**What is a Groovy Script?**

APACHE GROOVY is an object oriented and Java syntax compatible programming language built for the Java platform.

* Groovy source code gets compiled into Java Bytecode so it can run on any platform that has JRE is installed.
* It is designed as a companion, not a replacement for Java.
* Groovy supports Dynamic Typing.

**Types of Programming:**

* Static programming language: Static programming-means all checks happen at compile time. Static typing means that types are known and checked for correctness before running your program. This is often done by the language's compiler.
* Dynamic programming language: Dynamic programming means checks and error, bugs all that happen at run time
* Closure programming: A groovy closure is a piece of code wrapped as an object. It acts as a method or a function.

Features of Groovy

Groovy has the following features −

* Support for both static and dynamic typing.
* Support for operator overloading.
* Native syntax for lists and associative arrays.
* Native support for regular expressions.
* Native support for various markup languages such as XML and HTML.
* Groovy is simple for Java developers since the syntax for Java and Groovy are very similar.
* You can use existing Java libraries.
* Groovy extends the java.lang.Object.

Differences in java and groovy:

1. java-syntax compatible OOP for java platform
2. Can be used as programming language as well as scripting language
3. Source code gets compiled into Java Bytecode so it can run on any platform that has JRE is installed.
4. It is designed as a companion, not a replacement for Java
5. A superset of java
6. Some general purpose packages and classes are imported by default but in Java only java.lang.\* package imported by default.
7. Variables are defined using the keyword "def," and the type of a variable does not need to be declared in advance. The compiler figures out the variable type at runtime
8. Default access modifier-public in java
9. Behaviour of ==
10. If you want to write a multiline string: “ “ “ vvvvvvvv” “ “
11. Default Imports:In Groovy some general purpose packages and classes are imported by default:
    1. java.io.\*
    2. java.lang.\*
    3. java.math.BigDecimal
    4. java.math.BigInteger
    5. java.net.\*
    6. java.util.\*
    7. groovy.lang.\*
    8. groovy.util.\*

Where in Java only java.lang.\* package imported by default.

1. In Java default access modifier is package i.e. if you don’t specify access modifier for fields,methods or class it becomes package-private.

In Groovy it’s by default public.

1. Groovy traits:

ex:trait Marks{

void display(){

println(“hello”);}}

Groovy- Decision Making

if statement

if/else statement

nested if statement

switch statement

nested switch statement

* Like java to run a for loop in groovy we use terms:
* Upto: 0.upto(4) {println "$it"}

$it is a closure that gives the value of the current loop.

* Times: 4.times{ print "$it" }

1. Groovy-Lists: can store object references ex: ['Groovy', 2,4 2.6]
2. Groovy-Maps: A collection of key value pairs.

* [ : ] Represent an Empty map.

**Groovy-Closures**

A groovy closure is a piece of code wrapped as an object. It acts as a method or a function.

def myClosure = {

println "My First Closure"

}

myClosure()

def myClosure = {

a,b,c-> it represesents parameter and the arrow represents end of paramaters

y = a+b+c

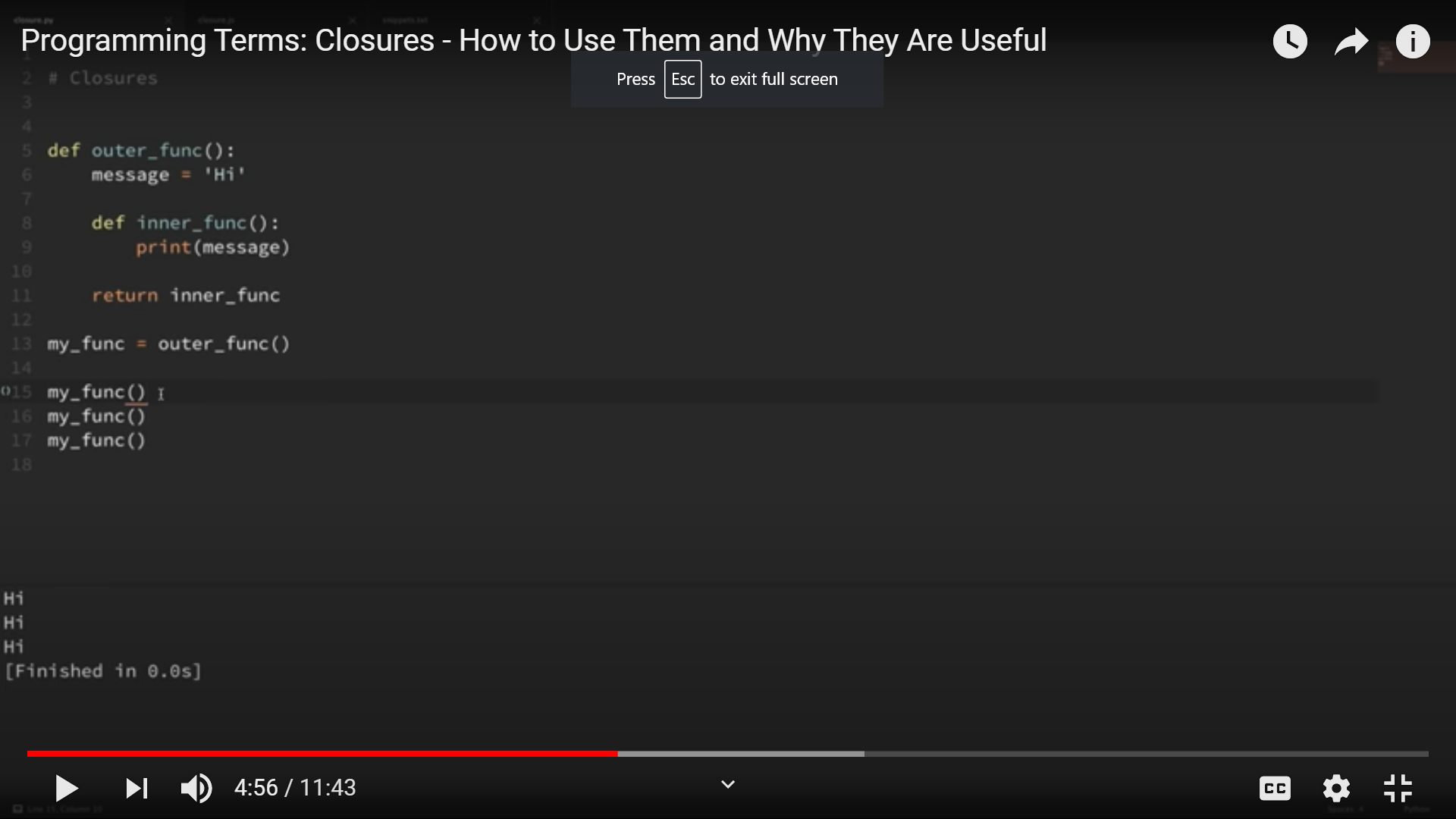
return y;

}

myClosure(1,2,3)

* In Groovy you don’t need that. Since Groovy is a Scripting language, there’s automatically a wrapping class called Script for every program.
* Closure: A inner function that remembers and has access to variables to local scope in which it was created even after the outer function has finished execution. The variables are remembered by closures . A closure closes over the free variables in their environment.

They are infact objects in groovy and can be passed to functions as arguments as well.



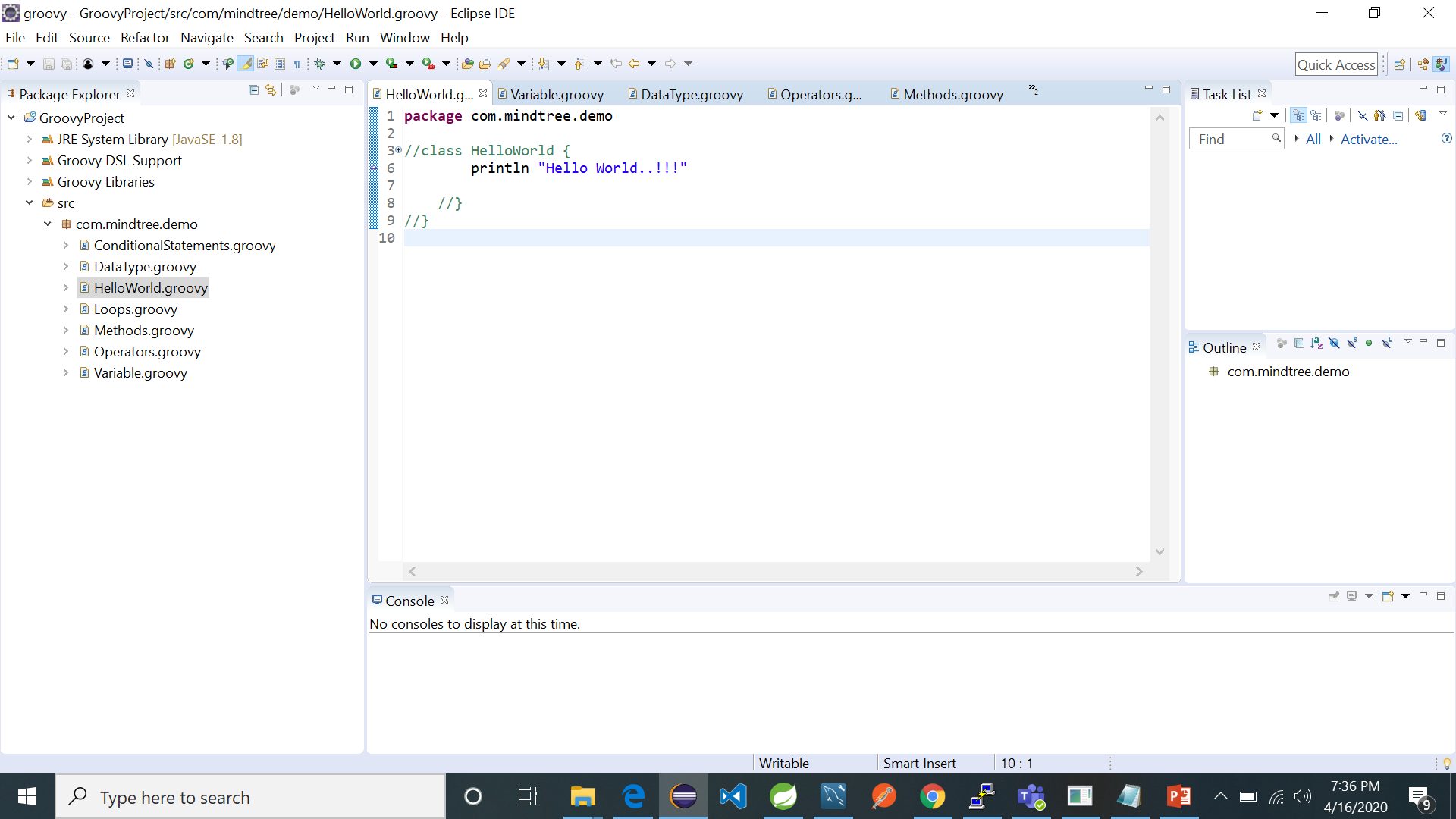
Traits are a structural construct of the language which allow −

* Composition of behaviors.
* Runtime implementation of interfaces.
* Compatibility with static type checking/compilation

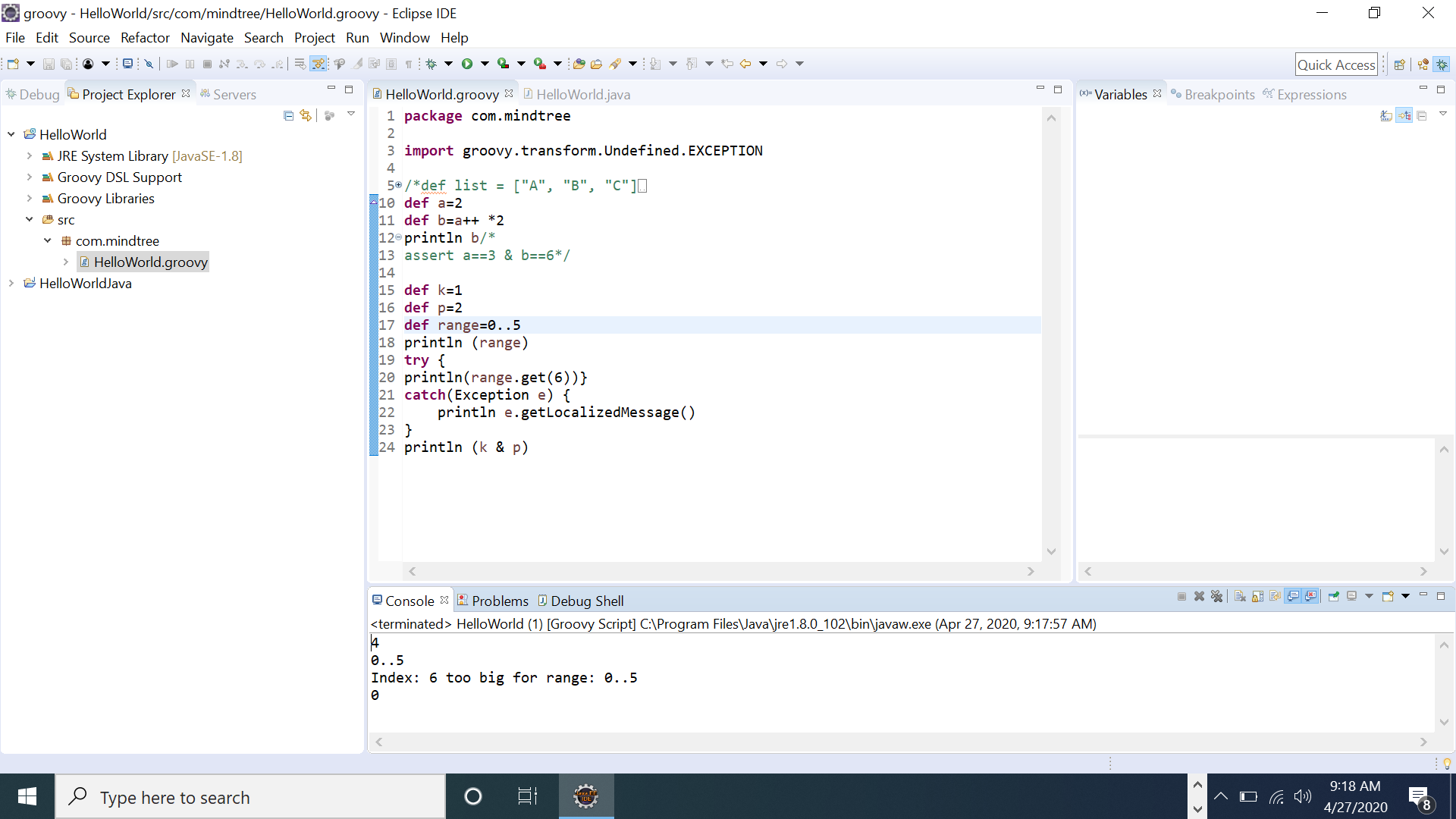
They can be seen as interfaces carrying both default implementations and static. A trait is defined using the trait keyword.

**Programs on groovy:**

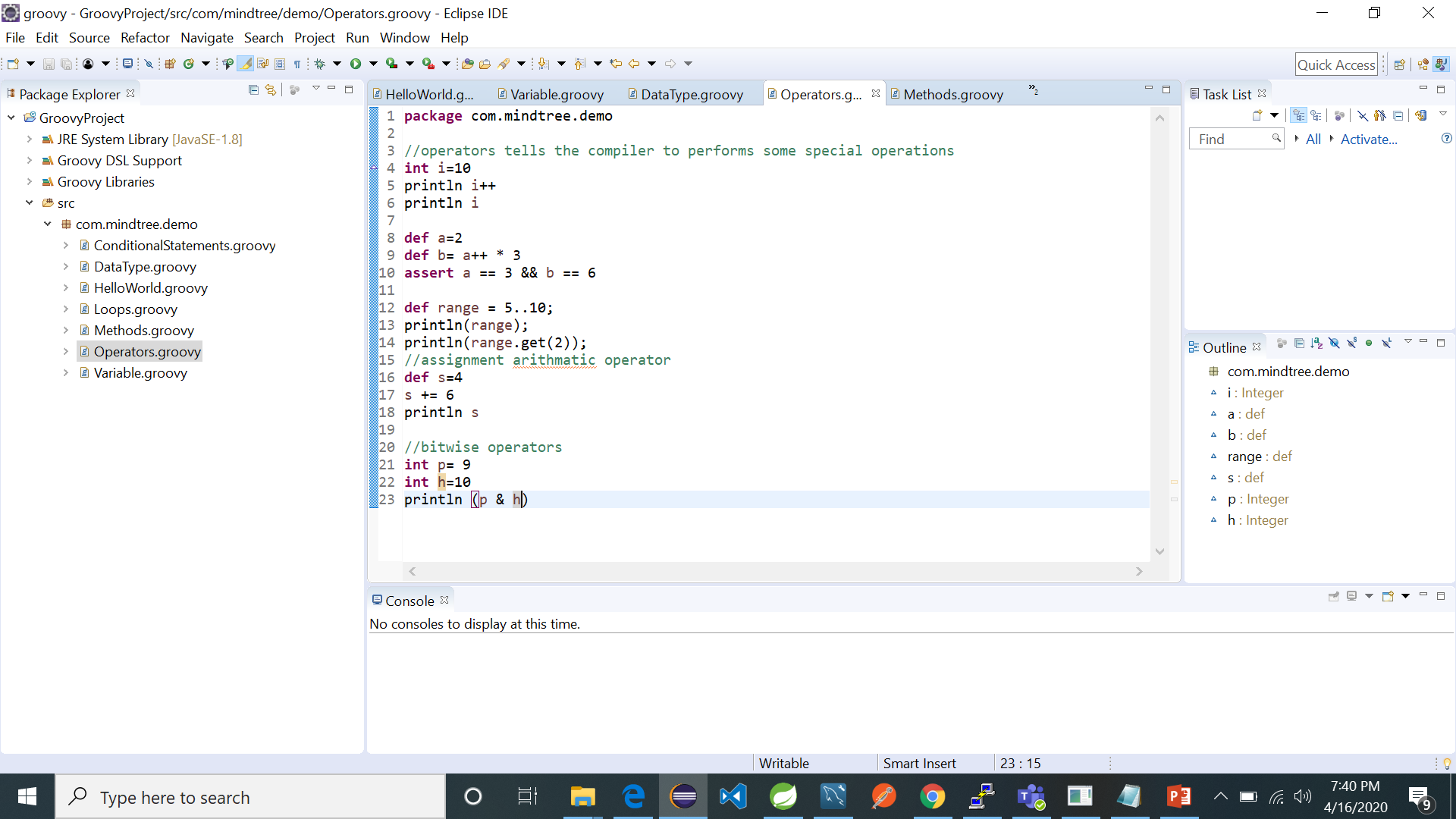
1.



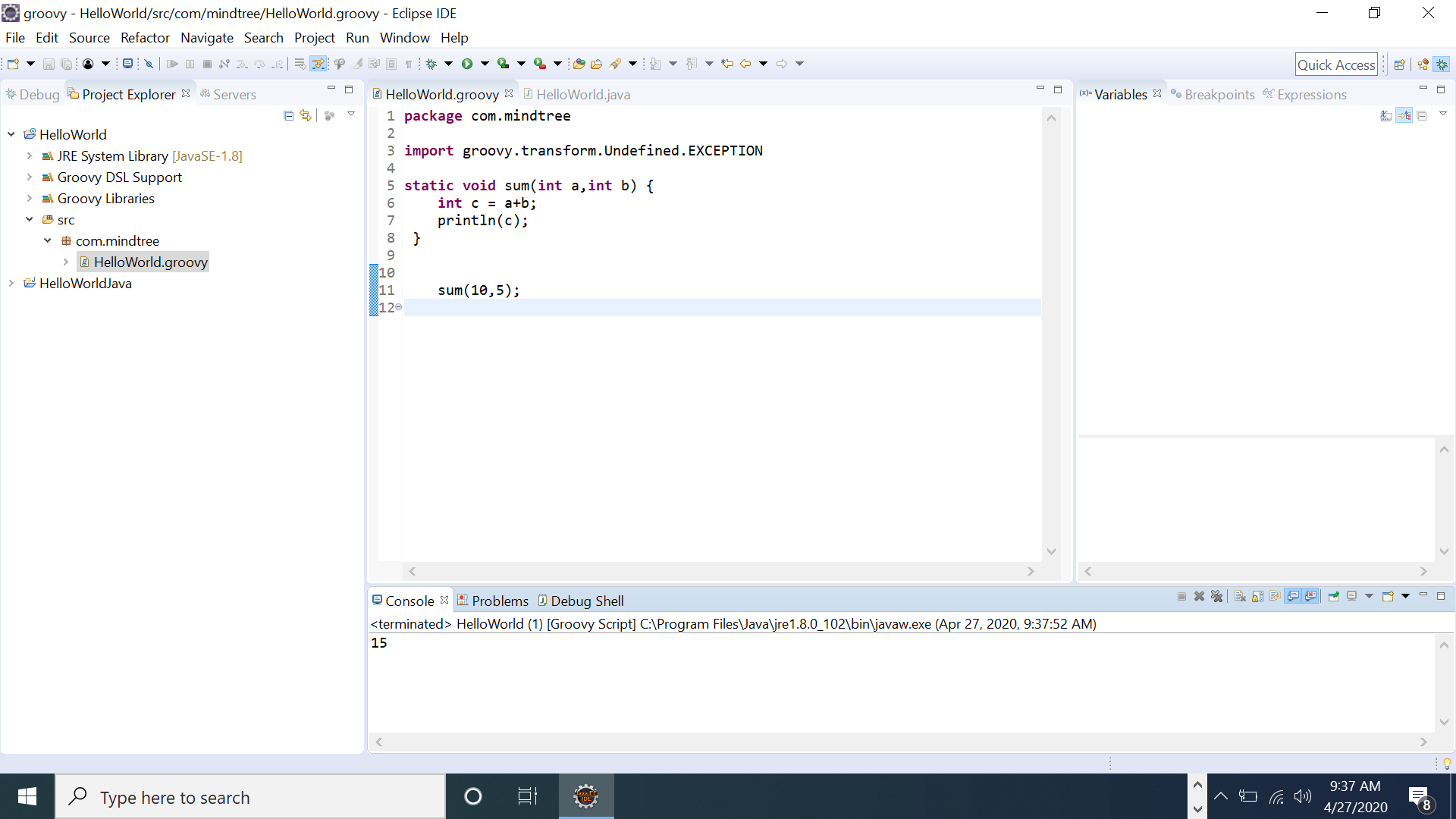
2.



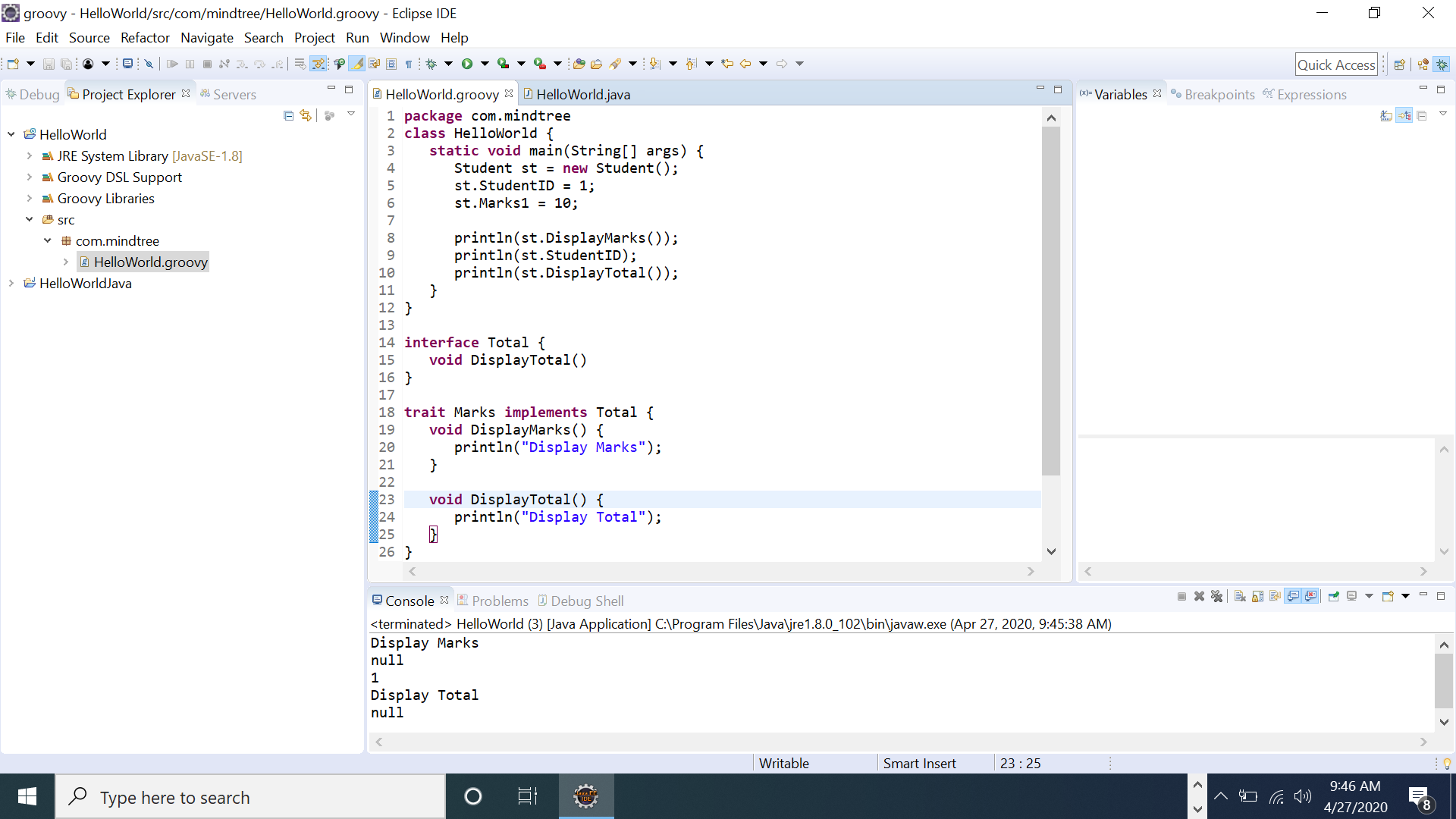
3.



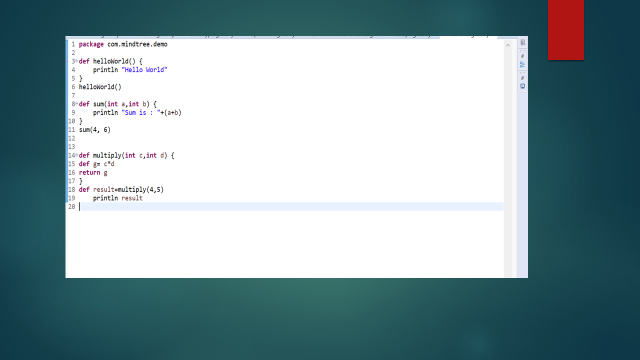
4.



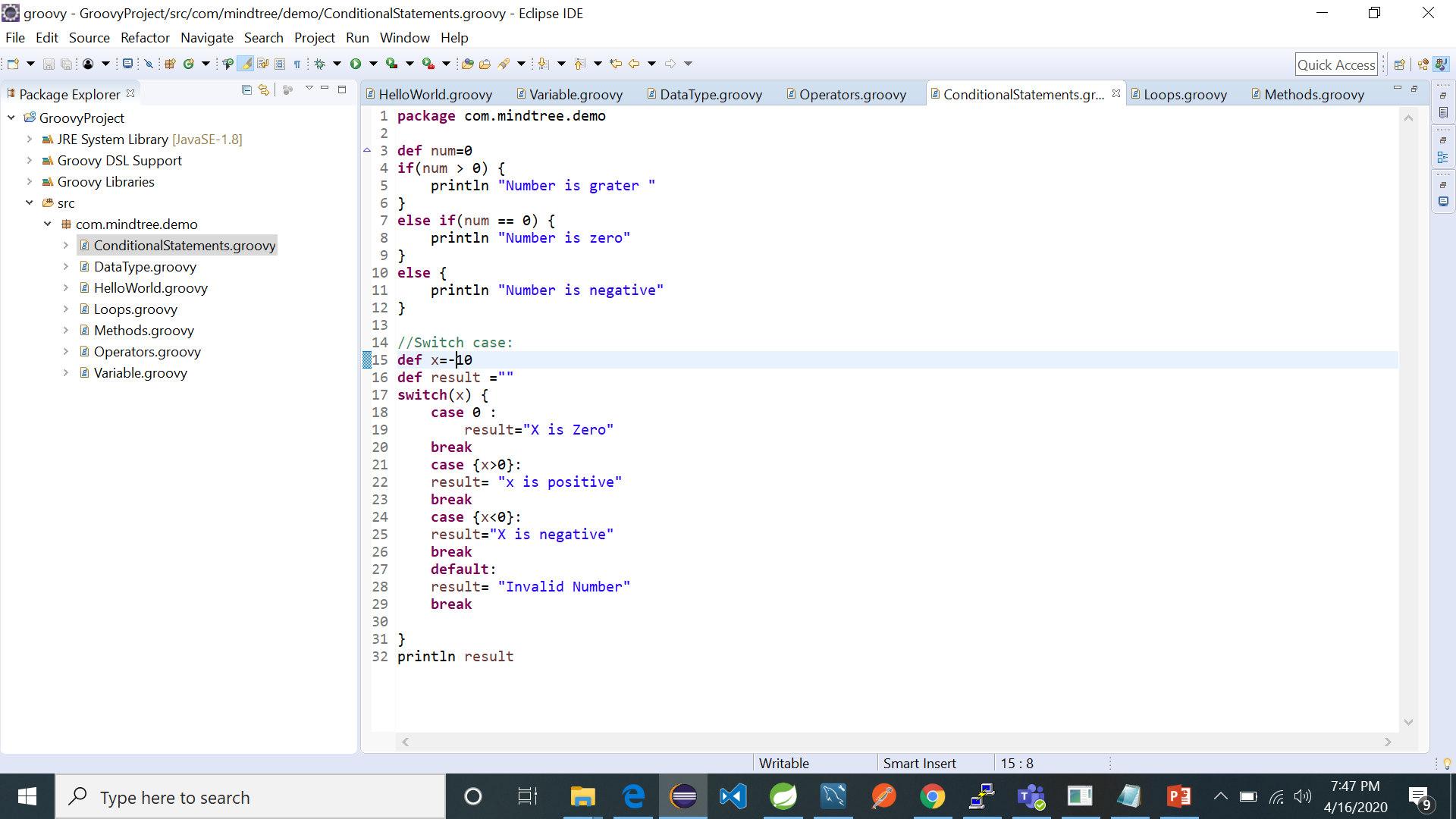
5.



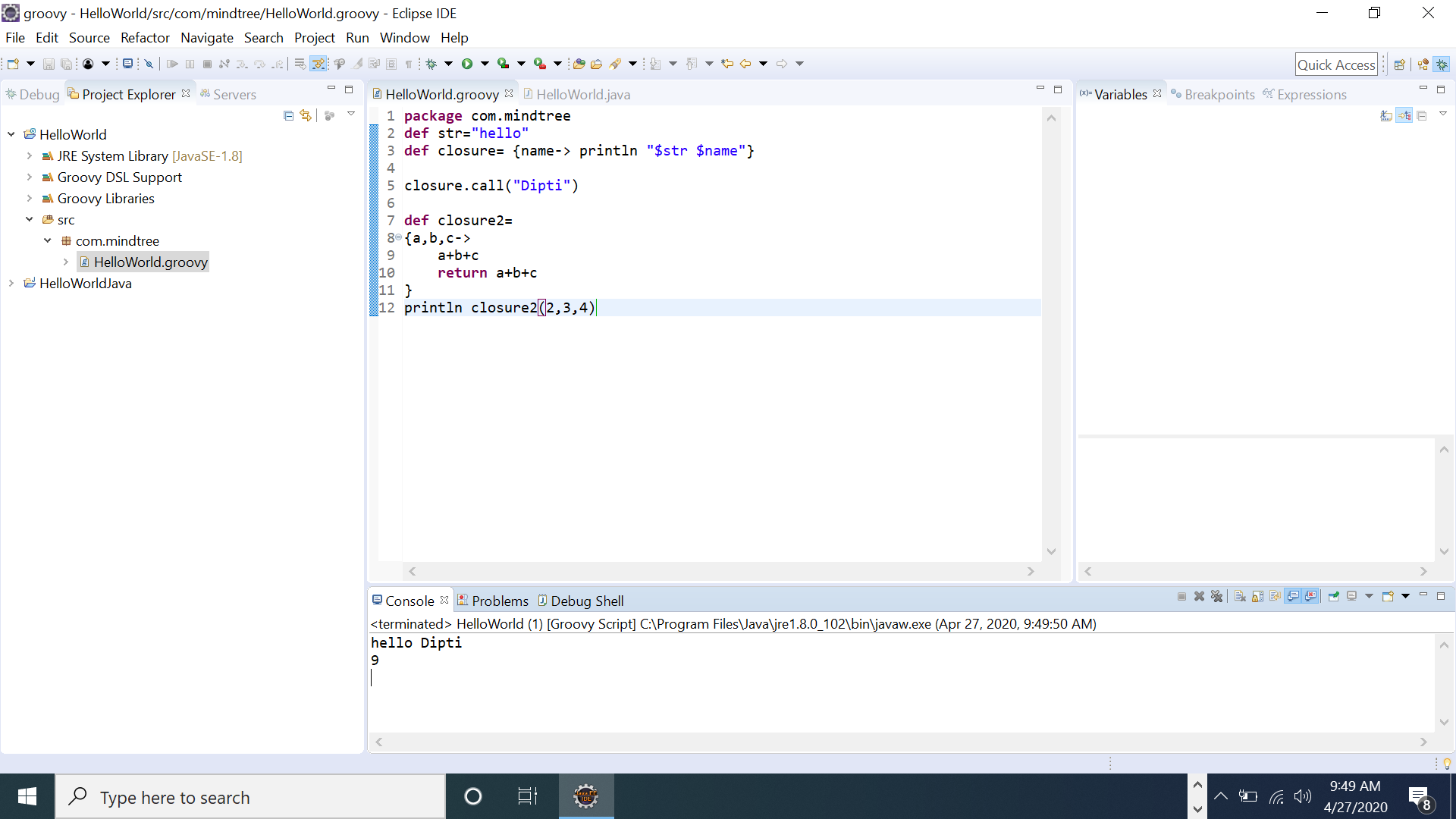
6.



7.



8.



EXAMPLE OF JENKINS PIPELINE USING GROOVY:

node{

stage('SCM Checkout'){

git 'https://github.com/Diptimb/Maven-Project'

}

stage('Compile Package'){

def mvnHome= tool name: 'Maven 3', type: 'maven'

sh "${mvnHome}/bin/mvn package"

}

}