
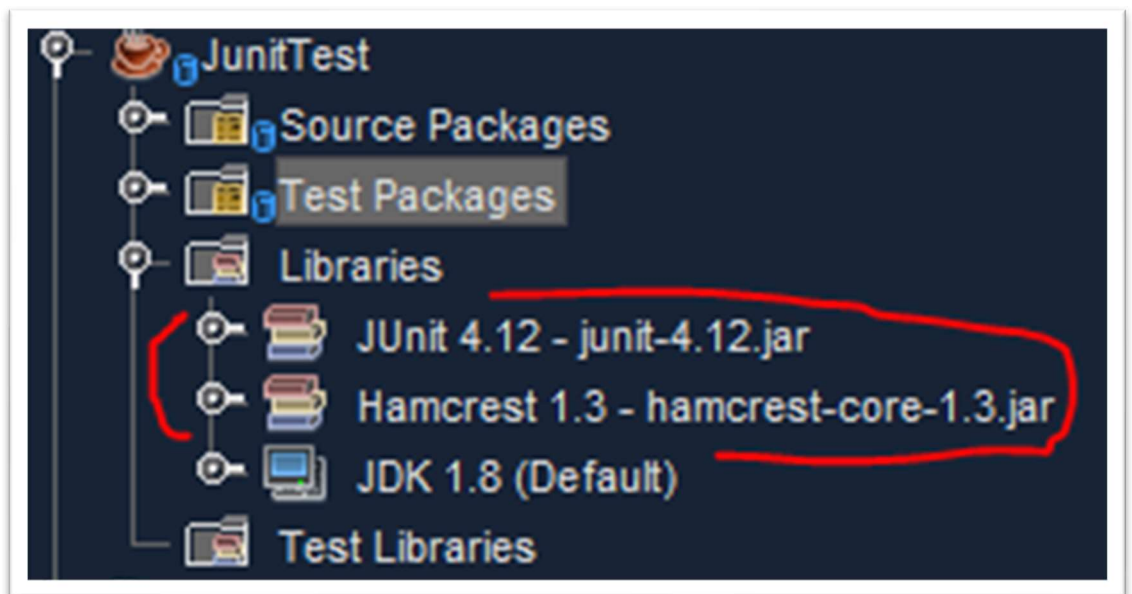


Atelier : JUnit

1- Configuration :

- a. Le choix de NetBeans comme environnement de travail 
- b. Télécharger la dernière version du librairie JUnit.jar
<https://junit.org/junit5/>
- c. Télécharger la librairie hamcrest.jar
<http://hamcrest.org/JavaHamcrest/>
- d. Créer un nouveau projet et ajouter ses 2 librairie à votre projet.

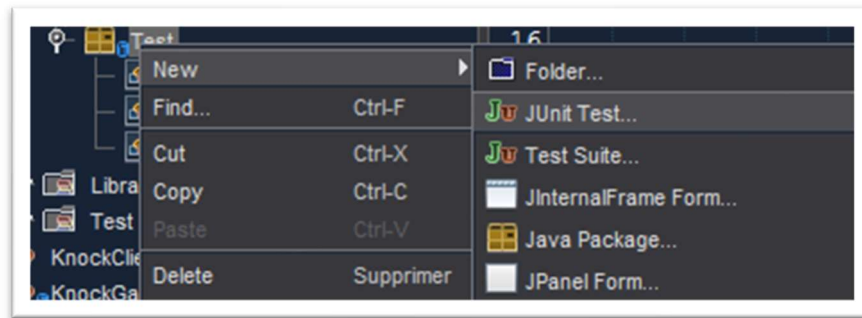


2- TP

- a. Créer une simple classe java avec des méthodes :
 - i. Somme
 - ii. Soustraction
 - iii. Si deux entiers sont égaux
 - iv. Prolonger la durée d'une méthode par **sleep(3000)**
 - v. Trier une liste avec Arrays.sort()

```
public class Exemple1 {  
  
    public int somme(int a, int b){  
        return a+b;  
    }  
  
    public int soustraction(int a,int b){  
        return a-b;  
    }  
  
    public boolean Equal(int a,int b){  
        return a==b;  
    }  
  
    public void time() throws InterruptedException{  
        sleep(3000);  
    }  
  
    public int[] tri(int[] a){  
        Arrays.sort(a);  
        return a;  
    }  
}
```

b. Créer la classe test de JUnit



c. Implémenter les différentes méthodes pour chaque méthode de notre classe initiale en utilisant les concepts que nous avons découverts lors de la présentation :

i. Somme :

```
@Test
public void TestSomme(){
    int a = T.somme(5, 5);
    assertEquals(10, a);
}
```

ii. Soustraction

```
@Test
public void TestSoustraction(){
    int a = T.soustraction(5, 5);
    assertEquals(0,a );
}
```

iii. Time

```
@Test(timeout = 2000)
public void TestTime() throws
InterruptedException{
    T.time();
}
```

iv. Tri

```
@Test
public void Array() {
    int [] data ={10,12,1,4,8};
    int [] ex ={1,4,8,10,12};
    T.tri(data);
    Assert.assertArrayEquals(data,ex);
}
```

v. L'utilisation de assertThat()

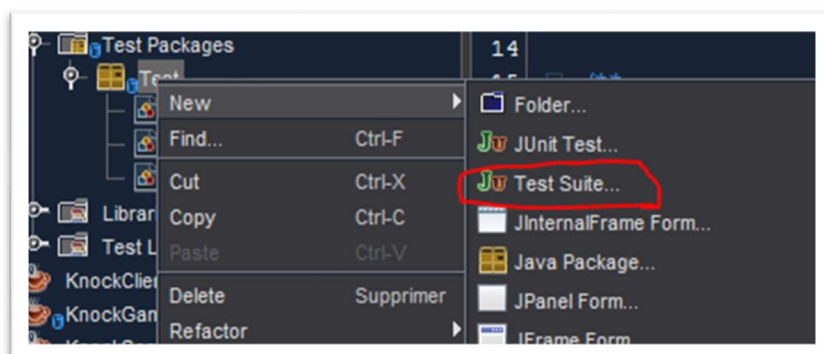
```
@Test
public void TestAssertThat(){

    assertEquals(1,1);
    assertThat(1,is(equalTo(1)));

    assertNotSame(new Object(),new Object());
    assertThat(new Object(), not(sameInstance(new Object())));

    assertThat("Miola 1A", CoreMatchers.containsString("1A"));
}
```

- d. JUnit permet l'exécution de plusieurs testclasse à la fois en utilisant la classe Test Suite



Merci