

# JAVA 11

## Contents

Khái quát .....	2
Oracle và OpenJDK.....	2
Develop Feature.....	2
New String Method.....	2
New File Method.....	3
Collection To An Array .....	3
The Not Predicate Method .....	4
Local-Variable Syntax for Lambda.....	5
Http Client.....	5
Nest Based Access Control.....	5
Running Java Files .....	6
Cải tiến hiệu năng .....	7
Tham Khảo: .....	7

## Khái quát

Oracle đã released java 11 trong tháng 9-2018, chỉ sau 6 tháng sau khi version 10 ra mắt. Java 11 là long-term support (LST) released sau java 8. Oracle cũng dừng hỗ trợ java 8 trong 1-2019. Do đó, nhiều người chúng ta phải nâng cấp lên java 11. Trong phần này chúng ta sẽ xem các tùy chọn của java 11. Những feature mới, những xóa bỏ và các cải tiến.

## Oracle và OpenJDK

Java 10 là phiên bản free cuối cùng của Oracle JDK cho chúng ta sử dụng trong thương mại mà không cần license, bắt đầu với java 11 không còn miễn phí hỗ trợ long-term support.

Rất may, Oracle tiếp tục cung cấp OpenJDK releases, Có thể download và sử dụng không cần trả phí. Bên cạnh đó cũng có nhiều nhà cung ứng OpenJDK khác. Amazon, azure, ...

## Develop Feature

### New String Method

Java 11 thêm một vài các phương thức mới cho String class: `isBlank`, `Lines`, `Strip`, `stripLeading`, `stripTrailing`, and `Repeat`.

`Repeat` a string following by the number definition before.

`Strip` was used that remove white space it better trip because it nhận biết (aware)

Kí tự khoảng trắng của unicode còn `trim()` thì không

`isBlank()` return true if string is blank hoặc chỉ contain only whitespace, it is aware whitespace of unicode just like `strip()` .

`lines` return stream from character separated by line terminators (`\n`, `\r`, `\n\r`). Nó được ưu tiên (preferred) sử dụng hơn `split(string)` vì nó better performance for beaking multi-lines.

```
workspace - java11feature/src/main/java/java11feature/methodsnew/strings/StringNewFeature.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

StringNewFeature.java
1 package java11feature.methodsnew.strings;
2
3 public class StringNewFeature {
4
5     // repeat 5 times a : aaaaa
6     static void repeatString() {}
7
8     // removed whitespace leading and trailing , it is aware of Unicode whitespace
9     // characters (\u2005)
10    static void tripleLeadAndTail() {}
11
12    // The isBlank() instance method returns true if the string is empty or contains
13    // only whitespace. Otherwise, it returns false:
14    static void isBlankCheck() {}
15
16    // lines return a stream extracted from the string , separated by line
17    // terminators ("\\n", "\\r", or "\\r\\n".)
18    static void getStreamFromLines() {}
19
20    public static void main(String[] args) {
21        repeatString();
22        tripLeadAndTail();
23        isBlankCheck();
24        getStreamFromLines();
25    }
26 }
```

## New File Method

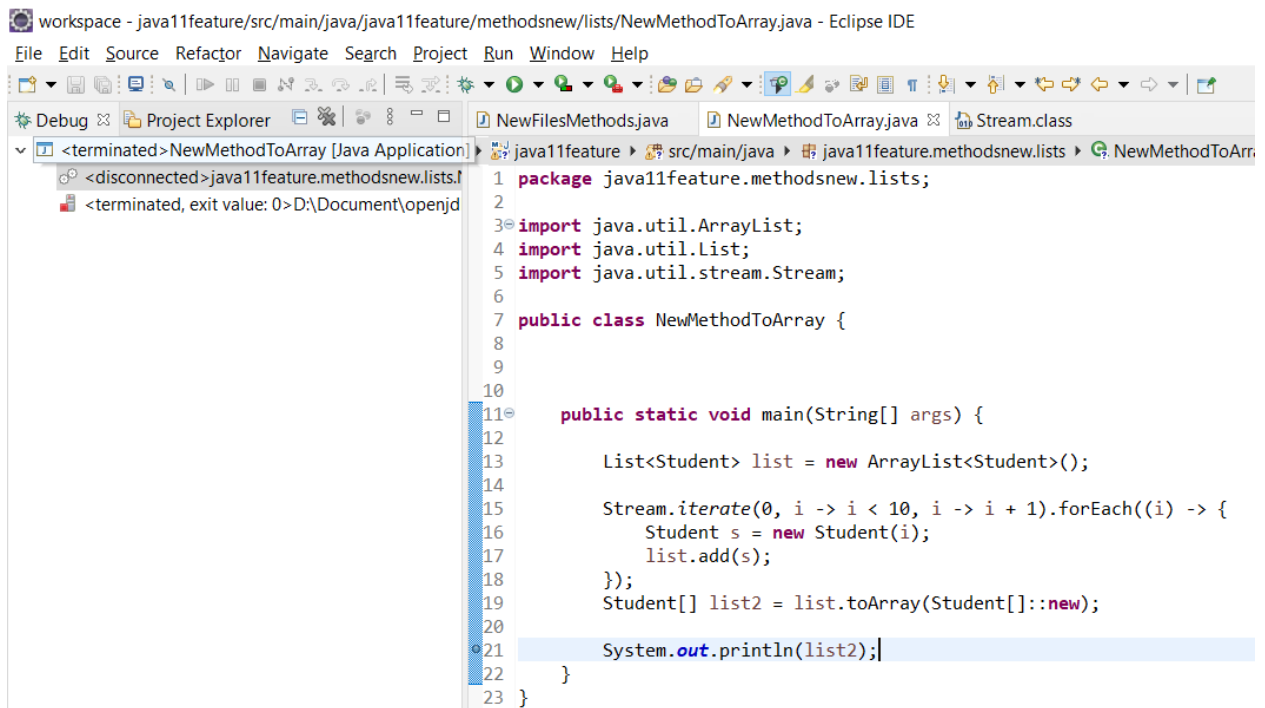
Ngoài ra, Java 11 cũng hỗ trợ cho việc read và ghi String dễ hơn từ các file.

```
workspace - java11feature/src/main/java/java11feature/methodsnew/files/NewFilesMethods.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

NewFilesMethods.java
1 package java11feature.methodsnew.files;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.nio.charset.StandardCharsets;
6 import java.nio.file.Files;
7 import java.nio.file.NoSuchFileException;
8 import java.nio.file.Path;
9 import java.nio.file.Paths;
10
11 public class NewFilesMethods {
12
13     static void writeStringsIntoFile() {}
14
15     static void readStringsFromFile() {}
16
17     public static void main(String[] args) throws IOException {
18         writeStringsIntoFile();
19         readStringsFromFile();
20     }
21 }
```

## Collection To An Array

The java.util.Collection interface chứa một phương thức mới là default toArray() cần 1 đối số là IntFunction. Tự truyền Student[]:new tự động jvm sẽ hiểu không cần khai báo gì thêm vd:

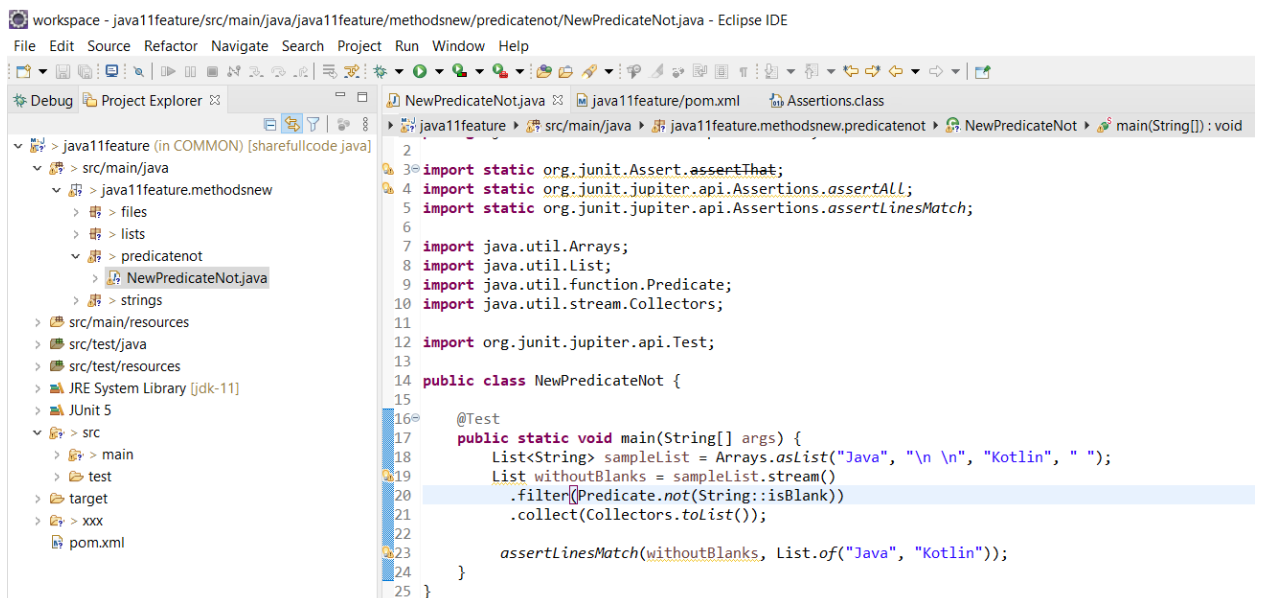


```
workspace - java11feature/src/main/java/java11feature/methodsnew/lists/NewMethodToArray.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug Project Explorer NewFilesMethods.java NewMethodToArray.java Stream.class
<terminated>NewMethodToArray [Java Application]
<disconnected>java11feature.methodsnew.lists.f
<terminated, exit value: 0>D:\Document\openjd

1 package java11feature.methodsnew.lists;
2
3 import java.util.ArrayList;
4 import java.util.List;
5 import java.util.stream.Stream;
6
7 public class NewMethodToArray {
8
9
10
11 public static void main(String[] args) {
12
13     List<Student> list = new ArrayList<Student>();
14
15     Stream.iterate(0, i -> i < 10, i -> i + 1).forEach((i) -> {
16         Student s = new Student(i);
17         list.add(s);
18     });
19     Student[] list2 = list.toArray(Student[]::new);
20
21     System.out.println(list2);
22 }
23 }
```

## The Not Predicate Method

Một static “not method” đã được thêm vào Predicate interface. Chúng ta có thể sử dụng nó để phủ định một predicate tồn tại. sử dụng `not(isBlank)` đọc sẽ tự nhiên hơn là dùng `isBlank.negate()`.

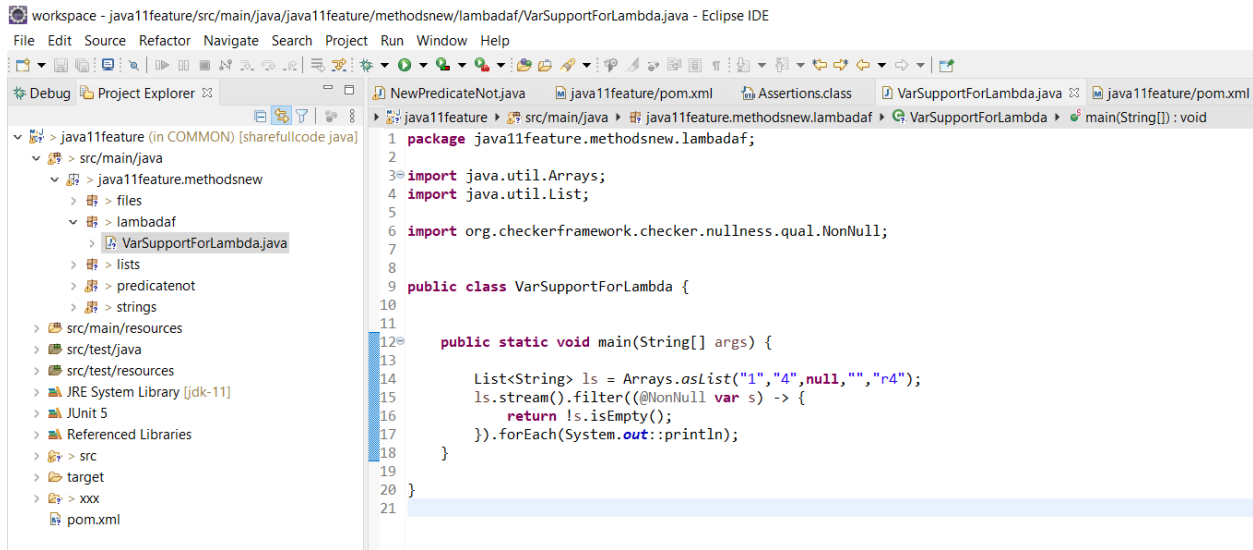


```
workspace - java11feature/src/main/java/java11feature/methodsnew/predicate/Not/NewPredicateNot.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug Project Explorer NewPredicateNot.java java11feature/pom.xml Assertions.class
java11feature (in COMMON) [sharefullcode java]
src/main/java
  java11feature.methodsnew
    files
    lists
    predicate/
      NewPredicateNot.java
    strings
src/main/resources
src/test/java
src/test/resources
JRE System Library [jdk-11]
JUnit 5
src
  main
  test
target
xxx
pom.xml

2
3 import static org.junit.Assert.assertThat;
4 import static org.junit.jupiter.api.Assertions.assertAll;
5 import static org.junit.jupiter.api.Assertions.assertLinesMatch;
6
7 import java.util.Arrays;
8 import java.util.List;
9 import java.util.function.Predicate;
10 import java.util.stream.Collectors;
11
12 import org.junit.jupiter.api.Test;
13
14 public class NewPredicateNot {
15
16 @Test
17 public static void main(String[] args) {
18     List<String> sampleList = Arrays.asList("Java", "\n\n", "Kotlin", " ");
19     List withoutBlanks = sampleList.stream()
20         .filter(Predicate.not(String::isBlank))
21         .collect(Collectors.toList());
22
23     assertLinesMatch(withoutBlanks, List.of("Java", "Kotlin"));
24 }
25 }
```

## Local-Variable Syntax for Lambda

Java 11 hỗ trợ việc sử dụng var keyword trong lambda parameters. Trong lambda function có thể ko cần phải thêm datatype trước biến nhưng mục đích thêm var hỗ trợ cho lambda là kết hợp cùng annotation.



The screenshot shows the Eclipse IDE interface. The Project Explorer on the left displays the project structure: java11feature (in COMMON) [sharefullcode java] > src/main/java > java11feature.methodsnew > lambadaf > VarSupportForLambda.java. The main editor shows the code for VarSupportForLambda.java. The code includes package, imports, and a main method. Line 15 highlights the use of the var keyword in a lambda function parameter.

```
1 package java11feature.methodsnew.lambadaf;
2
3 import java.util.Arrays;
4 import java.util.List;
5
6 import org.checkerframework.checker.nullness.qual.NonNull;
7
8
9 public class VarSupportForLambda {
10
11
12     public static void main(String[] args) {
13
14         List<String> ls = Arrays.asList("1", "4", null, "", "r4");
15         ls.stream().filter((@NonNull var s) -> {
16             return !s.isEmpty();
17         }).forEach(System.out::println);
18     }
19 }
20
21
```

## Http Client

The new http client từ java.net.http package đã được giới thiệu trong java 9. Nó bây giờ đã trở thành 1 standard feature trong java 11.

Http mới cải tiến tất cả hoạt động và cung cấp hỗ trợ cả http1.1 và http 2

## Nest Based Access Control

Java 11 giới thiệu khái niệm Nestmates và liên kết truy cập các luật trong JVM. Ví dụ sử dụng class outer and netsted class.

```
NewPredicateNot.java  java11feature/pom.xml  Assertions.class  VarSupportForLambda.java  java11feature/pom.xml  NetstedAndOuterClass.java
> java11feature > src/main/java > java11feature.methodsnew.classes > NetstedAndOuterClass > A_Level1 > method_A_Level1() : void
7  NetstedAndOuterClass(){
8      a_Level1 = new A_Level1();
9  }
10
11  // java 11 support class netsted access private field of class outer
12  private String propertiesOuter = " NetstedAndOuterClass ";
13
14  private void methodOuter() {
15      System.out.println(" NetstedAndOuterClass " + propertiesOuter);
16  }
17
18  class A_Level1 {
19
20      private A_Level2 a_Level2;
21
22      A_Level1(){
23          a_Level2 = new A_Level2();
24      }
25
26      private String propertiesA_Level1 = " A_Level1 ";
27      private void method_A_Level1() {
28          // java 11 use properties private of outer class
29          System.out.println("method_A_Level1 " + propertiesOuter + propertiesA_Level1);
30      }
31
32
33      class A_Level2 {
34          private String propertiesA_Level2 = " A_Level2 ";
35          private void method_A_Level2() {
36              // java 11 use properties private of outer class
37              System.out.println("A_Level2 calling " + NetstedAndOuterClass.this.propertiesOuter
38                  + A_Level1.this.propertiesA_Level1 + propertiesA_Level2);
39          }
40      }
41  }
```

## Running Java Files

Bây giờ chúng ta chỉ cần gọi : java file.java và run luôn không cần phải compile javac file trước nữa

```
Command Prompt
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Laptop>cd C:\Users\Laptop\Desktop

C:\Users\Laptop\Desktop>java --version
openjdk 11 2018-09-25
OpenJDK Runtime Environment 18.9 (build 11+28)
OpenJDK 64-Bit Server VM 18.9 (build 11+28, mixed mode)

C:\Users\Laptop\Desktop>java javas.jajva
Error: Could not find or load main class javas.jajva
Caused by: java.lang.ClassNotFoundException: javas.jajva

C:\Users\Laptop\Desktop>java javas.java
hello

C:\Users\Laptop\Desktop>
```

## Cải tiến hiệu năng

Dynamic Class-File Constants: Format của file java được mở rộng hỗ trợ một constant-pool mới có tên là CONSTANT-Dynamic.

Improve Aarch64...

Tham Khảo:

<https://www.baeldung.com/java-11-new-features>