Name: Lutika Kolhe

Core Java Assignment 10

# Assignments on Java 11 Feature

1. Write a program to calculate the simple Interest with minimal code using features of Java 11.

HINT: Use the concept of functional interface and Local variable syntax for lambda parameters.

Ans- Here I created 2 class first is SimpleIntrest.java and another is MainMetod.java

Class1 : SimpleIntrest.java

**package** Java11;

**interface** SimpleInterest

{

**int** calculate(**int** p, **int** rate, **int** time);

}

Class2 : MainMetod.java

**package** Java11;

**import** java.util.Scanner;

**public** **class** MainMetod

{

**public** **static** **void** main(String[] args)

{

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter Amount: ");

**int** p = sc.nextInt();

System.***out***.println("Enter the Rate:" );

**int** rate = sc.nextInt();

System.***out***.println("Enter the Time: ");

**int** time = sc.nextInt();

SimpleInterest i = (**int** a, **int** b, **int** c) -> (a\*b\*c)/100;

**int** res = i.calculate(p, rate, time);

System.***out***.println("Simple Interest: "+res);

}

}

**Output:**

Enter Amount:

303030

Enter the Rate:

330

Enter the Time:

3

Simple Interest: 2999997

1. Java 11 supports var keyword for variable declarations. List the scenarios where var keyword cannot be used for such variable declarations. Give reason in support of your answer for each scenario.

Scenario 1: var keyword cannot be used as an instance and a global variable

**package** Java11;

**public** **class** VarKeyword

{

// instance variable

**var** x = 50;

**public** **static** **void** main(String[] args)

{

System.***out***.println(x);

}

}

**Output:**

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

Cannot make a static reference to the non-static field x

at Java11.VarKeyword.main(VarKeyword.java:9)

Scenario 2: var cannot be used as a generic type

**package** Java11;

**import** java.util.\*;

**public** **class** VarKeyword

{

**public** **static** **void** main(String[] args)

{

**var**<**var**> al = **new** ArrayList<>();

al.add(123);

al.add(456);

System.***out***.println(al);

}

}

**Output:**

[123, 456]

Scenario 3: var cannot be used without explicit initialization

**package** Java11;

**import** java.io.\*;

**class** VarKeyword {

**public** **static** **void** main(String[] args) {

**var** variable;

**var** variable = **null**;

}

}

**Output:**

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

Cannot use 'var' on variable without initializer

Duplicate local variable variable

Cannot infer type for local variable initialized to 'null'

at Java11.VarKeyword.main(VarKeyword.java:8)

Scenario 4: var cannot be used with lambda expression

**package** Java11;

**interface** Integer {

**int** add(**int** a, **int** b);

}

**class** VarKeyword {

**public** **static** **void** main(String[] args) {

**var** v = (a,b) -> (a + b);

System.***out***.println(v.add(2,3));

}

}

**Output:**

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

Lambda expression needs an explicit target-type

Cannot invoke add(int, int) on the primitive type null

at Java11.VarKeyword.main(VarKeyword.java:9)

1. A quick brown fox jumps over the lazy dog”. Create an ArrayList from the given String .Such an ArrayList should include each word from the given sentence. Finally. convert such List to an array using Java 11 methods and print the output.

**package** Java11;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** ConvertToArray {

**public** **static** **void** main(String[] args)

{

String str = "A quick brown fox jumps over the lazy dog";

String words[] = str.split(" ");

List<String> l = **new** ArrayList<String>();

**for**(String text:words) {

l.add(text);

}

System.***out***.println(l);

//java11

String[] array = l.toArray(String[]::**new**);

System.***out***.println("In java 11 features : " + Arrays.*toString*(array));

}

}

**Output:**

[A, quick, brown, fox, jumps, over, the, lazy, dog]

In java 11 features : [A, quick, brown, fox, jumps, over, the, lazy, dog]

1. Using features of Java 11, read the data from a text file (File name: StudentList.bt). Calculate the count of students and print the names as well as the total count of students on the screen. (If any line in file doesn't contain a name, for such a record blank space should not be printed in the output)

Hint: Use java 11 features of files and String methods to reduce the lines of code to be written.

**package** Java11;

**import** java.io.IOException;

**import** java.nio.file.Files;

**import** java.nio.file.Path;

**import** java.nio.file.Paths;

**public** **class** StudentList

{

**public** **static** **void** main(String[] args)

{

Path filePath = Paths.*get*("C:/","Users","hp","Assignments","Lutika","Java11Assignments\_StudentList.txt");

**try**

{

String line;

**int** count =0;

String content = Files.*readString*(filePath);

System.***out***.println(content);

String words[] = content.split(" ");

count = count + words.length;

System.***out***.println("Total Name Count: " + count);

}

**catch** (IOException o)

{

o.printStackTrace();

}

}

}

**Output:**

Shubham

Niharika

Lutika

Shubhashish

Harshita

Vedant

Rohan

Total Name Count: 7

1. Write a code using HttpClient API which sends a GET request to <https://httpbin.org/get>, and print out the response header, status code, and body for the given URL Sample output could be (Note: date and other attribute values may differ in your case.

**package** Java11;

**import** java.io.IOException;

**import** java.net.URI;

**import** java.net.http.HttpClient;

**import** java.net.http.HttpRequest;

**import** java.net.http.HttpResponse;

**import** java.net.http.HttpClient.Version;

**import** java.net.http.HttpResponse.BodyHandlers;

**public** **class** SendGETRequest {

**public** **static** **void** main(String[] args) {

String uri = "https://httpbin.org/get";

HttpRequest req = HttpRequest.*newBuilder*()

.uri(URI.*create*(uri))

.GET()

.version(Version.***HTTP\_2***)

.build();

HttpClient client = HttpClient.*newBuilder*()

.build();

**try**

{

HttpResponse<String> resp = client.send(req, BodyHandlers.*ofString*());

System.***out***.println(resp.headers());

System.***out***.println(resp.statusCode());

System.***out***.println(resp.body());

}

**catch**(IOException | InterruptedException e)

{

e.printStackTrace();

}

}

}

**Output:**

java.net.http.HttpHeaders@4f7377d4 { {:status=[200], access-control-allow-credentials=[true], access-control-allow-origin=[\*], content-length=[244], content-type=[application/json], date=[Fri, 21 Jan 2022 13:03:04 GMT], server=[gunicorn/19.9.0]} }

200

{

"args": {},

"headers": {

"Host": "httpbin.org",

"User-Agent": "Java-http-client/17.0.1",

"X-Amzn-Trace-Id": "Root=1-61eaaf08-5f656b56456a2ec83e227b7c"

},

"origin": "182.68.46.21",

"url": "https://httpbin.org/get"

}