

GROOVY PROJECT

DECISION MAKING:

1. IF Statement:



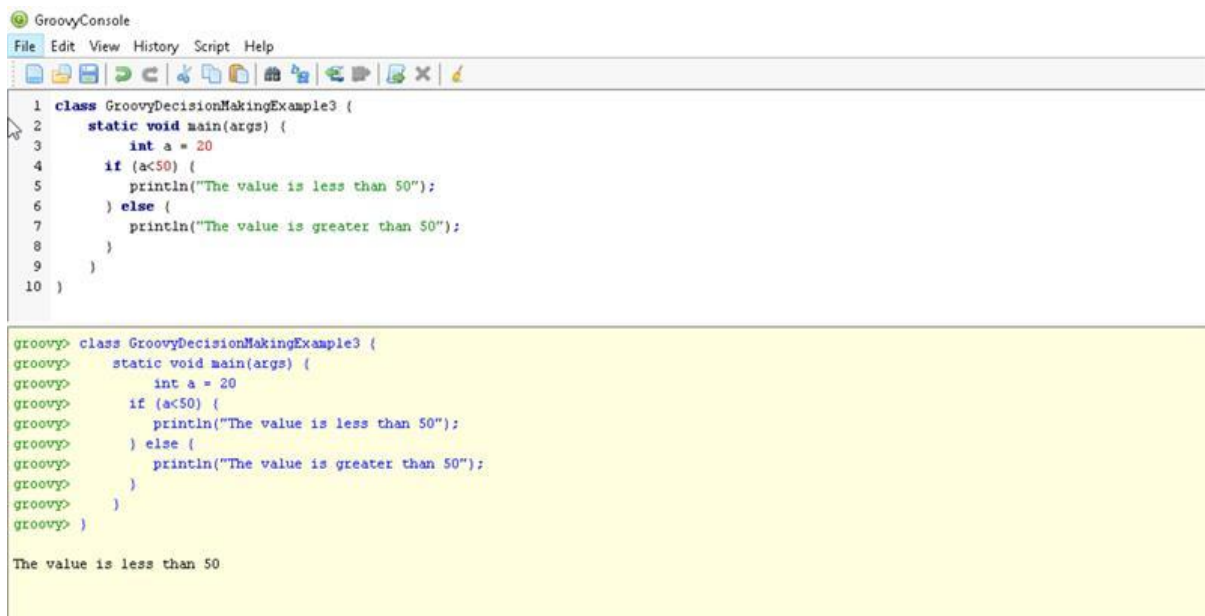
The screenshot shows the GroovyConsole application. The top menu bar includes File, Edit, View, History, Script, and Help. Below the menu is a toolbar with various icons. The main text area contains the following Groovy code:

```
1 class GroovyDecisionMakingExample1 {
2
3     static void main(args) {
4         int a = 10
5         if (a<50) {
6             println("javatpoint");
7         }
8     }
9 }
10 }
```

Below the code editor, the console output is displayed on a yellow background. It shows the command prompt 'groovy>' followed by the class definition and the execution of the main method. The output is 'javatpoint'.

```
groovy> class GroovyDecisionMakingExample1 {
groovy>
groovy>     static void main(args) {
groovy>         int a = 10
groovy>         if (a<50) {
groovy>             println("javatpoint");
groovy>         }
groovy>     }
groovy> }
groovy>
javatpoint
```

2. IF- ELSE:



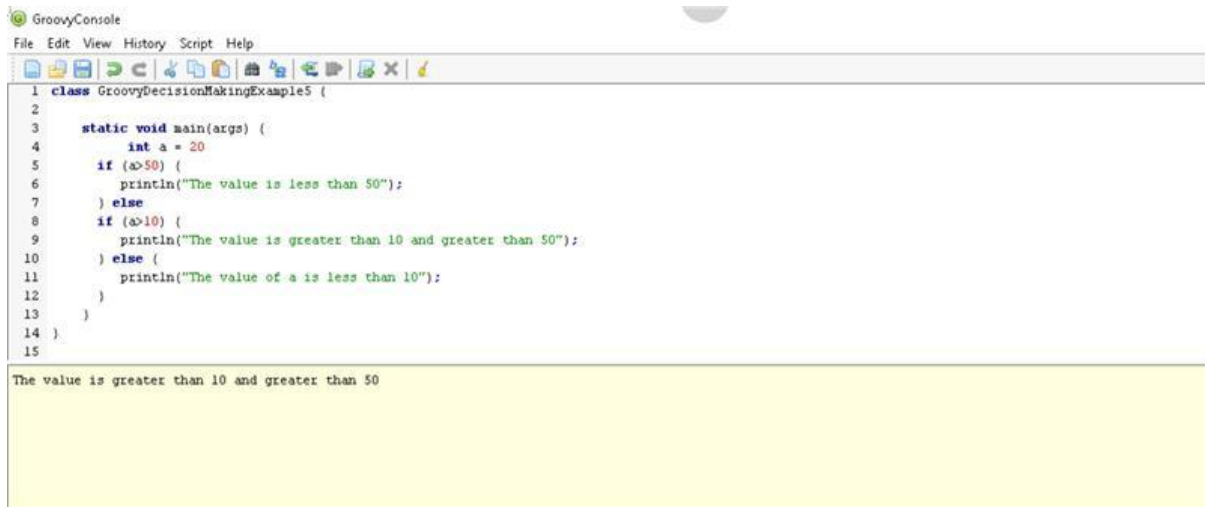
The screenshot shows the GroovyConsole application. The top menu bar includes File, Edit, View, History, Script, and Help. Below the menu is a toolbar with various icons. The main text area contains the following Groovy code:

```
1 class GroovyDecisionMakingExample3 {
2     static void main(args) {
3         int a = 20
4         if (a<50) {
5             println("The value is less than 50");
6         } else {
7             println("The value is greater than 50");
8         }
9     }
10 }
```

Below the code editor, the console output is displayed on a yellow background. It shows the command prompt 'groovy>' followed by the class definition and the execution of the main method. The output is 'The value is less than 50'.

```
groovy> class GroovyDecisionMakingExample3 {
groovy>     static void main(args) {
groovy>         int a = 20
groovy>         if (a<50) {
groovy>             println("The value is less than 50");
groovy>         } else {
groovy>             println("The value is greater than 50");
groovy>         }
groovy>     }
groovy> }
groovy>
The value is less than 50
```

3.NESTED – IF:

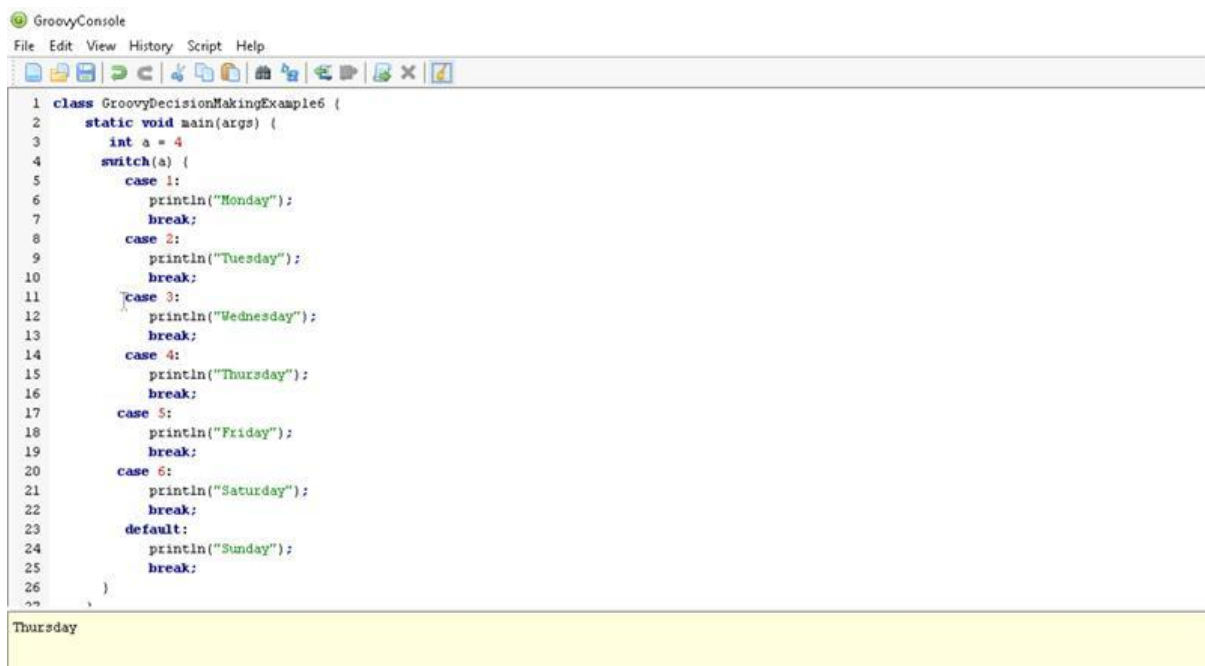


The screenshot shows the GroovyConsole application. The code defines a class `GroovyDecisionMakingExample5` with a `main` method. Inside `main`, an integer `a` is set to 20. A series of nested `if-else` statements check the value of `a`: first `a > 50`, then `a > 10`, and finally `a < 10`. Since `a` is 20, the condition `a > 10` is true, and the message "The value is greater than 10 and greater than 50" is printed. The output area at the bottom of the console shows this message.

```
1 class GroovyDecisionMakingExample5 {  
2  
3     static void main(args) {  
4         int a = 20  
5         if (a>50) {  
6             println("The value is less than 50");  
7         } else  
8         if (a>10) {  
9             println("The value is greater than 10 and greater than 50");  
10        } else {  
11            println("The value of a is less than 10");  
12        }  
13    }  
14 }  
15
```

The value is greater than 10 and greater than 50

4. SWITCH STATEMENTS:



The screenshot shows the GroovyConsole application. The code defines a class `GroovyDecisionMakingExample6` with a `main` method. Inside `main`, an integer `a` is set to 4. A `switch` statement is used to check the value of `a` against cases 1 through 6, each corresponding to a day of the week. Since `a` is 4, the case for "Thursday" is selected and printed. The output area at the bottom of the console shows "Thursday".

```
1 class GroovyDecisionMakingExample6 {  
2     static void main(args) {  
3         int a = 4  
4         switch(a) {  
5             case 1:  
6                 println("Monday");  
7                 break;  
8             case 2:  
9                 println("Tuesday");  
10                break;  
11            case 3:  
12                println("Wednesday");  
13                break;  
14            case 4:  
15                println("Thursday");  
16                break;  
17            case 5:  
18                println("Friday");  
19                break;  
20            case 6:  
21                println("Saturday");  
22                break;  
23            default:  
24                println("Sunday");  
25                break;  
26        }  
27    }  
28 }
```

Thursday

LOG FILE ANALYSER:

GroovyConsole

File Edit View History Script Help



```
1 import java.nio.file.*
2
3 def logFilePath = "C:/Users/Administrator/Documents/demologfile.txt"
4 def logFile = new File(logFilePath)
5 if (!logFile.exists()) {
6     println "Error: Log file not found at $logFilePath"
7     return
8 }
9 def logData = [:].withDefault { 0 }
10
11 logFile.eachLine { line ->
12     def matcher = line =~ /\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}\s+(INFO|WARNING|ERROR)\s+(.+)/
13     if (matcher.find()) {
14         def timestamp = matcher[0][1]
15         def statusCode = matcher[0][2]
16         def message = matcher[0][3]
17         logData[statusCode]++ // Count occurrences
18         println "[${timestamp}] Status: ${statusCode} - ${message}"
19     }
20 }
21 println "\nLog Analysis Summary:"
22 if (logData.isEmpty()) {
23     println "No matching log entries found."
24 } else {
25     logData.each { key, value -> println "$key : $value occurrences" }
26 }
27
```

```
[2025-02-17 10:15:32] Status: INFO - Server started successfully.
[2025-02-17 10:16:45] Status: WARNING - High memory usage detected.
[2025-02-17 10:17:12] Status: ERROR - Failed to connect to database.
[2025-02-17 10:20:05] Status: INFO - User 'admin' logged in from IP 192.168.1.100.
[2025-02-17 10:22:30] Status: ERROR - Timeout occurred while processing request.
[2025-02-17 10:25:50] Status: INFO - Scheduled job 'backup' started.
[2025-02-17 10:30:15] Status: WARNING - Disk space running low on /var/log.
[2025-02-17 10:35:40] Status: ERROR - File system corruption detected.
[2025-02-17 10:40:10] Status: INFO - User 'guest' logged out.
[2025-02-17 10:45:25] Status: INFO - Server shutting down for maintenance.
```

Log Analysis Summary:

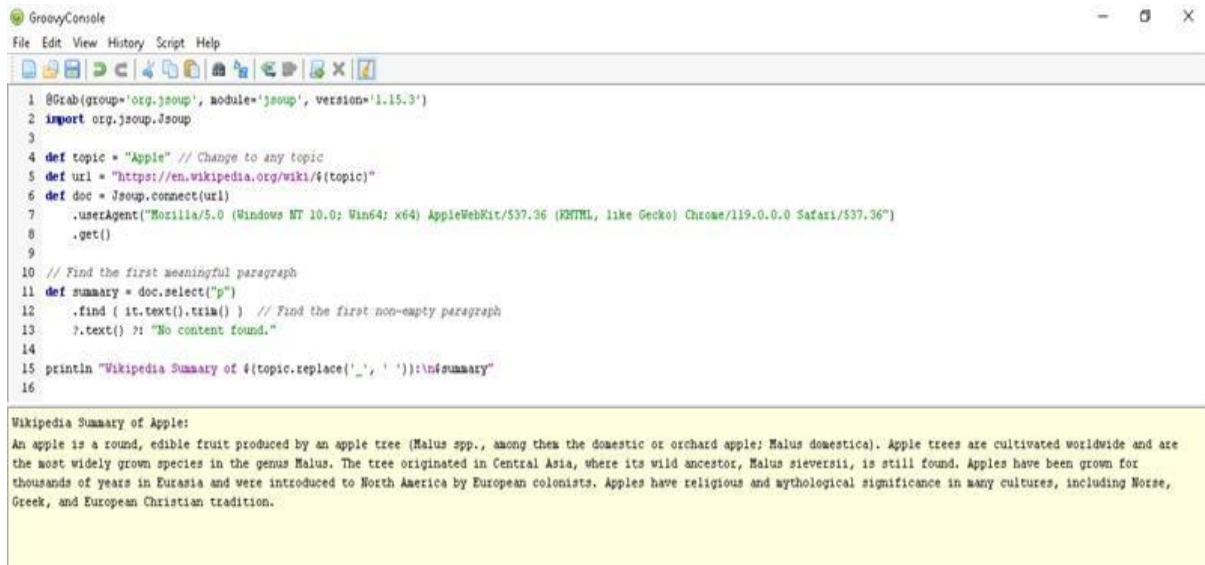
INFO : 5 occurrences

WARNING : 2 occurrences

ERROR : 3 occurrences

Result: [INFO:5, WARNING:2, ERROR:3]

WEB SCRAPING WITH GROOVY AND JSOUP:



The screenshot shows the GroovyConsole application window. The title bar reads "GroovyConsole". The menu bar includes "File", "Edit", "View", "History", "Script", and "Help". The toolbar contains icons for file operations and execution. The script area contains the following code:

```
1 @Grab(group='org.jsoup', module='jsoup', version='1.15.3')
2 import org.jsoup.Jsoup
3
4 def topic = "Apple" // Change to any topic
5 def url = "https://en.wikipedia.org/wiki/${topic}"
6 def doc = Jsoup.connect(url)
7   .userAgent("Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36")
8   .get()
9
10 // Find the first meaningful paragraph
11 def summary = doc.select("p")
12   .find { it.text().trim() } // Find the first non-empty paragraph
13   ?.text() ?: "No content found."
14
15 println "Wikipedia Summary of ${topic.replace('_', ' ')}:\n${summary}"
16
```

The output area displays the result of the script execution:

Wikipedia Summary of Apple:

An apple is a round, edible fruit produced by an apple tree (*Malus* spp., among them the domestic or orchard apple; *Malus domestica*). Apple trees are cultivated worldwide and are the most widely grown species in the genus *Malus*. The tree originated in Central Asia, where its wild ancestor, *Malus sieversii*, is still found. Apples have been grown for thousands of years in Eurasia and were introduced to North America by European colonists. Apples have religious and mythological significance in many cultures, including Norse, Greek, and European Christian tradition.