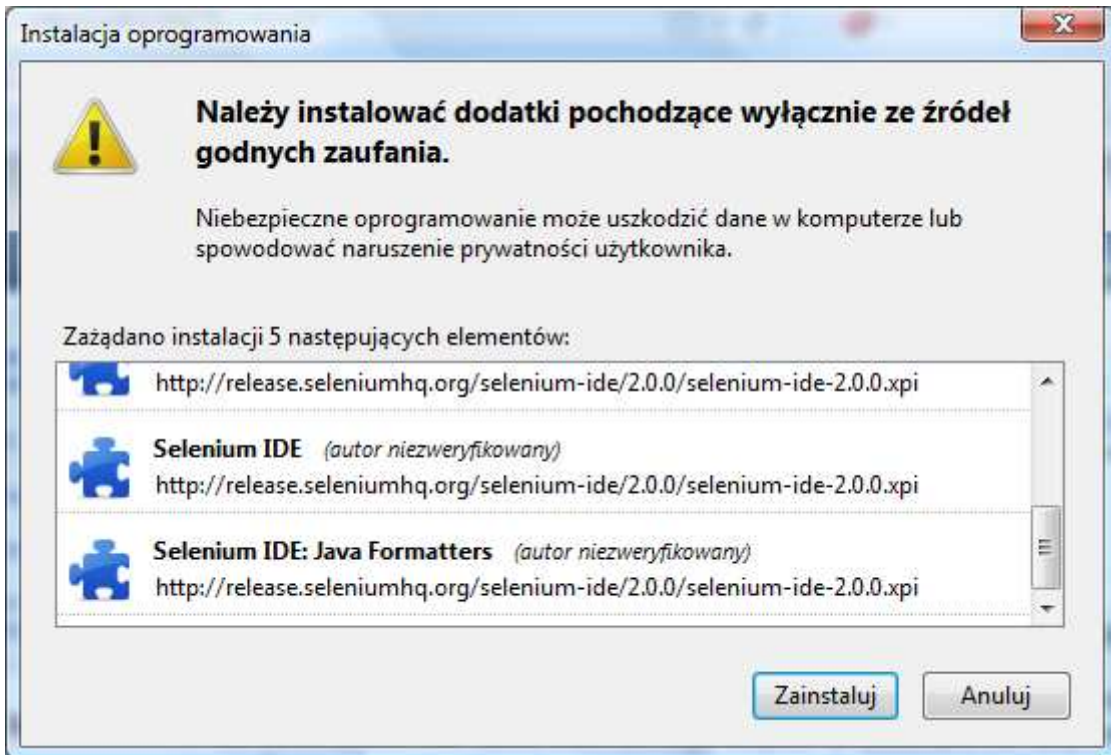
1. Po uruchomieniu przeglądarki Mozilla FireFox, należy uruchomić stronę <http://docs.seleniumhq.org>



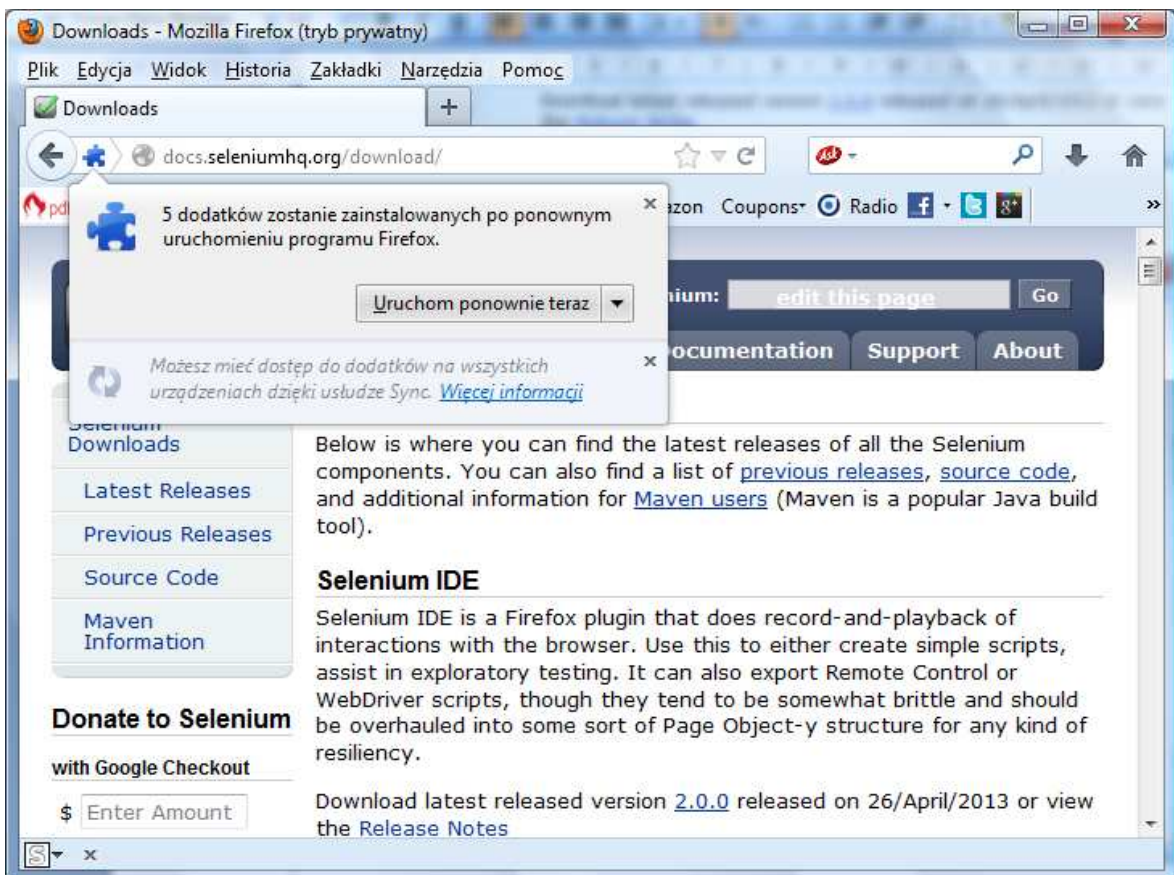
2. Po wybraniu zakładki **Download** w celu zainstalowania oprogramowania Selenium IDE należy kliknąć na link **2.0.0**



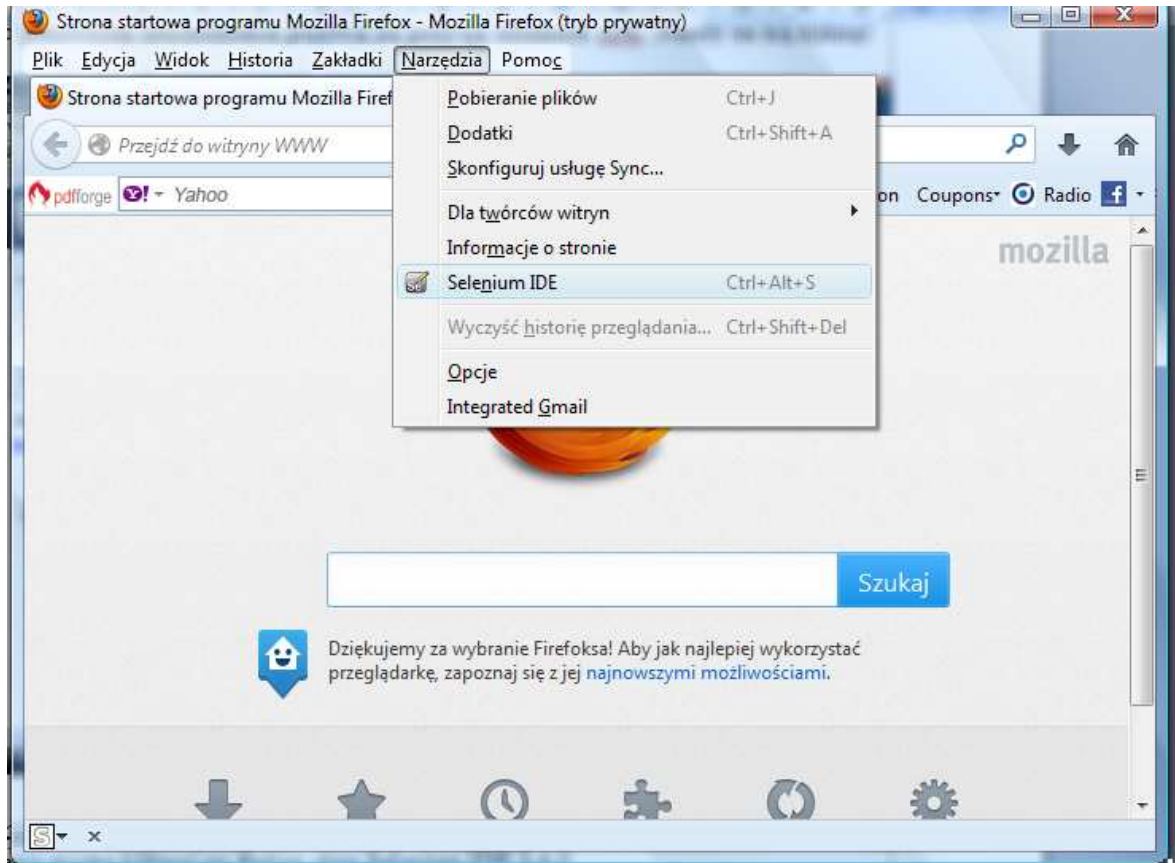
3. Po ukazaniu się formularza **Instalacja oprogramowania** należy kliknąć na przycisk **Zainstaluj**



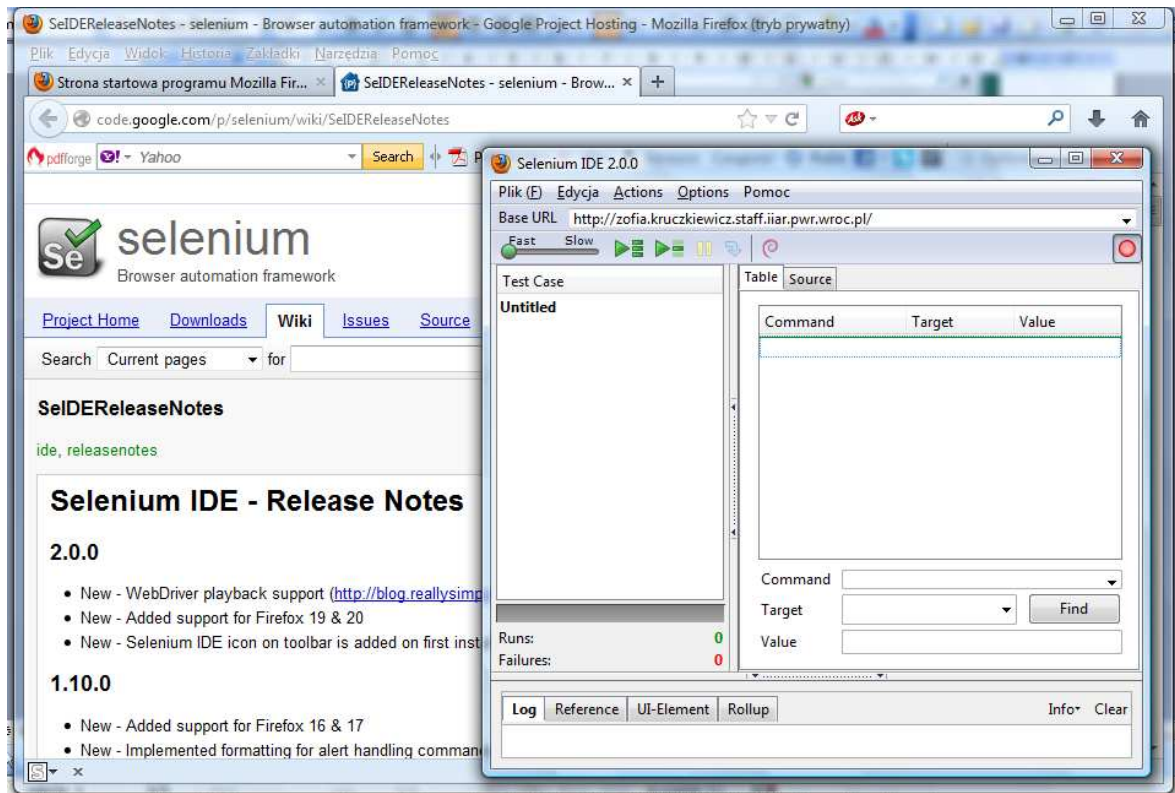
4. Po zainstalowaniu należy wybrać **Uruchom ponownie teraz**



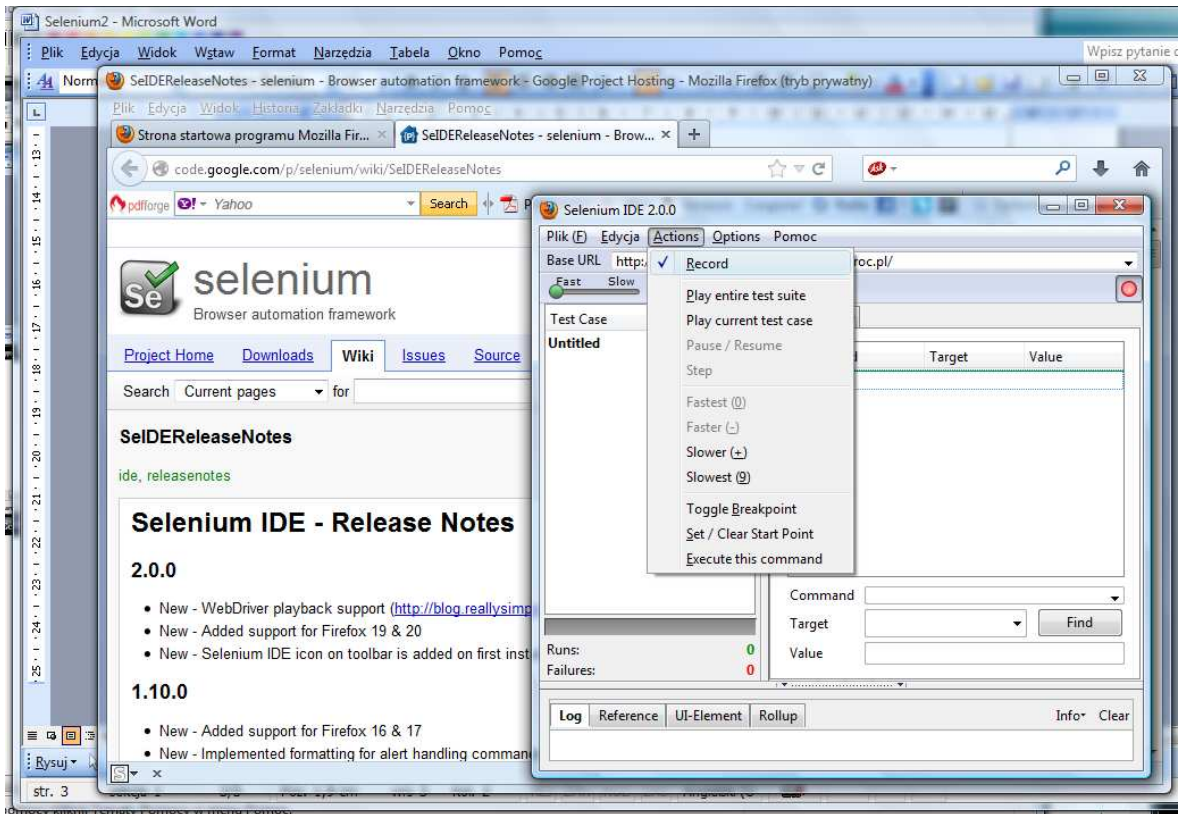
5. Po ponownym uruchomieniu pojawia się pozycja **Selenium IDE** w liście zakładki **Narzędzia**. W celu uruchomienia Selenium należy na kliknąć na pozycję **Selenium IDE**



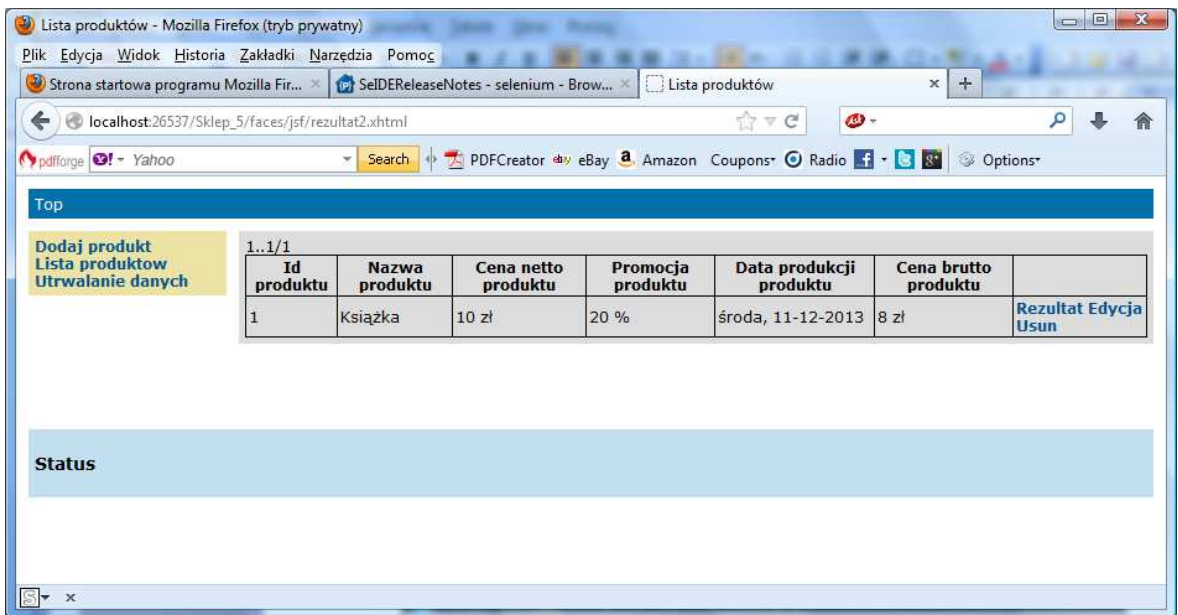
6. Teraz można kliknąć na pozycję **Action** okna **Selenium IDE 2.0.0**



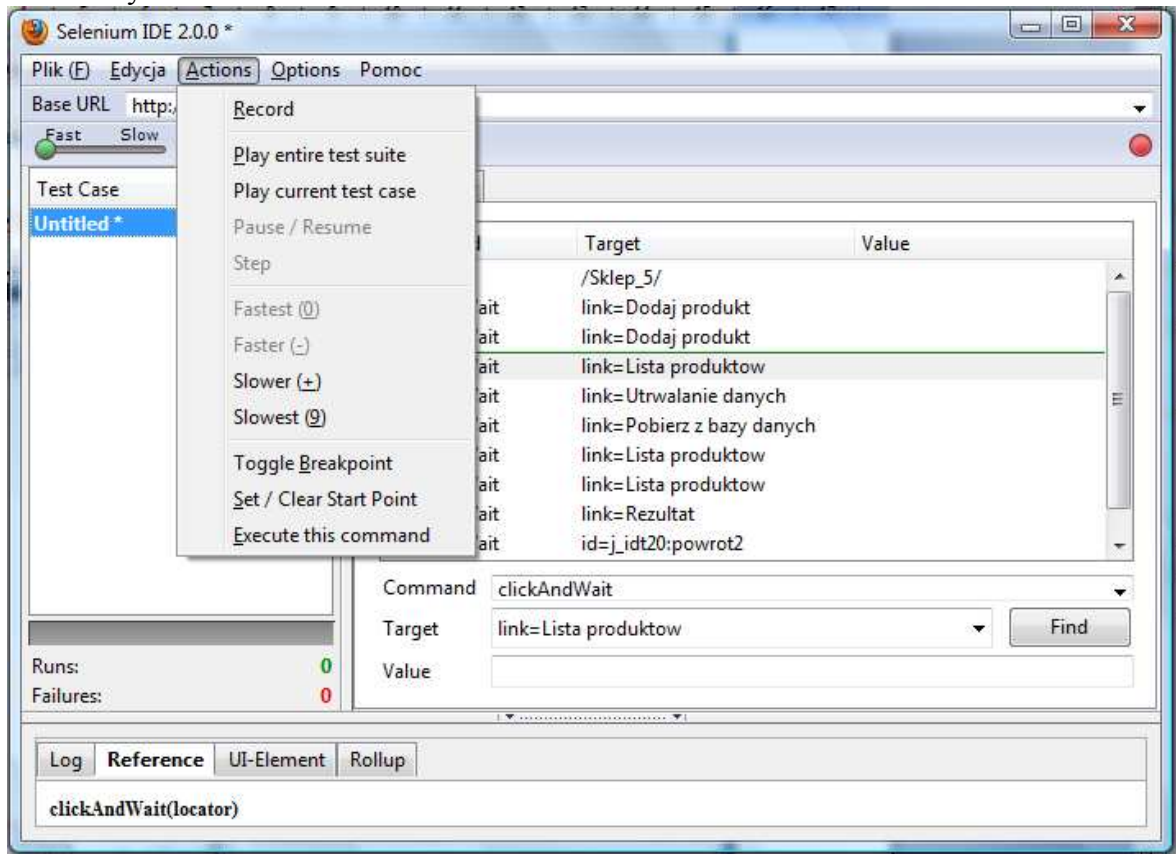
7. Następnie należy kliknąć na pozycję **Record** w celu nagrania testu



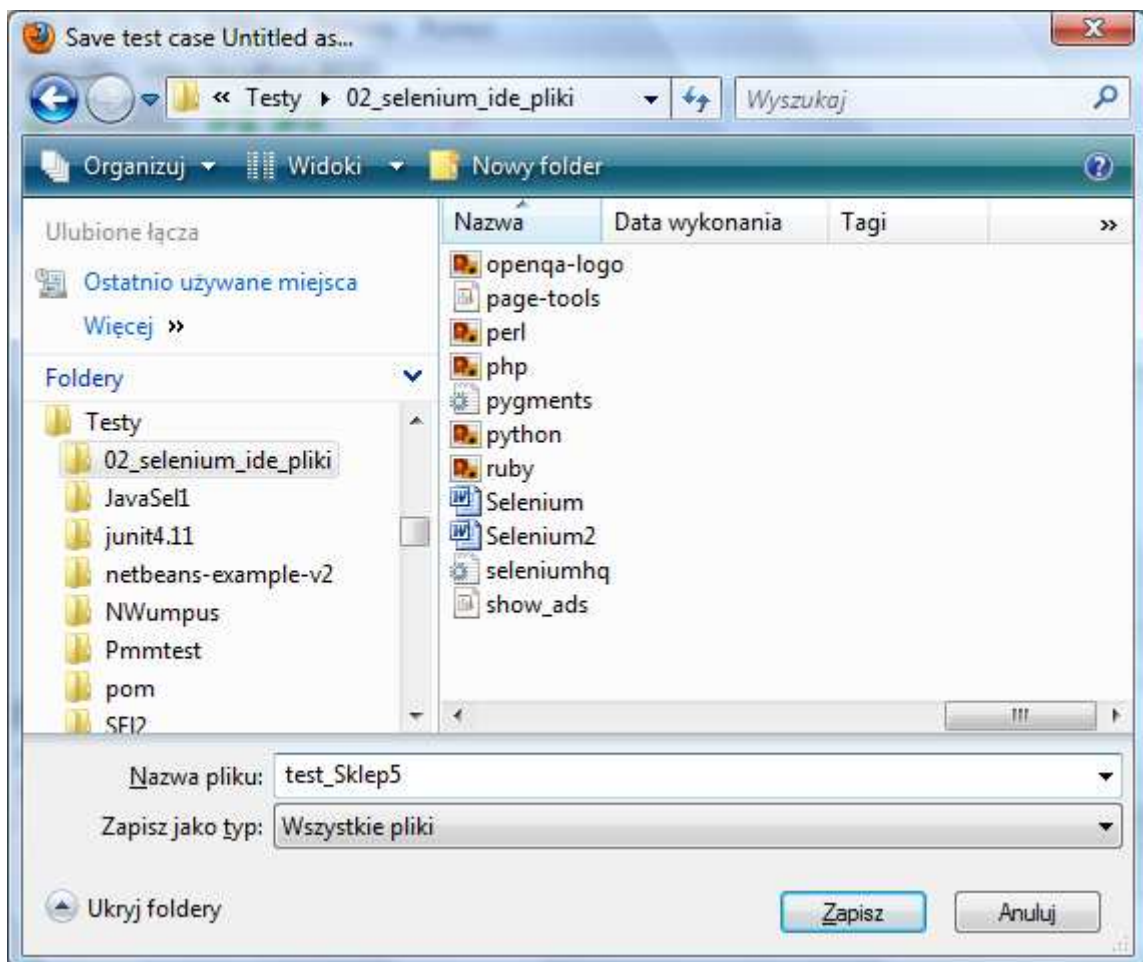
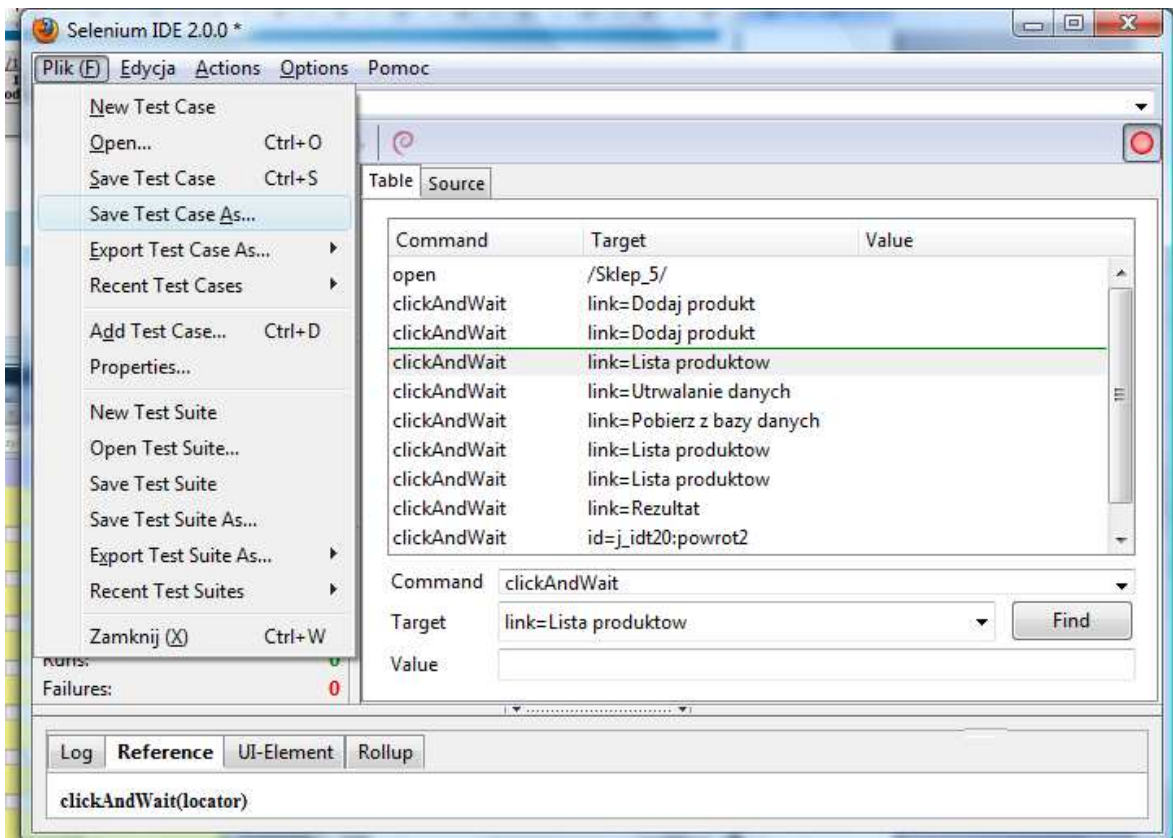
8. Po uruchomieniu testowanej aplikacji zostaną nagrane wszystkie czynności wykonywane przez aplikację



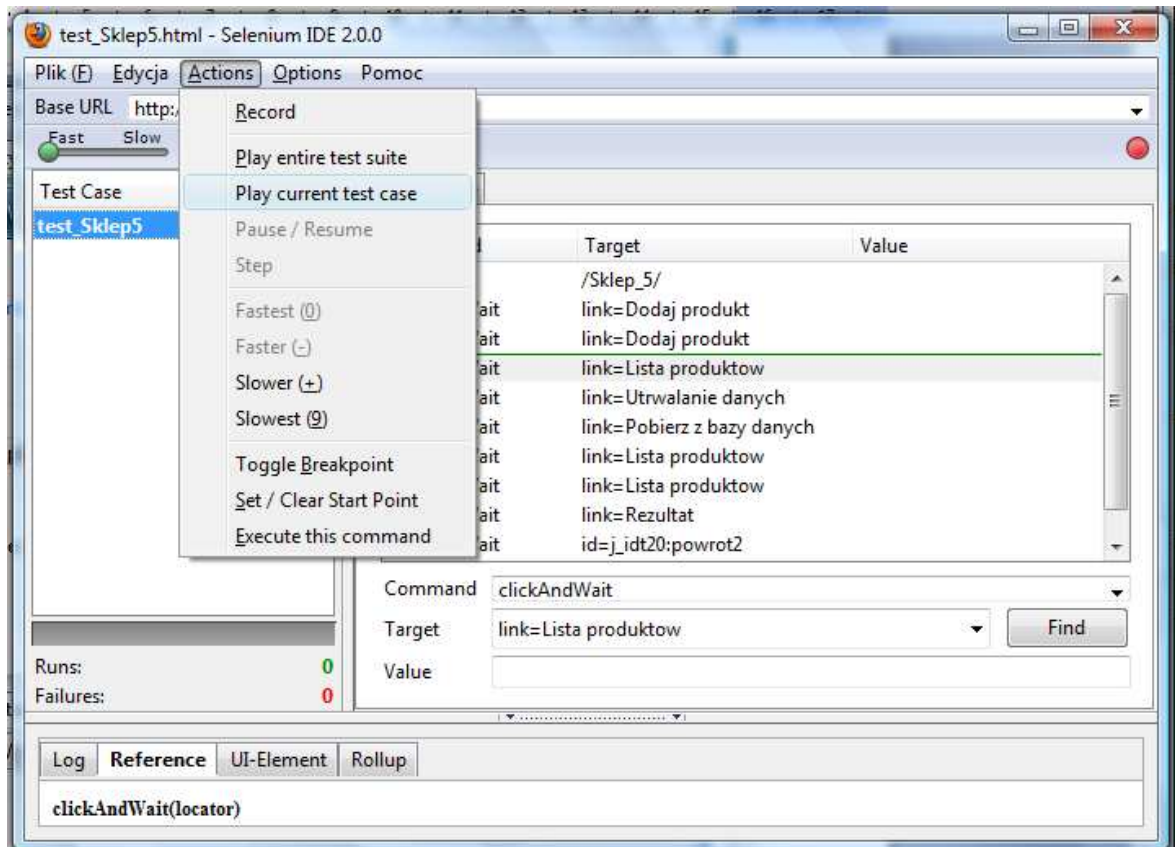
9. Po zakończeniu działania aplikacji należy wyłączyć Nagrywanie (**Record**) po wybraniu zakładki **Action**



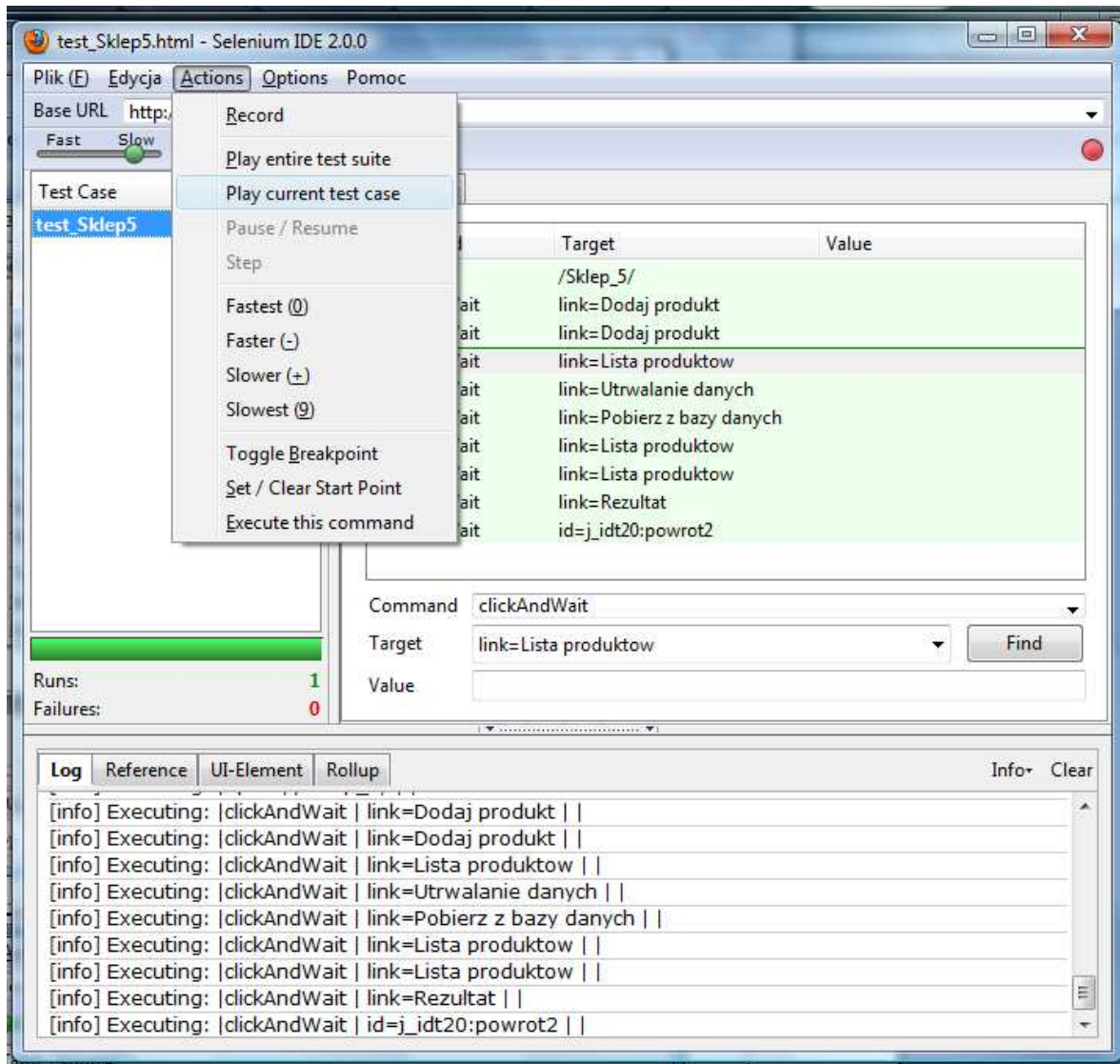
10. Teraz można zapisać test przez wybór **Plik(F)\Save Test Case As...**



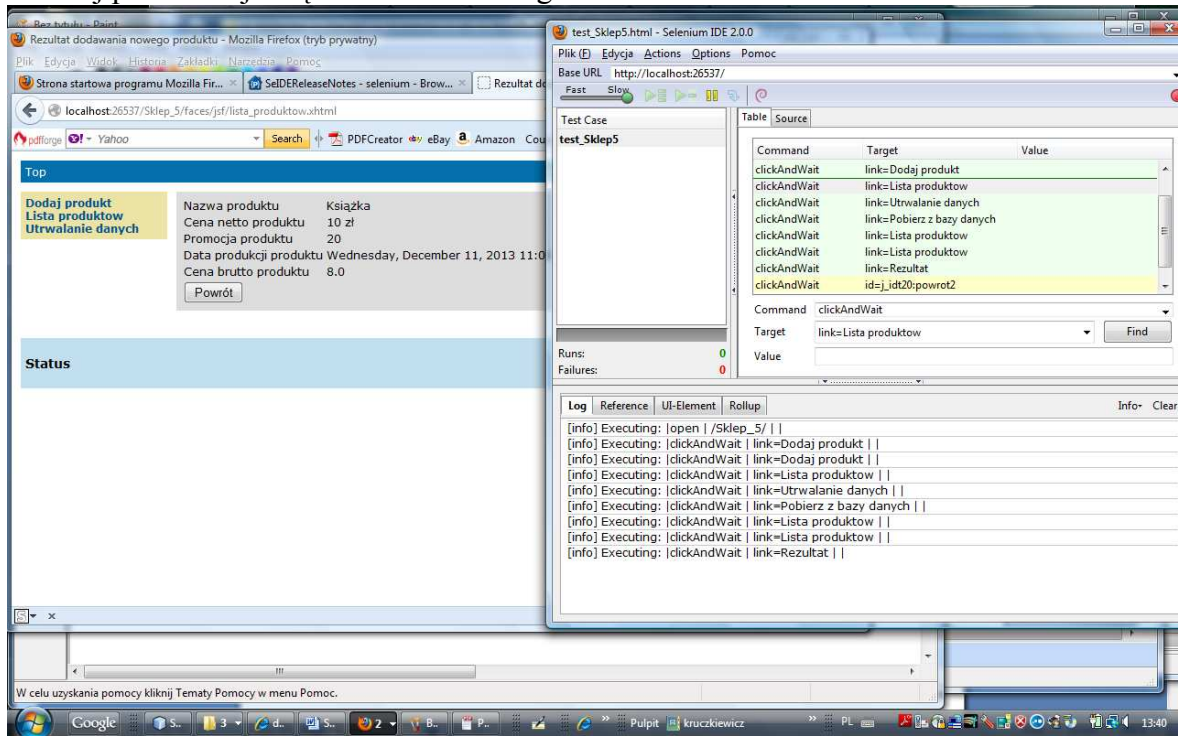
11. Teraz można odtworzyć test (**Action\Play current test case**) – aplikacja ponownie jest uruchamiana i działa automatycznie



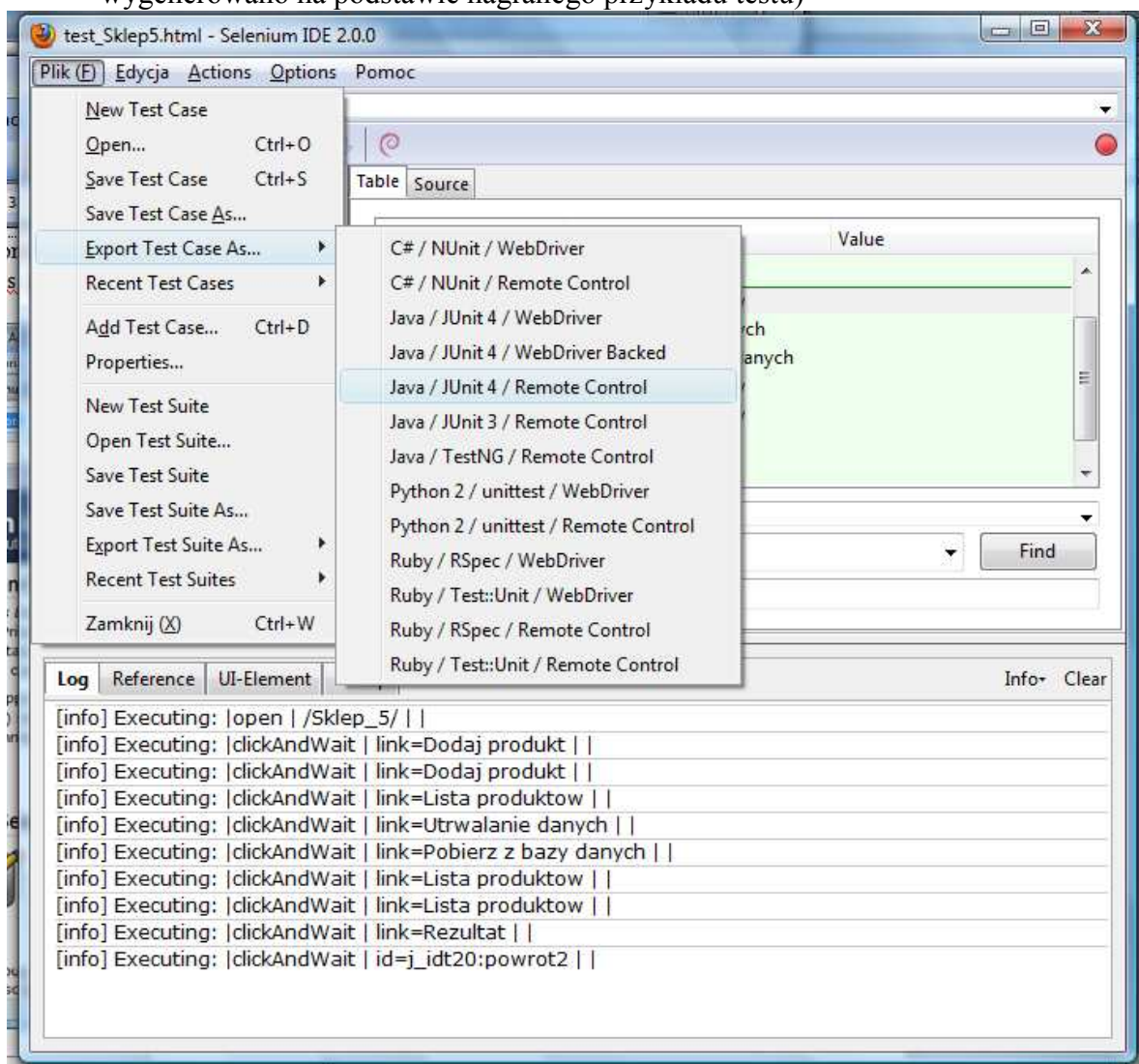
Można spowolnić działanie aplikacji (pozycja Slower lub za pomocą suwaka Fase-Slow na belce narzędziowej)



Poniżej pokazano jedną z faz odtwarzanego testu



12. Nagrany test można wyeksportować do środowiska **JUNIT4** lub **TestNG** w postaci pliku zawierającego definicję klasy testowej (podane dalej przykłady wygenerowano na podstawie nagranych przykładów testu)



12.1. Wygenerowana klasa realizująca test typu TestNG

```
package com.example.tests;

import com.thoughtworks.selenium.*;
import org.testng.annotations.*;
import static org.testng.Assert.*;
import java.util.regex.Pattern;

public class Sklep_5_TestNG_Remote extends SeleniumTestNgHelper {
    @Test
    public void testSklep_5_TestNG_Remote() throws Exception {
        selenium.open("/Sklep_5/");
        selenium.click("link=Dodaj produkt");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Dodaj produkt");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Lista produktow");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Utrwalanie danych");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Pobierz z bazy danych");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Lista produktow");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Lista produktow");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Rezultat");
        selenium.waitForPageToLoad("30000");
        selenium.click("id=j_idt20:powrot2");
        selenium.waitForPageToLoad("30000");
    }
}
```

12.2. Wygenerowana klasa realizująca test typu JUnit4/Remote Control

```
package com.example.tests;

import com.thoughtworks.selenium.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;
import java.util.regex.Pattern;

public class Sklep_5_Remote {
    private Selenium selenium;

    @Before
    public void setUp() throws Exception {
        selenium = new DefaultSelenium("localhost", 4444, "*chrome",
"http://localhost:26537/");
        selenium.start();
    }

    @Test
    public void testSklep_5_Remote() throws Exception {
        selenium.open("/Sklep_5/");
        selenium.click("link=Dodaj produkt");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Dodaj produkt");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Lista produktow");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Utrwalanie danych");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Pobierz z bazy danych");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Lista produktow");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Lista produktow");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Rezultat");
        selenium.waitForPageToLoad("30000");
        selenium.click("id=j_idt20:powrot2");
        selenium.waitForPageToLoad("30000");
    }

    @After
    public void tearDown() throws Exception {
        selenium.stop();
    }
}
```


12.3. Wygenerowana klasa realizująca test typu JUnit4/WebDriver Backed

```
package com.example.tests;

import com.thoughtworks.selenium.Selenium;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebDriverBackedSelenium;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;
import java.util.regex.Pattern;
import static org.apache.commons.lang3.StringUtils.join;

public class Sklep_5_WebDriverBack {
    private Selenium selenium;

    @Before
    public void setUp() throws Exception {
        WebDriver driver = new FirefoxDriver();
        String baseUrl = "http://localhost:26537/";
        selenium = new WebDriverBackedSelenium(driver, baseUrl);
    }

    @Test
    public void testSklep_5_WebDriverBack() throws Exception {
        selenium.open("/Sklep_5/");
        selenium.click("link=Dodaj produkt");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Dodaj produkt");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Lista produktow");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Utrwalanie danych");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Pobierz z bazy danych");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Lista produktow");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Lista produktow");
        selenium.waitForPageToLoad("30000");
        selenium.click("link=Rezultat");
        selenium.waitForPageToLoad("30000");
        selenium.click("id=j_idt20:powrot2");
        selenium.waitForPageToLoad("30000");
    }

    @After
    public void tearDown() throws Exception {
        selenium.stop();
    }
}
```

12.4. Wygenerowana klasa realizująca test typu JUnit4/WebDriver

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class Sklep5WebDriver {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://localhost:26537/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testSklep5WebDriver() throws Exception {
        driver.get(baseUrl + "/Sklep_5/");
        driver.findElement(By.linkText("Dodaj produkt")).click();
        driver.findElement(By.linkText("Dodaj produkt")).click();
        driver.findElement(By.linkText("Lista produktow")).click();
        driver.findElement(By.linkText("Utrwalanie danych")).click();
        driver.findElement(By.linkText("Pobierz z bazy danych")).click();
        driver.findElement(By.linkText("Lista produktow")).click();
        driver.findElement(By.linkText("Lista produktow")).click();
        driver.findElement(By.linkText("Rezultat")).click();
        driver.findElement(By.id("j_idt20:powrot2")).click();
    }

    @After
    public void tearDown() throws Exception {
        driver.quit();
        String verificationErrorString = verificationErrors.toString();
        if (!"".equals(verificationErrorString)) {
            fail(verificationErrorString);
        }
    }

    private boolean isElementPresent(By by) {
        try {
            driver.findElement(by);
            return true;
        } catch (NoSuchElementException e) {
            return false;
        }
    }

    private boolean isAlertPresent() {
        try {
            driver.switchTo().alert();
            return true;
        } catch (NoAlertPresentException e) {
            return false;
        }
    }
}
```

```
private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
```