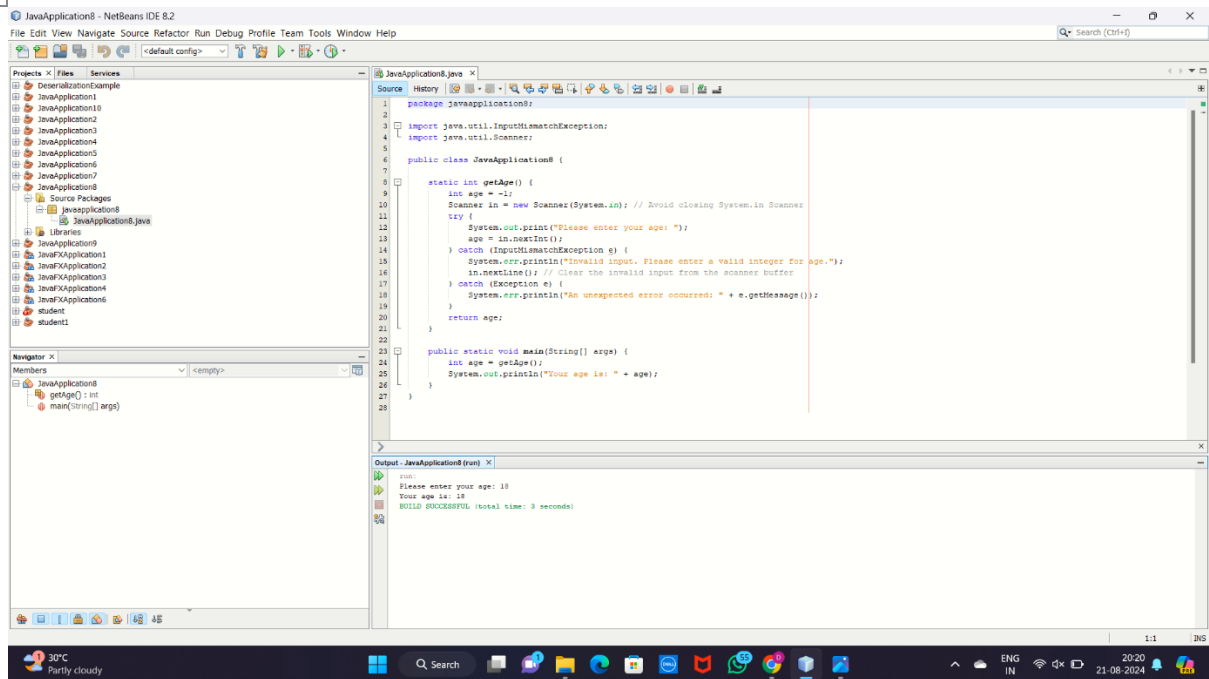


## 1. LINE READING METHOD USING NUMBER IN JAVA



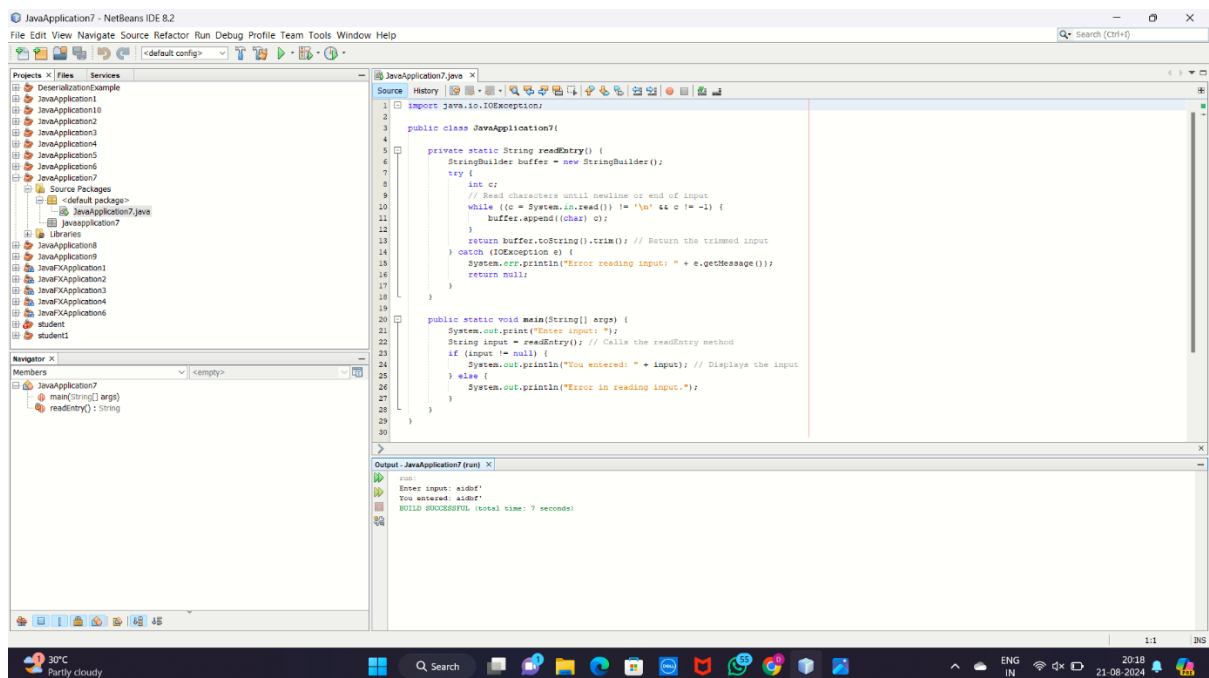
The screenshot shows the NetBeans IDE interface for a project named 'JavaApplication8'. The 'Source' window displays the code for 'JavaApplication8.java'. The code uses the 'Scanner' class to read input from the user. The 'Output' window shows the program's execution, where the user enters '18' and the program prints 'Your age is: 18'.

```
1 package javaapplication8;
2
3 import java.util.InputMismatchException;
4 import java.util.Scanner;
5
6 public class JavaApplication8 {
7
8     static int getAge() {
9         int age = -1;
10        Scanner in = new Scanner(System.in); // Avoid closing System.in Scanner
11        try {
12            System.out.print("Please enter your age: ");
13            age = in.nextInt();
14        } catch (InputMismatchException e) {
15            System.err.println("Invalid input. Please enter a valid integer for age.");
16            in.nextLine(); // Clear the invalid input from the scanner buffer
17        } catch (Exception e) {
18            System.err.println("An unexpected error occurred: " + e.getMessage());
19        }
20        return age;
21    }
22
23    public static void main(String[] args) {
24        int age = getAge();
25        System.out.println("Your age is: " + age);
26    }
27
28 }
```

Output - JavaApplication8 (run) x

```
run
Please enter your age: 18
Your age is: 18
BUILD SUCCESSFUL (total time: 3 seconds)
```

## 2. LINE READER METHOD USING ASCII.JAVA



The screenshot shows the NetBeans IDE interface for a project named 'JavaApplication7'. The 'Source' window displays the code for 'JavaApplication7.java'. The code uses the 'BufferedReader' class to read input from the user. The 'Output' window shows the program's execution, where the user enters 'a1d1e' and the program prints 'You entered: a1d1e'.

```
1 import java.io.IOException;
2
3 public class JavaApplication7 {
4
5     private static String readEntry() {
6         StringBuilder buffer = new StringBuilder();
7         try {
8             int c;
9             // Read characters until newline or end of input
10            while ((c = System.in.read()) != '\n' && c != -1) {
11                buffer.append((char) c);
12            }
13            return buffer.toString().trim(); // Return the trimmed input
14        } catch (IOException e) {
15            System.err.println("Error reading input: " + e.getMessage());
16            return null;
17        }
18    }
19
20    public static void main(String[] args) {
21        System.out.print("Enter input: ");
22        String input = readEntry(); // Calls the readEntry method
23        if (input != null) {
24            System.out.println("You entered: " + input); // Displays the input
25        } else {
26            System.out.println("Error in reading input.");
27        }
28    }
29
30 }
```

Output - JavaApplication7 (run) x

```
run
Enter input: a1d1e
You entered: a1d1e
BUILD SUCCESSFUL (total time: 3 seconds)
```