

Task1:

Create a separate **priority package**. It contains the **Priority1 class**. In this class, create tests **a, b, c, d, e, f, g** like this:

```
@Test
public void a() { assertTrue( condition: true); }
```

Make sure that when this class is run, these tests are run in reverse alphabetical order. Come up with at least two ways
How can I do that.

Task2:

Create a separate **grouping package**. It contains the **Groups class**. In this class, create tests **one, two, three, four, five, six, seven, eight** like this:

```
@Test
public void one(){
    assertTrue( condition: true);
}
```

Make sure that tests called odd numbers belong to the **first group**, and tests of even numbers belong to the **second group**.

Create a separate xml file **testngGroupingHome.xml** in which the files of the first group were run sequentially, and then the files of the **second group**.

PS In each group, tests must be taken in ascending order.

Task3:

Create a separate package **parallelismus**. It contains the class **ParallelClass1**. In which to create tests **parallel1**, **parallel2**, **parallel3**, **parallel4**, **parallel5**. Create a class **ParallelClass2**. In which to create tests **parallel6**, **parallel7**, **parallel8**, **parallel9**, **parallel10**. Tests in classes should be as follows:

```
@Test
public void parallel1() throws InterruptedException {
    Thread.sleep( millis: 2000);
    assertTrue( condition: true);
}
```

Create a separate xml file **testngParallelHome.xml** in which the two classes created above, **ParallelClass1** and **ParallelClass2**, will be run in parallel.

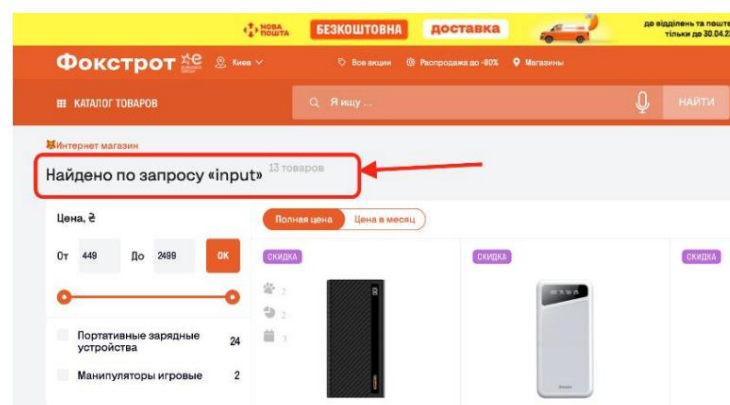
Task4:

Write a test that will check the search string of the site <https://www.foxtrot.com.ua/>

Use the following three words as test words:

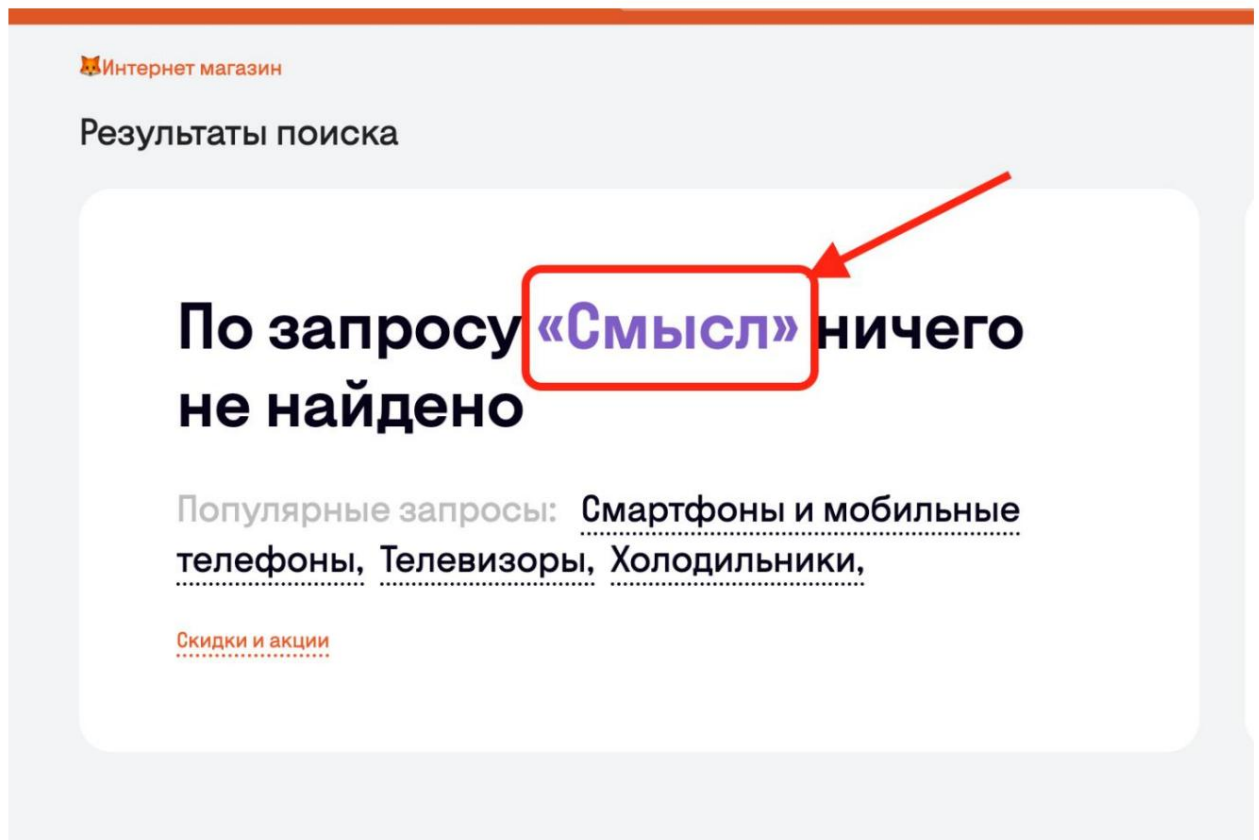
“machine”, “input”, “meaning”.

- If after searching for the entered word the user sees the page of such a type:



Then in your assertion you need to make sure that the selected element contains the search word.

- If after searching for the entered word the user sees the page of such a type:



Then in your assertion you also need to make sure that the selected element contains the search word.

I use @DataProvider to solve this problem .

Task5:

Solve the previous problem using the **@Parameters** annotation. And also create an additional xml file **testngParametersHome.xml** to work with this test.