

```
1 package com.formation;
2
3 import java.nio.file.Path;
4 import java.nio.file.Paths;
5
6 import org.openqa.selenium.By;
7 import org.openqa.selenium.WebDriver;
8 import org.openqa.selenium.WebElement;
9
10 public class AgeCalculatorPage {
11     //WebElements
12     private WebElement dayOfBirth = null;
13     private WebElement monthOfBirth = null;
14     private WebElement yearOfBirth = null;
15     private WebElement age = null;
16     private WebElement zodiacSign = null;
17     private WebElement calculate = null;
18
19     ///WebDriver
20     private WebDriver driver;
21     Path sampleFile = Paths.get("pages/exercise_6_1.html");
22
23     private String url = sampleFile.toUri().toString();
24
25     //Class Constructor
26     public AgeCalculatorPage(WebDriver webDriver) {
27         driver = webDriver;
28     }
29
30     //Methods to open and close the WebDriver
31     public void open() {
32         this.driver.get(url);
33     }
34     public void close() {
35         this.driver.quit();
36     }
37
38     //Method to execute the test
39     public void calculate(String day, String month, String year) {
40         getDayOfBirth().sendKeys(day);
41         getMonthOfBirth().sendKeys(month);
42         getYearOfBirth().sendKeys(year);
43         getCalculate().click();
44     }
45
46     //Methods to read values from required WebElements
47     public String getAge() {
48         age = driver.findElement(By.id("age"));
49         return age.getText();
50     }
51
52     public String getZodiacSign() {
53         zodiacSign = driver.findElement(By.id("zodiacSign"));
54         return zodiacSign.getText();
55     }
```

```

56
57     public WebElement getDayOfBirth() {
58         dayOfBirth = driver.findElement(By.id("dayOfBirth"));
59         return dayOfBirth;
60     }
61
62     public WebElement getMonthOfBirth() {
63         monthOfBirth = driver.findElement(By.id("monthOfBirth"));
64         return monthOfBirth;
65     }
66
67     public WebElement getYearOfBirth() {
68         yearOfBirth = driver.findElement(By.id("yearOfBirth"));
69         return yearOfBirth;
70     }
71
72     public WebElement getCalculate() {
73         calculate = driver.findElement(By.id("calculate"));
74         return calculate;
75     }
76 }

```

Création de la classe Java AgeCalculatorScript

Créez une classe **AgeCalculatorScript** dans le package **com.formation**

```

1 package com.formation.script;
2
3
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeDriver;
6
7 import com.formation.AgeCalculatorPage;
8
9 public class AgeCalculatorScript {
10
11     public static void main(String[] args) throws Exception {
12         checkAgeCalculator();
13     }

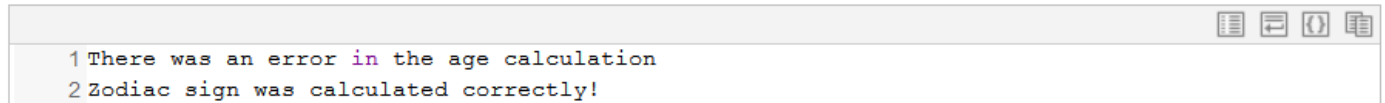
```

```

14
15 private static void checkAgeCalculator() throws Exception {
16     WebDriver driver = new ChromeDriver();
17     // Create an instance of AgeCalculatorPage class and open it
18     AgeCalculatorPage ageCalculatorPage = new AgeCalculatorPage(driver);
19     ageCalculatorPage.open();
20
21     // Start the test by means of the calculate method
22     ageCalculatorPage.calculate("11", "February", "1982");
23
24     // Verify results
25     if (ageCalculatorPage.getAge().equals("36")) {
26         System.out.println("Age was calculated correctly!");
27     } else {
28         System.out.println("There was an error in the age calculation");
29     }
30
31     if (ageCalculatorPage.getZodiacSign().equals("Aquarius")) {
32         System.out.println("Zodiac sign was calculated correctly!");
33     } else {
34         System.out.println("There was an error in the zodiac sign calculation");
35     }
36
37     ageCalculatorPage.close();
38 }
39 }

```

Nous obtenons la sortie suivante dans la console



```

1 There was an error in the age calculation
2 Zodiac sign was calculated correctly!

```