

اختبارات في إطار عمل hamcrest

المقارنة كائنات من مجموعة توفير على إطار عمل hamcrest. hamcrest هو إطار عمل اختبارات. في للكتابة قابلية أكثر تصريحات لتكتب

مشرعك: إلى hamcrest الاختبارية أضف أولاً،

ملفك إلى هذا أضف للمبانى،

```
<dependency>
  <groupId>org.hamcrest</groupId>
  <artifactId>hamcrest</artifactId>
  <version>2.2</version>
  <scope>test</scope>
</dependency>
```

ملفك إلى هذا أضف لغرادل،

```
testImplementation 'org.hamcrest:hamcrest:2.2'
```

ملفك مع hamcrest للاستخدام أساسى مثال إليك

```
import static org.hamcrest.MatcherAssert.assertThat;
import static org.hamcrest.Matchers.*;

import org.junit.Test;

public class ExampleTest {
    @Test
    public void testExamples() {
        //
        assertThat(5, is(5));

        //
        String text = "Hello World";
        assertThat(text, containsString("World"));
        assertThat(text, startsWith("Hello"));

        //
        List<String> items = Arrays.asList("apple", "banana", "orange");
        assertThat(items, hasItem("banana"));
```

```

        assertThat(items, hasSize(3));

        //
        assertThat(10, greaterThan(5));
        assertThat(3.14, closeTo(3.0, 0.2));
    }
}

```

3. في الشائعة المقارنات:

- `is()`: المساواة اختباري جري
- `equalTo()`: المساواة اختباري جري
- `containsString()`: فرع نص على يحتوي النص كان إذا ما فحصي جري
- `startsWith()` □ `endsWith()`: للاباءة □ اللاحقة فحص
- `greaterThan()` □ `lessThan()`: العدديّة المقارنات
- `hasItem()`: عن صر على تحتوي المجموعة كانت إذا ما فحصي جري
- `hasSize()`: المجموعة حجم فحصي جري
- `closeTo()`: نطاق ضمن العدد كان إذا ما فحصي جري
- `not()`: مقارنات أي ينفي
- `allOf()`: و □ مقارنات عدة بيني جري
- `anyOf()`: أو □ مقارنات عدة بيني جري

4. المخصصة: المقارنات على مثال

```

import org.hamcrest.BaseMatcher;
import org.hamcrest.Description;
import org.hamcrest.Matcher;

public class CustomMatchers {
    public static Matcher<String> isPalindrome() {
        return new BaseMatcher<String>() {
            @Override
            public boolean matches(Object item) {
                String str = (String) item;
                String reversed = new StringBuilder(str).reverse().toString();
                return str.equals(reversed);
            }
        };
    }

    @Override
    public void describeTo(Description description) {
        description.appendText("a palindrome");
    }
}

```

```

    }
};

}

}

// :
@Test
public void testPalindrome() {
    assertThat("racecar", isPalindrome());
}

```

5. المقارنات: بين المجموع

```

@Test
public void testCombined() {
    List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);
    assertThat(numbers, allOf(
        hasSize(5),
        hasItem(greaterThan(4)),
        everyItem(lessThan(10))
    ));
}

```

للتوسيع قابل مرن المقارنة شروط أفضل فشل رسائل للقرائة قابلية أكثر اختبارات تصريحات الرئيسية: المزايا
المخصصة المقارنات مع

وثنائى من تحقق معقدة لشروط المقارنات بجمع قم إكمالاً أكثر الكود لجعل `import static` استخدم نصوص:
الأخرى الاختبارات وأطر،،،،،، مع جدي بشكل،،،،،، يعمل المقارنات من كاملة مجموعة على للحصول
أخرى؟ أمثلة أظهر أن أو محددة مقارنات أي أشرح أن في ترغب هل