

Testarea automată utilizând LeanFT

1. Scopul lucrării

Obiectivele de învățare ale acestei sesiuni de laborator sunt:

- Înțelegerea și utilizarea instrumentului de testare automată LeanFT oferit de către Hewlett Packard Enterprise (HPE)
- Scrierea și aplicarea metodelor de testare automată utilizând LeanFT
- Testarea automată a aplicației CalcMVC (din laboratorul GUI-I) utilizând LeanFT

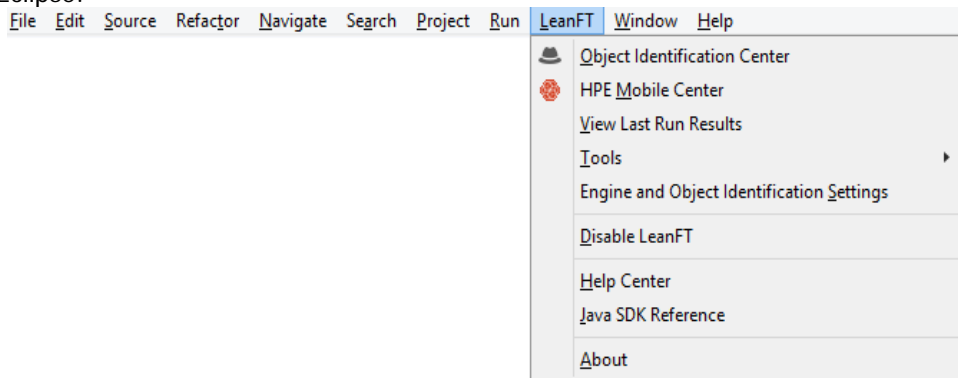
2. Introducere în LeanFT

HPE Lean Functional Testing (LeanFT) este o soluție puternică de testare funcțională construită special pentru testarea continuă și integrarea continuă. LeanFT este ideal pentru testarea funcționalităților aplicațiilor care au interfață grafică întrucât permite automatizarea interacțiunilor cu componentele grafice ale acestora. Astfel, testele scrise vor conține astfel de interacțiuni cu interfața grafică, vor furniza date de test și vor verifica dacă rezultatul așteptat în urma execuției testelor coincide cu cel așteptat.

Instalarea LeanFT pentru mediul de dezvoltare Eclipse presupune:

- descărcarea kit-ului de instalare LeanFT în variantă Trial (trebuie să vă creați în prealabil un cont): <https://saas.hpe.com/en-us/software/leanft>
- descărcarea kit-ului de instalare pentru Node.js, instrument gratis necesar pentru a putea lucra cu LeanFT: <https://nodejs.org/en/>
- se va instala prima dată Node.js iar apoi LeanFT (varianta Trial a acestuia vă permite să îl folosiți 60 de zile pe sistemul pe care l-ați instalat)
 - la instalarea LeanFT va trebui să specificați calea către mediul de dezvoltare Eclipse pe care îl aveți instalat

După ce ați instalat LeanFT, veți observa prezența lui în meniul principal (orizontal) al mediului de dezvoltare Eclipse:



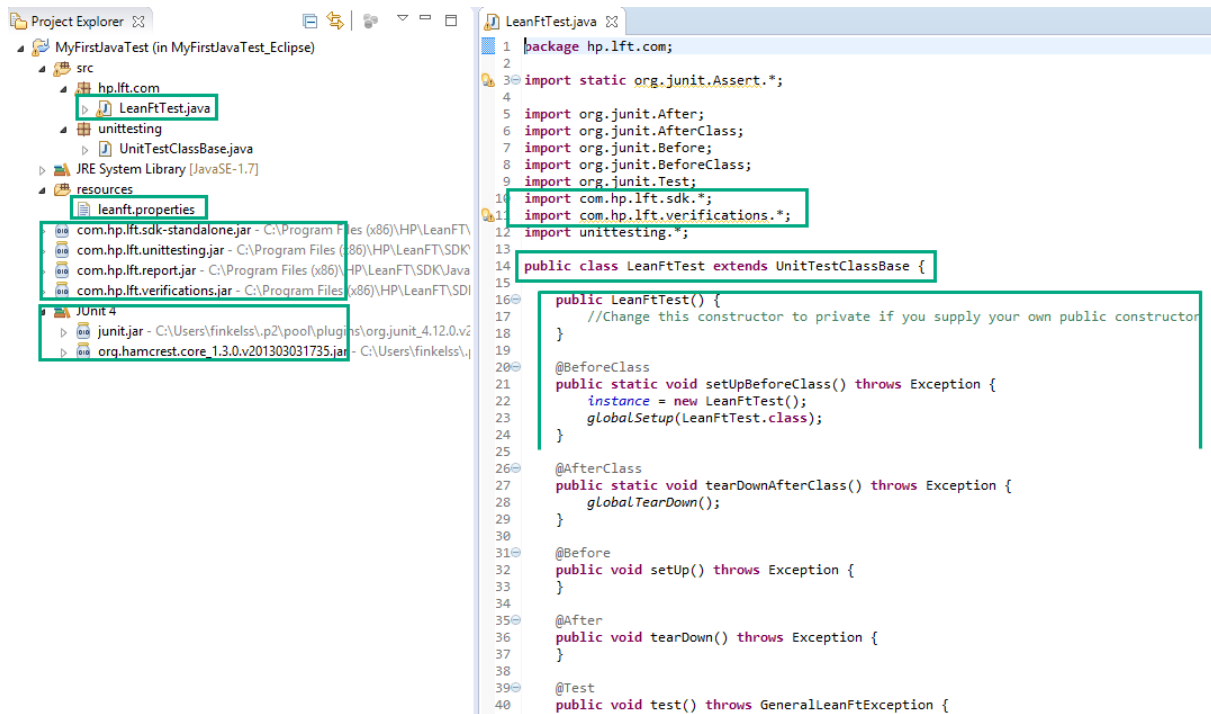
Mai multe despre LeanFT puteți studia, accesând "Help Center" din meniul LeanFT din figura de mai sus, sau direct http://leanft-help.saas.hpe.com/en/12.54/HelpCenter/Content/Resources/TopNav/TopNav_Home.htm

SDK-ul utilizat de LeanFT poate fi găsit accesând "Java SDK Reference" din meniul LeanFT din figura de mai sus.

3. Testarea aplicațiilor utilizând LeanFT

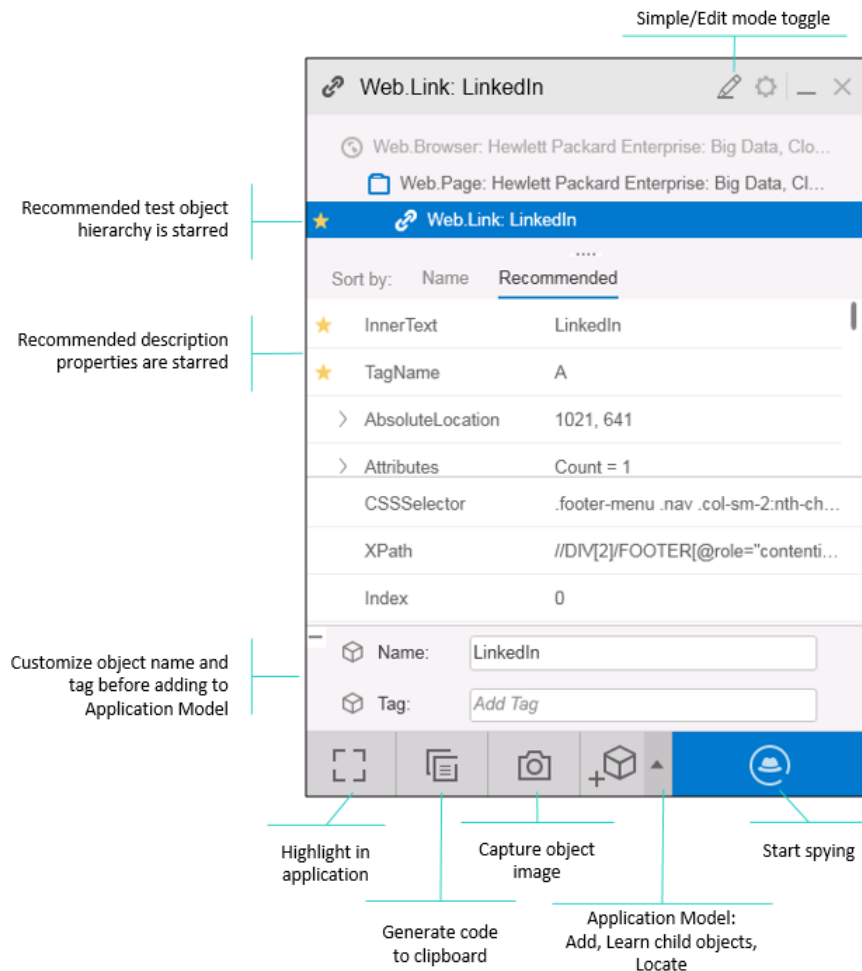
Scenariul de bază pentru testarea aplicațiilor utilizând LeanFT presupune următorii pași:

- Crearea proiectului de test
 - Se face din Eclipse direct din meniul File->New->Other->LeanFT->LeanFT JUnit Project



➤ Definirea obiectelor de test

- Cel mai ușor se face prin accesarea "Object Identification Center" din meniul LeanFT după ce în prealabil se deschide aplicația care urmează a fi testată



- „Spionarea” asupra componentelor interfeței grafice a aplicației ce urmează a fi testată se face prin apăsarea butonului “Start spying” din fereastra “Object Identification Center”
 - Descrierea componentelor din interfața grafică a aplicației se face prin selectarea lor prin click cu mouse-ul. În acest moment se poate genera codul care descrie componenta respectivă prin apăsarea butonului “Generate code to clipboard” din fereastra “Object Identification Center”. Codul generat este folosit în clasele de test pentru identificarea componentelor grafice
- Scrierea testelor
- Se va face utilizând SDK-ul LeanFT existent. Testele se pot scrie în clasa `LeanFtTest` care a fost creată automat în momentul creării proiectului LeanFT de test
- Execuția testelor
- Se va rula în Eclipse proiectul ca și un JUnit Test. În acest moment are loc interacțiunea automată cu interfața grafică a aplicației care se testează, furnizarea datelor de test și apoi execuția automată a tuturor testelor scrise (asemenea testelor simple scrise cu JUnit). Trebuie să așteptați până la terminarea tuturor testelor.
- Analiza rezultatelor
- După terminarea tuturor testelor mediul de dezvoltare Eclipse, prin JUnit, o să afișeze că toate testele au fost completate cu succes (dacă acestea nu au fost întrerupte din rulare lor). Acest lucru înseamnă că toate testele au fost executate, dar nu înseamnă că ele au trecut cu succes.
 - Pentru a vedea raportul terminării fiecărui test (cu succes sau cu eșec) trebuie să accesați din meniul LeanFT intrarea “View Last Run Results”

Tutorialul complet LeanFT cu exerciții pentru crearea și rulare automată de scenarii de test poate fi studiat accesând:

http://leanft-help.saas.hpe.com/en/12.54/HelpCenter/Content/HowTo/HowTo_End2End.htm

4. Testarea aplicației CalcMVC utilizând LeanFT

Descărcați proiectul **CalcMVC_LeanFT** pus la dispoziție. Acesta este un proiect de testare automată utilizând LeanFT a aplicației simple CalcMVC prezentată în laboratorul GUI-I.

Studiați structura și conținutul fișierului **LeanFtTest.java**. Acestea sunt asemănătoare claselor de test scrise cu JUnit. Clasa conține trei metode de test:

- `TestfromXlsx()`
- Testează funcționalitatea calculatorului prin introducerea succesivă de valori în căsuța Input și apăsarea ulterioară a butonului Multiply
 - Se apasă inițial butonul Clear al aplicației cu scopul resetării câmpului Total la 1
 - Se citește din fișierul **input_expected.xlsx** o serie de patru perechi de valori (*valoare_input* – coloana A, *valoare_asteptata* – coloana B) aflate fiecare pe câte o linie separată în fișierul Excel

	A	B
1	2	2
2	3	6
3	2	13
4	2	24

- După citirea fiecărei perechi se introduce automat *valoare_input* în căsuța Input a aplicației și se apasă automat butonul Multiply. Se citește automat de pe interfața grafică valoarea câmpului Total și are loc testarea egalității valorii citite cu *valoare_asteptata*

- Observație: între introducerea, pe rând, a celor patru valori în casuța Input și apăsarea butonului Multiply nu se apasă butonul Clear, deci câmpul Total nu este resetat astfel că acesta devine prim operand în următoarea operație de înmulțire!
- InvalidValuesTestsforString()
 - Testează dacă validarea textului introdus în căsuța Input funcționează: textul introdus trebuie să reprezinte un număr întreg, altfel trebuie să apară un mesaj de eroare
 - Metoda testează că acel mesaj de eroare apare. Deci testele vor fi terminate cu succes dacă în căsuța Input se introduc valori care nu sunt numere întregi
 - Se testează prin introducerea pe rând a patru valori String: "A", "2.0", "\$", "2"
- CheckClearButton()
 - Testează dacă prin apăsarea butonului Clear, câmpul Total este resetat la valoarea 1
 - Se execută o operație de înmulțire și se apasă apoi automat butonul Clear
 - Se testează prin citire automată dacă valoarea câmpului Total este identică cu 1

Descrierea elementelor de interfața grafică sunt inserate în același fișier cu cod sursă. Acestea au fost generate semi-automat utilizând „spionare” pe interfața grafică cu "Object Identification Center". Metodele care descriu aceste entități sunt:

- inputField() – descrie căsuța de Input prin care se citesc numerele
- multiplyButton() – descrie butonul de Multiply
- totalField() – descrie câmpul Total în care se afișează rezultatul înmulțirii
- clearButton() – descrie butonul Clear
- errorMessage() – descrie fereastra mesajului de eroare care este afișată atunci când valoarea introdusă nu reprezintă un număr întreg
- errorMessageText() – descrie textul mesajului de eroare
- errorMessageOkButton() – descrie butonul OK de pe fereastra mesajului de eroare
- simpleCalcWindow() – descrie fereastra principală a aplicației CalcMVC

Scheletul clasei LeanFtTest este asemănător cu cel al unei clase de test JUnit :

```
public class LeanFtTest extends UnitTestClassBase {

    ...

    public LeanFtTest() {...}

    @BeforeClass
    public static void setUpBeforeClass() throws Exception {...}

    @AfterClass
    public static void tearDownAfterClass() throws Exception {...}

    @Before
    public void setUp() throws Exception {...}

    @After
    public void tearDown() throws Exception {...}

    @Test
    public void TestfromXlsx() throws Exception {...}

    @Test
    public void InvalidValuesTestsforString() throws Exception {...}

    @Test
    public void CheckClearButton() throws Exception {...}

    public void InvalidTesting(String valueforTest) throws Exception {...}
}
```

```

//declare Simple CalcMVC Objects

//input Field
public Editor inputField() throws Exception {...}

//multiply Button
public Button multiplyButton() throws Exception {...}

//total Field
public Editor totalField() throws Exception {...}

//clear Button
public Button clearButton() throws Exception {...}

//error Message dialog
public Dialog errorMessage() throws Exception {...}

//error Message Text
public Label errorMessageText() throws Exception {...}

//error Message OK Button
public Button errorMessageOkButton() throws Exception {...}

// main Window
public static Window simpleCalcWindow() throws Exception {...}
}

```

Fișierul **openAUT.bat** este executat prin program înaintea pornirii testelor automate. El are misiunea să deschidă automat aplicația CalcMVC. Va trebui să editați prima linie din acest fișier astfel încât aceasta să conțină calea absolută către fișierul bytecode CalcMVC.class:

```

cd d:\Work-applications\eclipse-workspace\CalcMVC_JUnit\bin\
java CalcMVC

```

Porniți execuția proiectului CalcMVC_LeanFT ca și JUnit Test. Veți observa cum se deschide mai întâi automat aplicația și apoi se rulează automat toate testele. Veți observa interacțiunea automată cu interfața grafică conform celor descrise mai sus. Așteptați terminarea testelor și în final vizualizați și analizați raportul generat prin accesarea "View Last Run Results" din meniul LeanFT:

Run Results

Execution Time 2017-01-03 18:36:28 Duration 00:00:23 Tool Name LeanFT 12.54

Errors list Test flow Search...

- ✖ Check if multiply works for 2 x 6 = 13 +00:00:02
- ✖ Value: 2 is valid and should be invalid +00:00:23

Verification Details

Verification
Check if multiply works for 2 x 6 = 13

Execution Time
2017-01-03 18:36:30

Operation
areEqual

Data

arg0	arg1
"13"	"12"

5. Mersul lucrării

- 5.1. Parcurgeți și înțelegeți exemplele din laborator.
- 5.2. Creați un proiect nou și urmăriți pașii descriși în lucrare pentru testarea unei alte aplicații GUI cu ajutorul instrumentului de testare automată LeanFT.