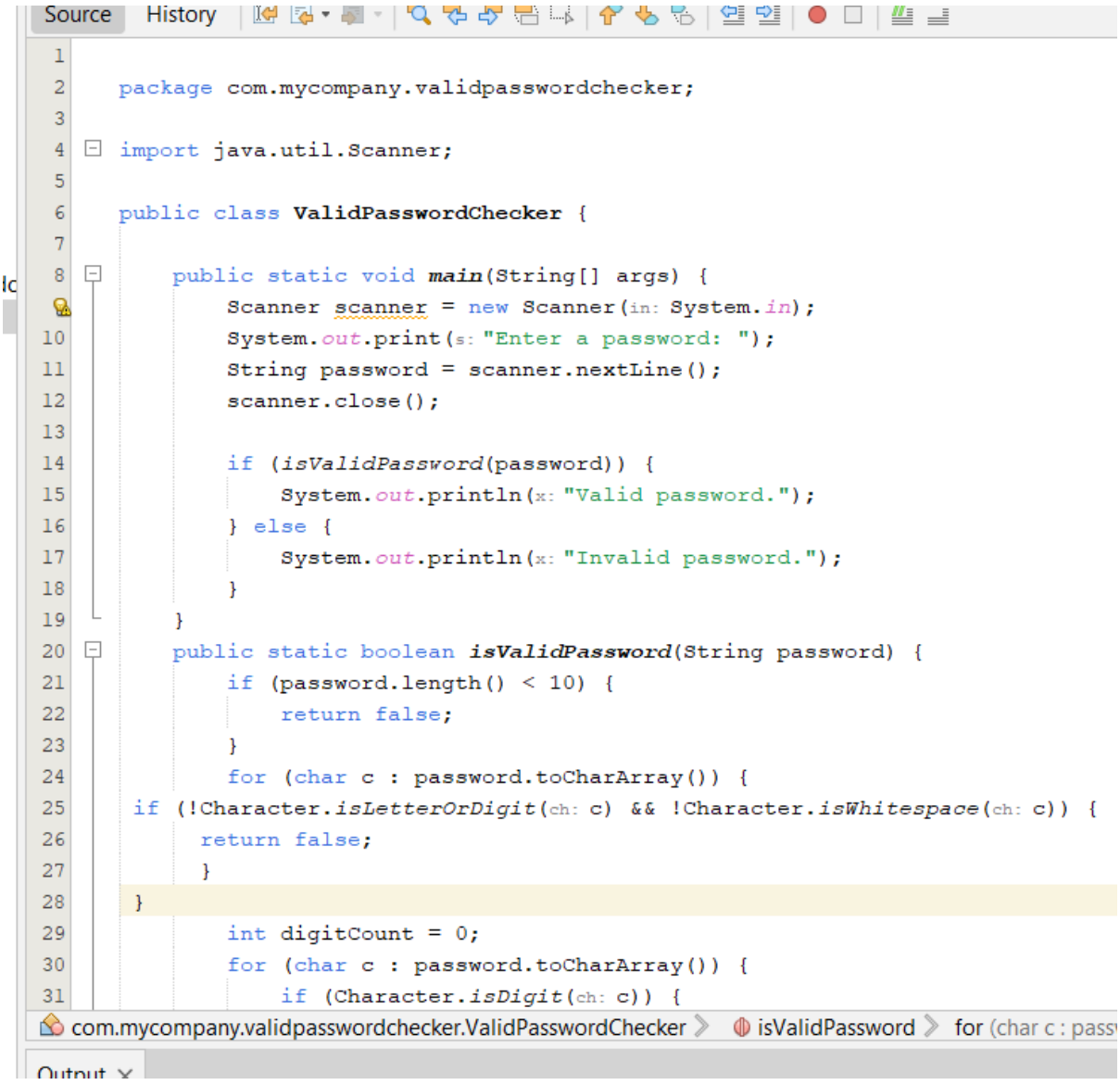
First Task: Write a Java program that checks whether a string is a valid password. Suppose the password rule is as follows:

- A password must have at least ten characters.
- A password consists of only letters and digits.
- A password must contain at least two digits.
- A password must contain a whitespace.

(*You have to take input of the* password *from the user*).

Source Code:



```
Source  History

1
2   package com.mycompany.validpasswordchecker;
3
4   import java.util.Scanner;
5
6   public class ValidPasswordChecker {
7
8       public static void main(String[] args) {
9           Scanner scanner = new Scanner(in: System.in);
10          System.out.print(s: "Enter a password: ");
11          String password = scanner.nextLine();
12          scanner.close();
13
14          if (isValidPassword(password)) {
15              System.out.println(x: "Valid password.");
16          } else {
17              System.out.println(x: "Invalid password.");
18          }
19      }
20      public static boolean isValidPassword(String password) {
21          if (password.length() < 10) {
22              return false;
23          }
24          for (char c : password.toCharArray()) {
25  if (!Character.isLetterOrDigit(ch: c) && !Character.isWhitespace(ch: c)) {
26        return false;
27        }
28  }
29          int digitCount = 0;
30          for (char c : password.toCharArray()) {
31              if (Character.isDigit(ch: c)) {

com.mycompany.validpasswordchecker.ValidPasswordChecker > isValidPassword > for (char c : pass

Output
```

```
Source  History
22                  return false;
23              }
24              for (char c : password.toCharArray()) {
25      if (!Character.isLetterOrDigit(ch: c) && !Character.isWhitespace(ch: c)) {
26          return false;
27          }
28      }
29              int digitCount = 0;
30              for (char c : password.toCharArray()) {
31                  if (Character.isDigit(ch: c)) {
32                      digitCount++;
33                      if (digitCount >= 2) {
34                          break;
35                      }
36                  }
37              }
38              if (digitCount < 2) {
39                  return false;
40              }
42              if (!password.contains(s: " ")) {
43                  return false;
44              }
45              return true;
46          }
47      }
48
49
```

com.mycompany.validpasswordchecker.ValidPasswordChecker > isValidPassword > for (char c : passw

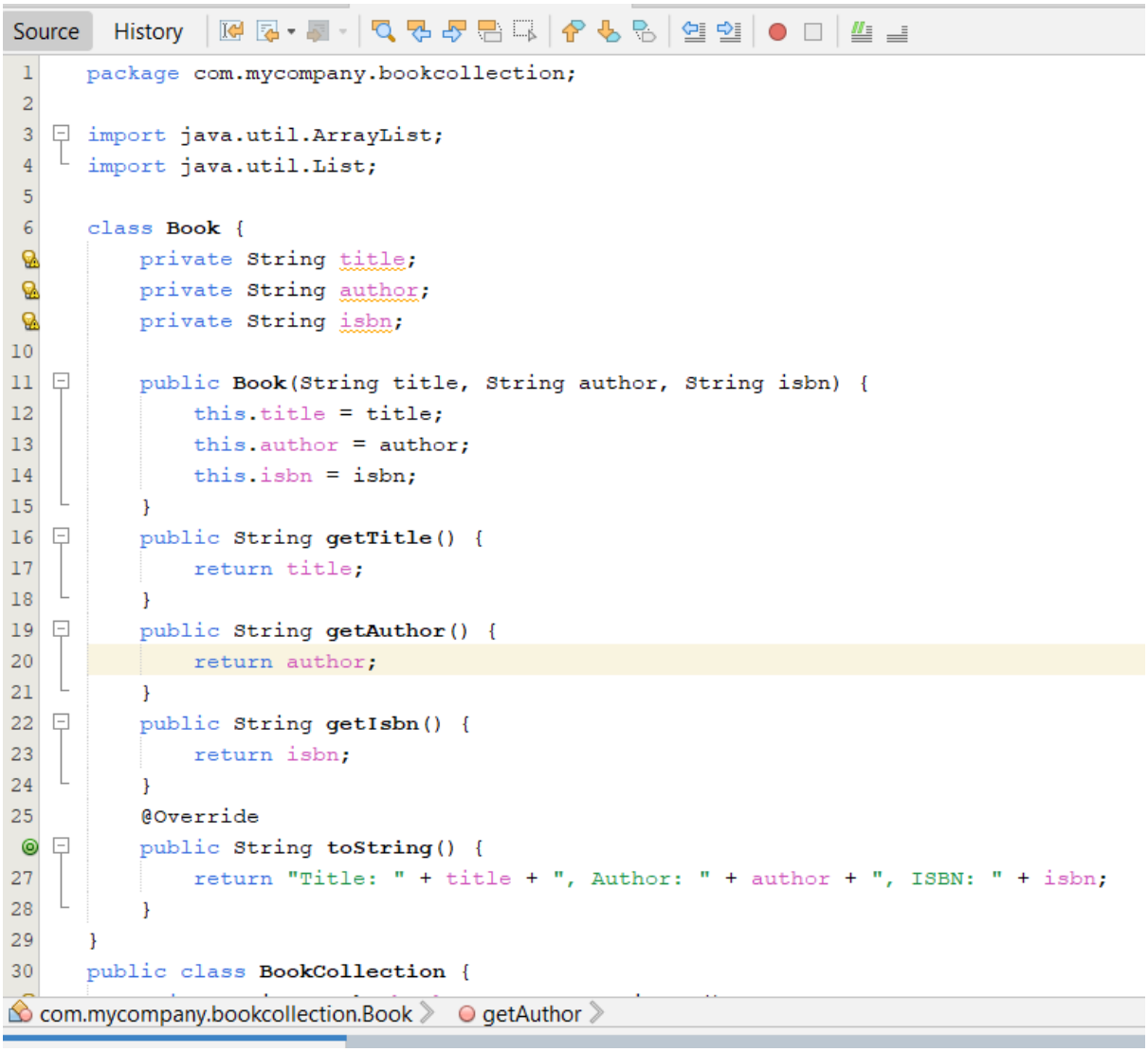
Output:

```
--- resources:3.3.0:resources (default-resources) @ ValidPasswordChecker ---
skip non existing resourceDirectory C:\Users\HP\OneDrive\Documents\NetBeansProjects\ValidPasswordChecker\src\main\resources

--- compiler:3.10.1:compile (default-compile) @ ValidPasswordChecker ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ ValidPasswordChecker ---
Enter a password: Javaoop lab124
Valid password.
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  11.539 s
Finished at: 2023-09-10T22:41:32+06:00
------------------------------------------------------------------------
```

Discussion: Here is a Java program that checks whether a given string is a valid password or not. This program takes input from the user and checks whether the entered password meets the specified rules. It uses a combination of loops and conditional statements to validate the password according to the given criteria.

Second Task: Write a Java program to create a class called "Book" with attributes for title, author, and ISBN, and methods to add and remove books from a collection.

Source Code:



```java
public class BookCollection {
    private List<Book> books = new ArrayList<>();
    public void addBook(Book book) {
        books.add(e: book);
    }
    public void removeBook(Book book) {
        books.remove(o: book);
    }
    public List<Book> getBooks() {
        return books;
    }
    public static void main(String[] args) {
        BookCollection collection = new BookCollection();
collection.addBook(new Book(title: "The Cow", author: "Gorge Henri Cow", isbn: "027-147389"));
collection.addBook(new Book(title: "Hamba Hamba", author: "Robert H. Hamba", isbn: "027-873629"));
collection.addBook(new Book(title: "Siuuu", author: "Christiano Ronaldo", isbn: "027-753895"));
        List<Book> books = collection.getBooks();
        System.out.println(x: "Books in the collection:");
        for (Book book : books) {
            System.out.println(x: book);
        }
        Book bookToRemove = new Book(title: "The Cow", author: "Gorge Henri Cow", isbn: "027-14738");
        collection.removeBook(book: bookToRemove);

        books = collection.getBooks();
        System.out.println(x: "\nBooks in the updated collection:");
        for (Book book : books) {
            System.out.println(x: book);
        }
    }
}
```

Output:

```
Output - Run (BookCollection)
----------------------< com.mycompany:BookCollection >----------------------
Building BookCollection 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.0:resources (default-resources) @ BookCollection ---
skip non existing resourceDirectory C:\Users\HP\OneDrive\Documents\NetBeansPr

--- compiler:3.10.1:compile (default-compile) @ BookCollection ---
Changes detected - recompiling the module!
Compiling 1 source file to C:\Users\HP\OneDrive\Documents\NetBeansProjects\Bc

--- exec:3.1.0:exec (default-cli) @ BookCollection ---
Books in the collection:
Title: The Cow, Author: Gorge Henri Cow, ISBN: 027-147389
Title: Hamba Hamba, Author: Robert H. Hamba, ISBN: 027-873629
Title: Siuuu, Author: Christiano Ronaldo, ISBN: 027-753895

Books in the updated collection:
Title: The Cow, Author: Gorge Henri Cow, ISBN: 027-147389
Title: Hamba Hamba, Author: Robert H. Hamba, ISBN: 027-873629
Title: Siuuu, Author: Christiano Ronaldo, ISBN: 027-753895
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.130 s
```

Discussion: Here's a Java program that defines a Book class with attributes for title, author, and ISBN, and includes methods to add and remove books from a collection. In this program, we have a Book class with attributes for title, author, and ISBN, along with getters to access these attributes. The BookCollection class manages a collection of books using a List<Book>. It provides methods to add and remove books from the collection. Finally, in the main method, we demonstrate how to create a collection, add books to it, remove a book, and display the contents of the collection.