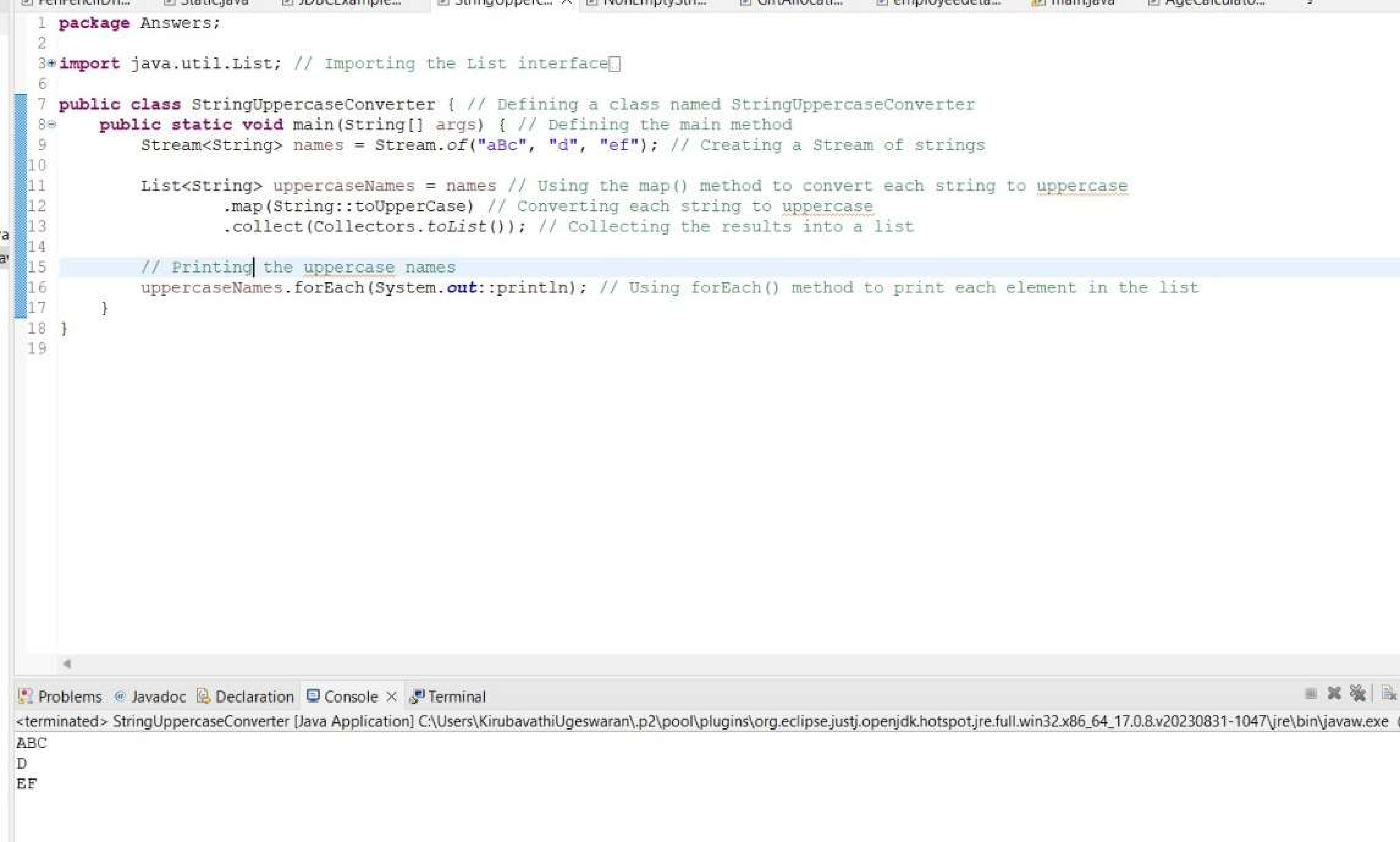
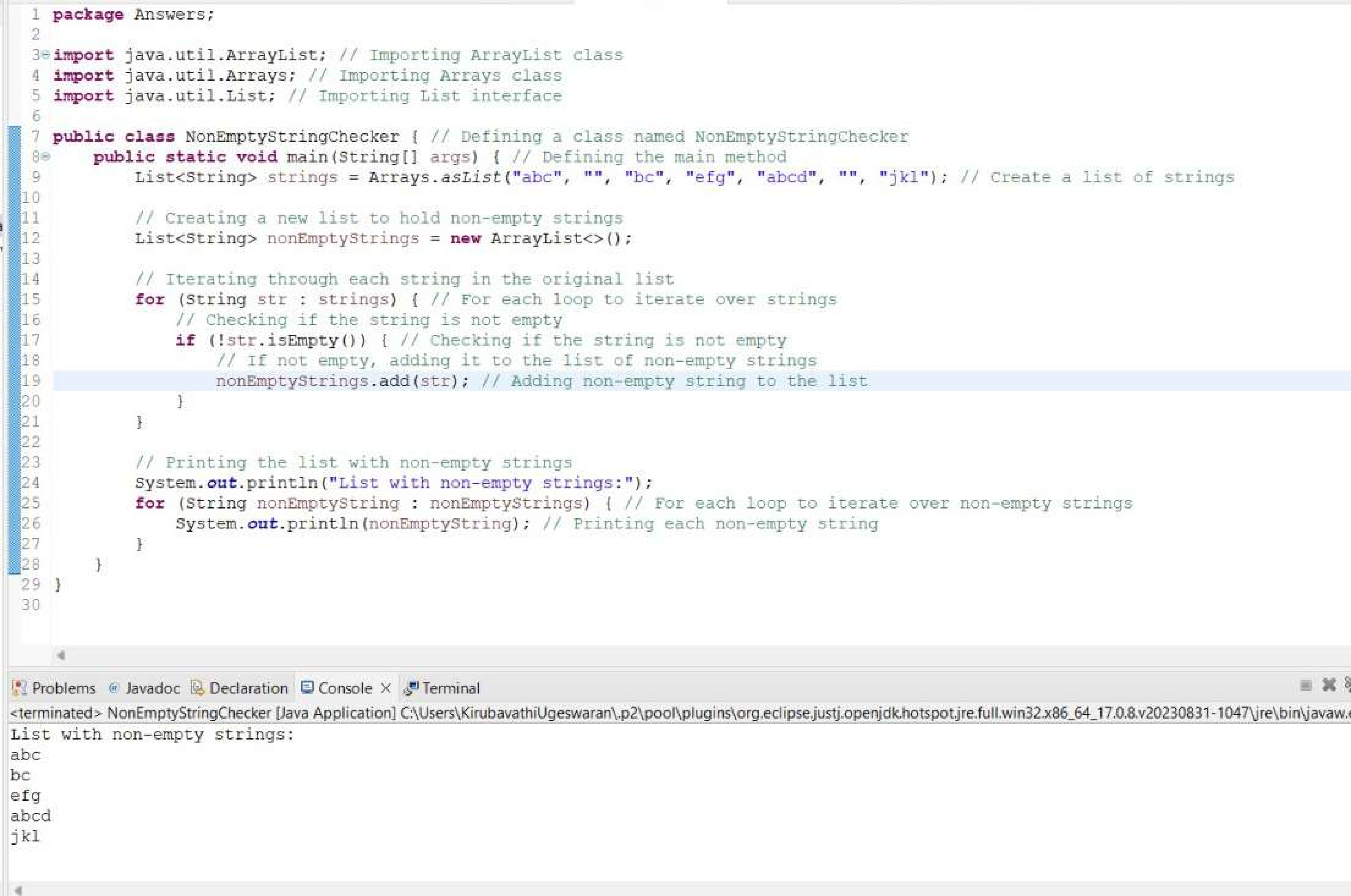


```java
package Answers;

import java.util.List; // Importing the List interface

public class StringUppercaseConverter { // Defining a class named StringUppercaseConverter
    public static void main(String[] args) { // Defining the main method
        Stream<String> names = Stream.of("aBc", "d", "ef"); // Creating a Stream of strings

        List<String> uppercaseNames = names // Using the map() method to convert each string to uppercase
                .map(String::toUpperCase) // Converting each string to uppercase
                .collect(Collectors.toList()); // Collecting the results into a list

        // Printing the uppercase names
        uppercaseNames.forEach(System.out::println); // Using forEach() method to print each element in the list
    }
}
```

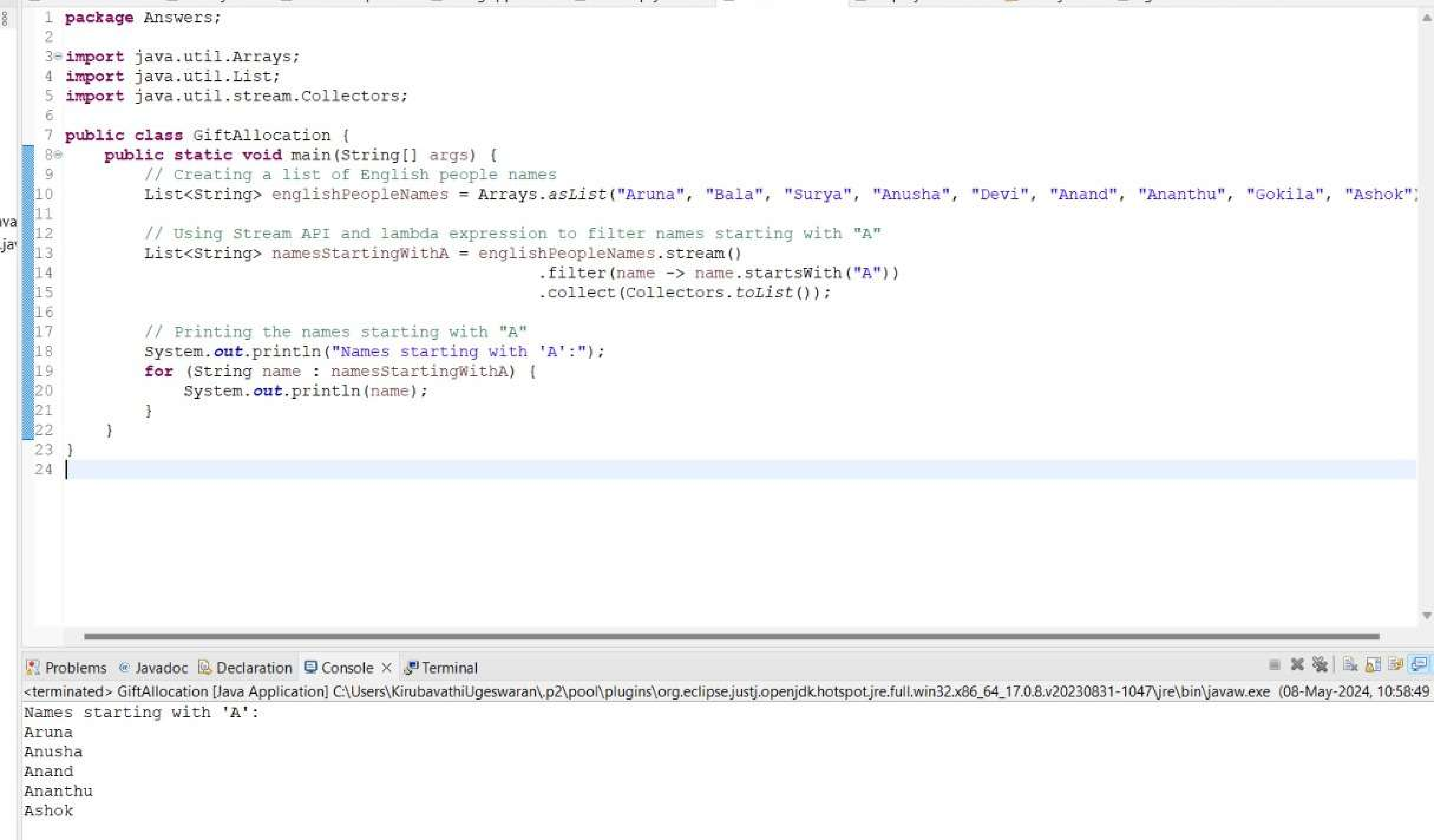
<terminated> StringUppercaseConverter [Java Application] C:\Users\KirubavathiUgeswaran\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.exe

```
ABC
D
EF
```



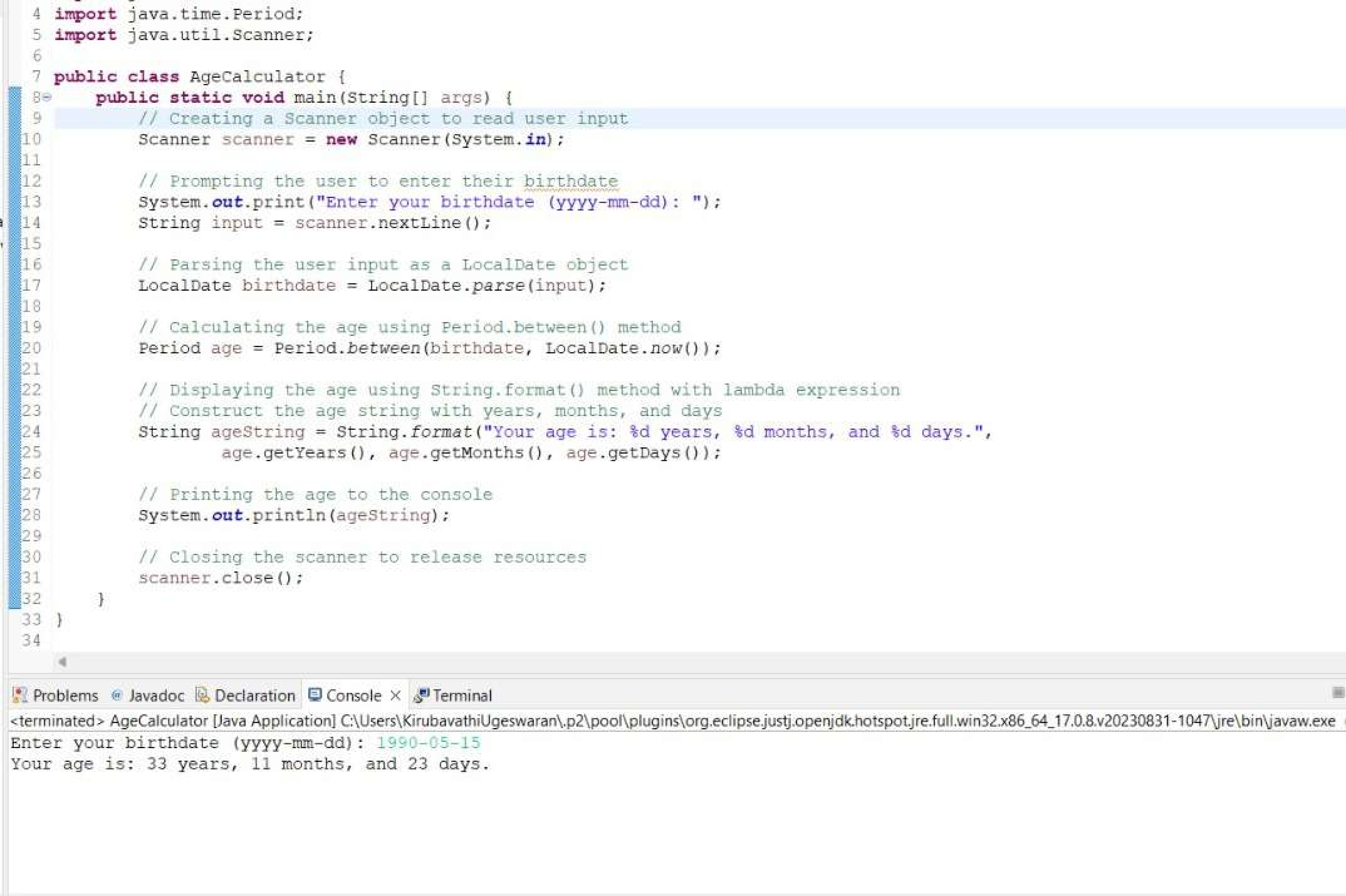
```java
package Answers;

import java.util.ArrayList; // Importing ArrayList class
import java.util.Arrays; // Importing Arrays class
import java.util.List; // Importing List interface

public class NonEmptyStringChecker { // Defining a class named NonEmptyStringChecker
    public static void main(String[] args) { // Defining the main method
        List<String> strings = Arrays.asList("abc", "", "bc", "efg", "abcd", "", "jkl"); // Create a list of strings

        // Creating a new list to hold non-empty strings
        List<String> nonEmptyStrings = new ArrayList<>();

        // Iterating through each string in the original list
        for (String str : strings) { // For each loop to iterate over strings
            // Checking if the string is not empty
            if (!str.isEmpty()) { // Checking if the string is not empty
                // If not empty, adding it to the list of non-empty strings
                nonEmptyStrings.add(str); // Adding non-empty string to the list
            }
        }

        // Printing the list with non-empty strings
        System.out.println("List with non-empty strings:");
        for (String nonEmptyString : nonEmptyStrings) { // For each loop to iterate over non-empty strings
            System.out.println(nonEmptyString); // Printing each non-empty string
        }
    }
}
```

```
<terminated> NonEmptyStringChecker [Java Application] C:\Users\KirubavathiUgeswaran\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.
List with non-empty strings:
abc
bc
efg
abcd
jkl
```



```java
package Answers;

import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class GiftAllocation {
    public static void main(String[] args) {
        // Creating a list of English people names
        List<String> englishPeopleNames = Arrays.asList("Aruna", "Bala", "Surya", "Anusha", "Devi", "Anand", "Ananthu", "Gokila", "Ashok")

        // Using Stream API and lambda expression to filter names starting with "A"
        List<String> namesStartingWithA = englishPeopleNames.stream()
                                                .filter(name -> name.startsWith("A"))
                                                .collect(Collectors.toList());

        // Printing the names starting with "A"
        System.out.println("Names starting with 'A':");
        for (String name : namesStartingWithA) {
            System.out.println(name);
        }
    }
}

```

```
<terminated> GiftAllocation [Java Application] C:\Users\KirubavathiUgeswaran\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.exe  (08-May-2024, 10:58:49
Names starting with 'A':
Aruna
Anusha
Anand
Ananthu
Ashok
```



```java
import java.time.Period;
import java.util.Scanner;

public class AgeCalculator {
    public static void main(String[] args) {
        // Creating a Scanner object to read user input
        Scanner scanner = new Scanner(System.in);

        // Prompting the user to enter their birthdate
        System.out.print("Enter your birthdate (yyyy-mm-dd): ");
        String input = scanner.nextLine();

        // Parsing the user input as a LocalDate object
        LocalDate birthdate = LocalDate.parse(input);

        // Calculating the age using Period.between() method
        Period age = Period.between(birthdate, LocalDate.now());

        // Displaying the age using String.format() method with lambda expression
        // Construct the age string with years, months, and days
        String ageString = String.format("Your age is: %d years, %d months, and %d days.",
                age.getYears(), age.getMonths(), age.getDays());

        // Printing the age to the console
        System.out.println(ageString);

        // Closing the scanner to release resources
        scanner.close();
    }
}
```

<terminated> AgeCalculator [Java Application] C:\Users\KirubavathiUgeswaran\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.exe

```
Enter your birthdate (yyyy-mm-dd): 1990-05-15
Your age is: 33 years, 11 months, and 23 days.
```