

```
J ANameFinder.java X
J ANameFinder.java > ...
1 import java.util.*;
2 import java.util.stream.*;
3
4 public class ANameFinder {
5     public static void main(String[] args) {
6         List<String> names = Arrays.asList("Anu", "Bhanu", "Charitha", "Shyam", "Arjun");
7
8         List<String> aNames = names.stream()
9                                 .filter(name -> name.startsWith("A"))
10                                .collect(Collectors.toList());
11
12         aNames.forEach(System.out::println);
13     }
14 }
15
```

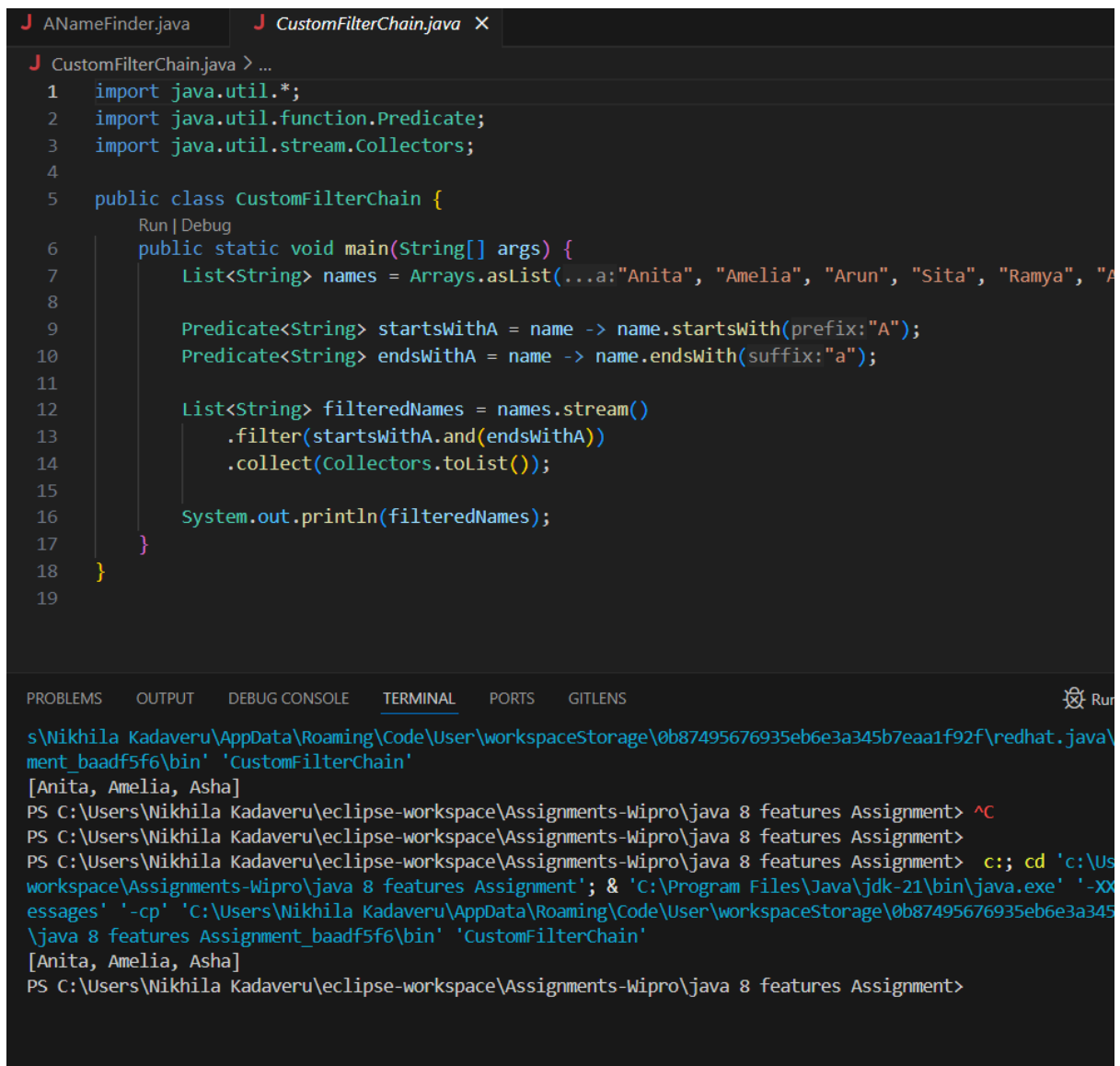
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

Run: ANameFinder + - [ ] ... [ ] X

PS C:\Users\Nikhila Kadaveru\workspace\Assignments-wipro\java 8 features Assignment> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\workspaceStorage\0b87495676935eb6e3a345b7ea1f92f\redhat.java\jdt\_ws\java 8 features Assignment\_baadf5f6\bin' 'ANameFinder'

Anu  
Arjun  
PS C:\Users\Nikhila Kadaveru\workspace\Assignments-wipro\java 8 features Assignment>

Figure 1 ANameFinder



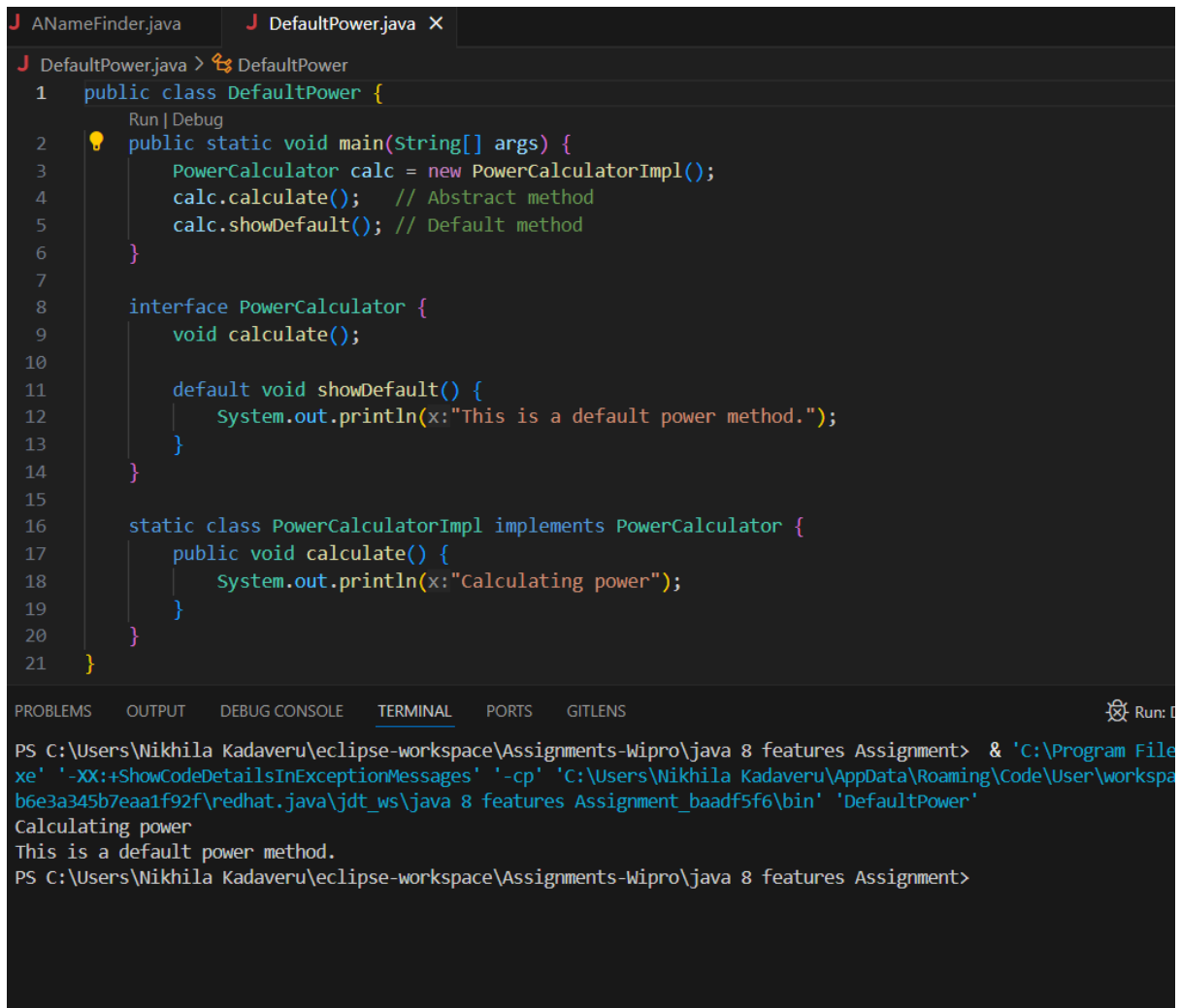
The screenshot displays the Eclipse IDE interface. The top editor shows the `CustomFilterChain.java` file with the following code:

```
1 import java.util.*;
2 import java.util.function.Predicate;
3 import java.util.stream.Collectors;
4
5 public class CustomFilterChain {
6     public static void main(String[] args) {
7         List<String> names = Arrays.asList("Anita", "Amelia", "Arun", "Sita", "Ramya", "Asha");
8
9         Predicate<String> startsWithA = name -> name.startsWith("A");
10        Predicate<String> endsWithA = name -> name.endsWith("a");
11
12        List<String> filteredNames = names.stream()
13            .filter(startsWithA.and(endsWithA))
14            .collect(Collectors.toList());
15
16        System.out.println(filteredNames);
17    }
18 }
19
```

The bottom panel shows the **TERMINAL** view with the following output:

```
s\Nikhila Kadaveru\AppData\Roaming\Code\User\workspaceStorage\0b87495676935eb6e3a345b7eaa1f92f\redhat.java\
ment_baadf5f6\bin' 'CustomFilterChain'
[Anita, Amelia, Asha]
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment> ^C
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment>
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment> c.;; cd 'C:\Us
workspace\Assignments-Wipro\java 8 features Assignment'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX
essages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\workspaceStorage\0b87495676935eb6e3a345
\java 8 features Assignment_baadf5f6\bin' 'CustomFilterChain'
[Anita, Amelia, Asha]
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment>
```

Figure 1 CustomFilterChain



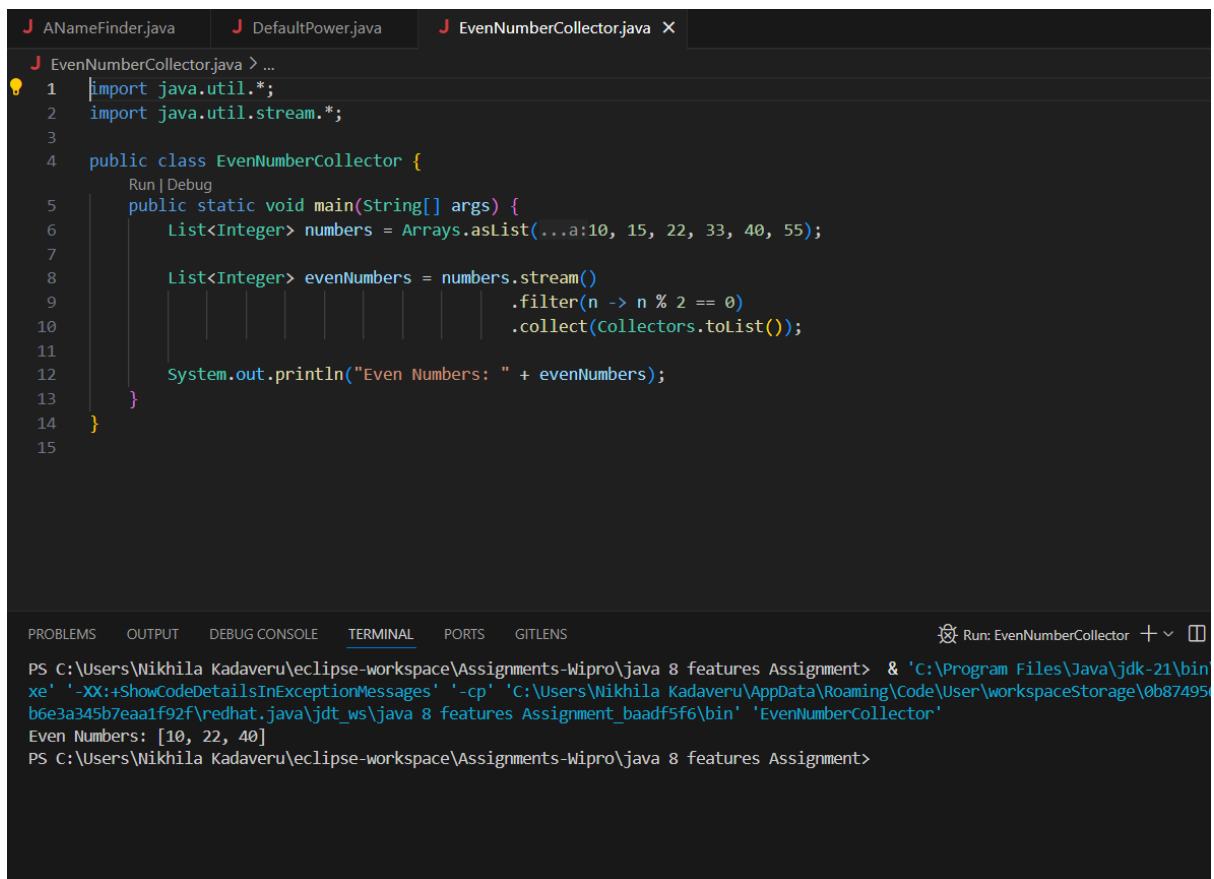
The screenshot displays the Eclipse IDE interface. The top editor shows the `DefaultPower.java` file with the following code:

```
1 public class DefaultPower {  
2     public static void main(String[] args) {  
3         PowerCalculator calc = new PowerCalculatorImpl();  
4         calc.calculate(); // Abstract method  
5         calc.showDefault(); // Default method  
6     }  
7  
8     interface PowerCalculator {  
9         void calculate();  
10  
11         default void showDefault() {  
12             System.out.println(x:"This is a default power method.");  
13         }  
14     }  
15  
16     static class PowerCalculatorImpl implements PowerCalculator {  
17         public void calculate() {  
18             System.out.println(x:"Calculating power");  
19         }  
20     }  
21 }
```

The bottom panel shows the **TERMINAL** view with the following output:

```
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment> & 'C:\Program File  
xe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\workspa  
b6e3a345b7eaa1f92f\redhat.java\jdt_ws\java 8 features Assignment_baadf5f6\bin' 'DefaultPower'  
Calculating power  
This is a default power method.  
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment>
```

Figure 3 DefaultPower



The screenshot displays the Eclipse IDE interface. At the top, three tabs are visible: 'ANameFinder.java', 'DefaultPower.java', and 'EvenNumberCollector.java'. The 'EvenNumberCollector.java' tab is active, showing the following Java code:

```
1 import java.util.*;
2 import java.util.stream.*;
3
4 public class EvenNumberCollector {
5     public static void main(String[] args) {
6         List<Integer> numbers = Arrays.asList(10, 15, 22, 33, 40, 55);
7
8         List<Integer> evenNumbers = numbers.stream()
9             .filter(n -> n % 2 == 0)
10            .collect(Collectors.toList());
11
12        System.out.println("Even Numbers: " + evenNumbers);
13    }
14 }
15
```

Below the code editor, the 'TERMINAL' tab is selected, showing the command prompt output:

```
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\workspaceStorage\0b87495b6e3a345b7eaa1f92f\redhat.java\jdt_ws\java 8 features Assignment_baadf5f6\bin' 'EvenNumberCollector'
Even Numbers: [10, 22, 40]
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment>
```

Figure 4 EvenNumberCollector

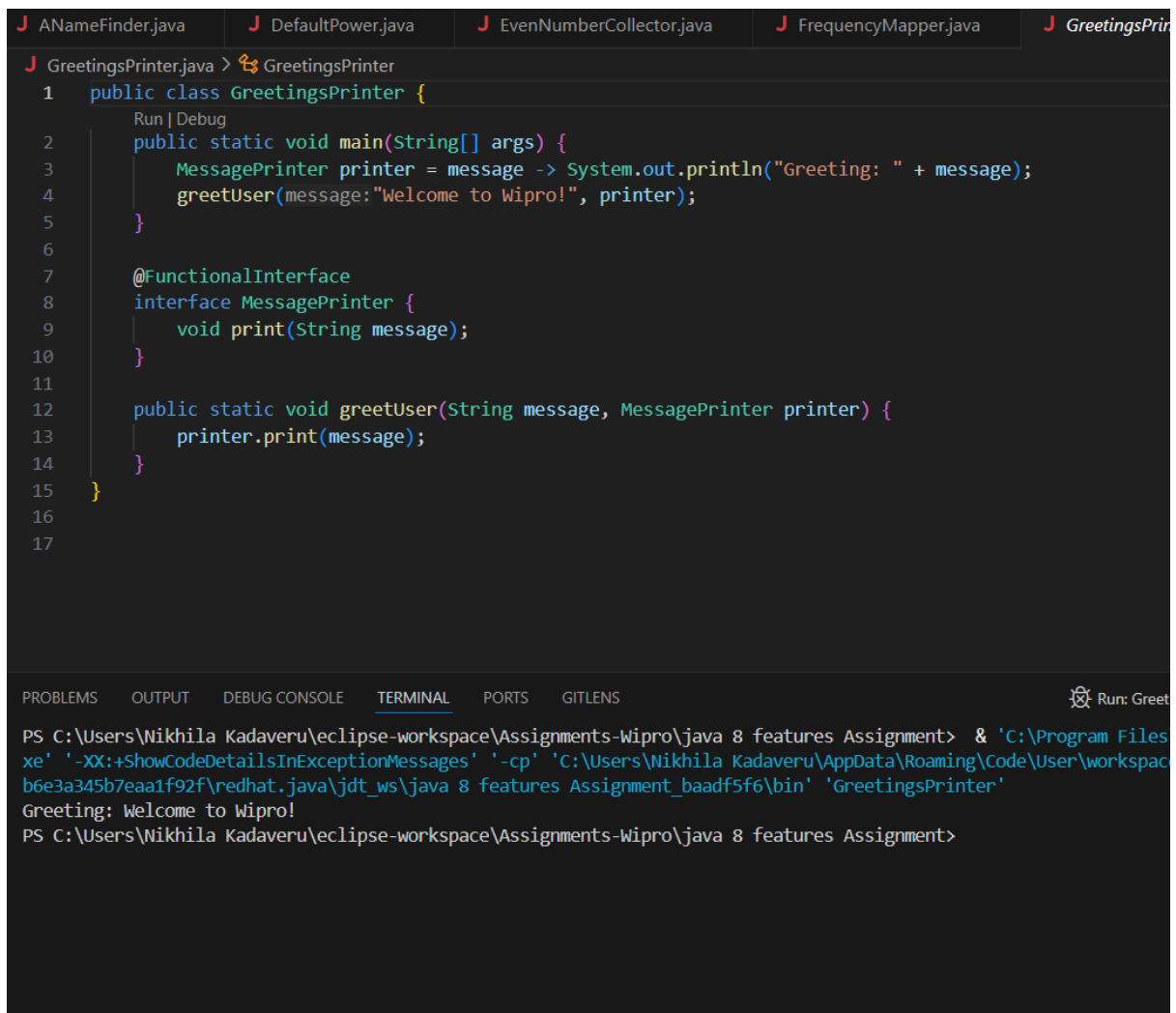
The screenshot shows the Eclipse IDE with the `FrequencyMapper.java` file open. The code uses Java 8 streams to count the frequency of names in an array. The terminal output shows the command to run the program and the resulting map of name frequencies.

```
C:\Users\Nikhila Kadaveru\workspace\Assignments-Wipro\java 8 features
Assignment\ANameFinder.java
2  import java.util.stream.Collectors;
3
4  public class FrequencyMapper {
5      Run | Debug
6      public static void main(String[] args) {
7          List<String> names = Arrays.asList(...a:"Alice", "Bob", "Alice", "David", "Bob", "Alice
8
9          Map<String, Long> nameCountMap = names.stream()
10             .collect(Collectors.groupingBy(name -> name, Collectors.counting()));
11
12         System.out.println(nameCountMap);
13     }
14 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Nikhila Kadaveru\workspace\Assignments-Wipro\java 8 features Assignment> & 'C:\Program F  
xe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\work  
b6e3a345b7eaa1f92f\redhat.java\jdt\_ws\java 8 features Assignment\_baadf5f6\bin' 'FrequencyMapper'  
{Bob=2, Alice=3, David=1}  
PS C:\Users\Nikhila Kadaveru\workspace\Assignments-Wipro\java 8 features Assignment>

Figure 5 FrequencyMapper



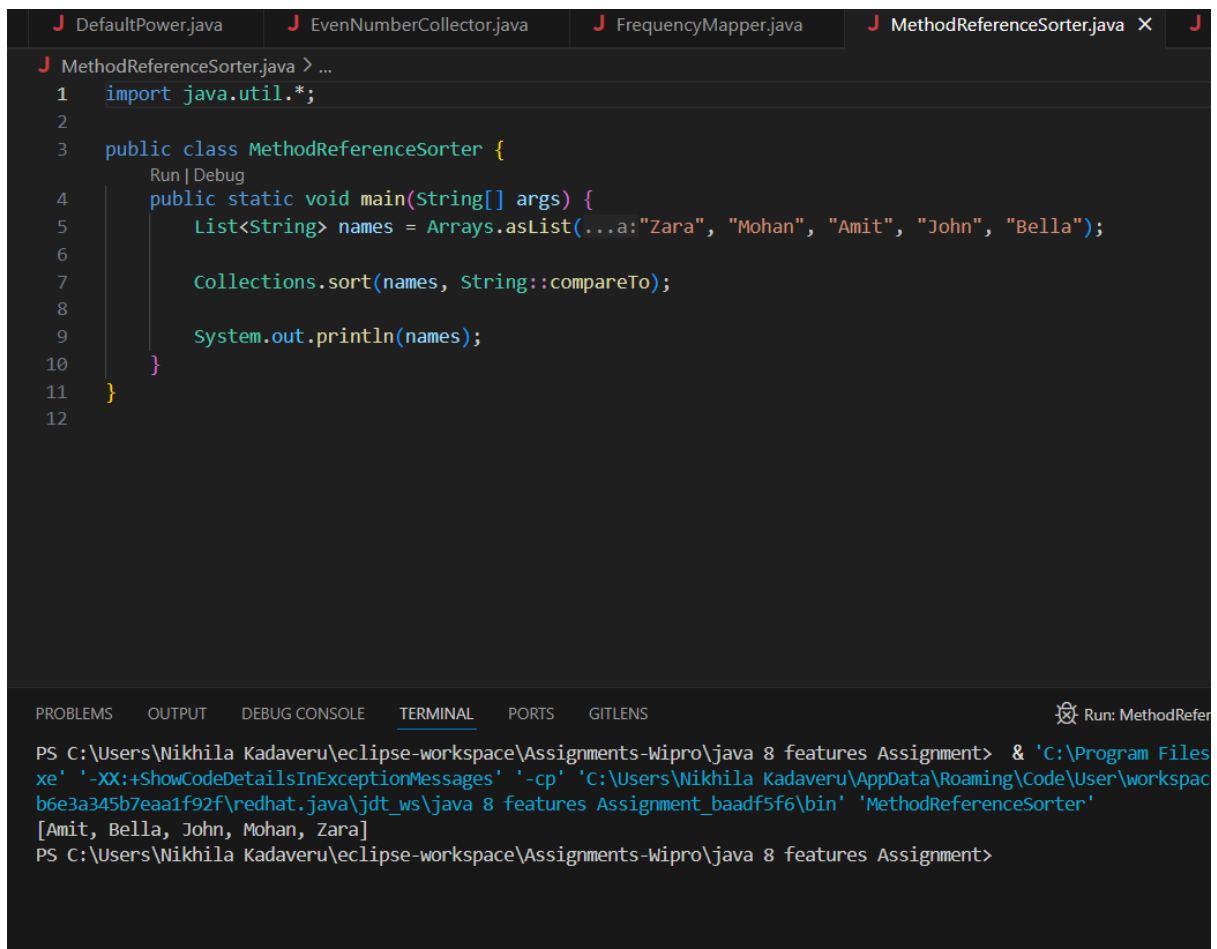
The screenshot displays the Eclipse IDE interface. The top toolbar shows several Java files: ANameFinder.java, DefaultPower.java, EvenNumberCollector.java, FrequencyMapper.java, and GreetingsPrinter.java. The main editor window is open to GreetingsPrinter.java, showing the following code:

```
1 public class GreetingsPrinter {
2     Run | Debug
3     public static void main(String[] args) {
4         MessagePrinter printer = message -> System.out.println("Greeting: " + message);
5         greetUser(message:"Welcome to Wipro!", printer);
6     }
7
8     @FunctionalInterface
9     interface MessagePrinter {
10         void print(String message);
11     }
12
13     public static void greetUser(String message, MessagePrinter printer) {
14         printer.print(message);
15     }
16 }
17
```

Below the editor, the 'TERMINAL' tab is active, showing the command used to run the program and its output:

```
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment> & 'C:\Program Files\Java\jdk-11.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\workspace\ba6e3a345b7eaa1f92f\redhat.java\jdt_ws\java 8 features Assignment_baadf5f6\bin' 'GreetingsPrinter'
Greeting: Welcome to Wipro!
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment>
```

Figure 6 GreetingsPrinter



The screenshot displays the Eclipse IDE interface. The top tab bar shows four open files: DefaultPower.java, EvenNumberCollector.java, FrequencyMapper.java, and MethodReferenceSorter.java. The MethodReferenceSorter.java file is active, showing the following code:

```
1 import java.util.*;
2
3 public class MethodReferenceSorter {
4     public static void main(String[] args) {
5         List<String> names = Arrays.asList(...a:"Zara", "Mohan", "Amit", "John", "Bella");
6
7         Collections.sort(names, String::compareTo);
8
9         System.out.println(names);
10    }
11 }
12
```

Below the code editor, the 'TERMINAL' tab is selected, showing the command prompt output:

```
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment> & 'C:\Program Files
xe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\workspac
b6e3a345b7eaa1f92f\redhat.java\jdt_ws\java 8 features Assignment_baadf5f6\bin' 'MethodReferenceSorter'
[Amit, Bella, John, Mohan, Zara]
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment>
```

Figure 7 MethodReferenceSorter

The screenshot shows the Eclipse IDE interface. At the top, there are tabs for several Java files: DefaultPower.java, EvenNumberCollector.java, FrequencyMapper.java, MethodReferenceSorter.java, and NameSorter.java. The NameSorter.java tab is active, showing the following code:

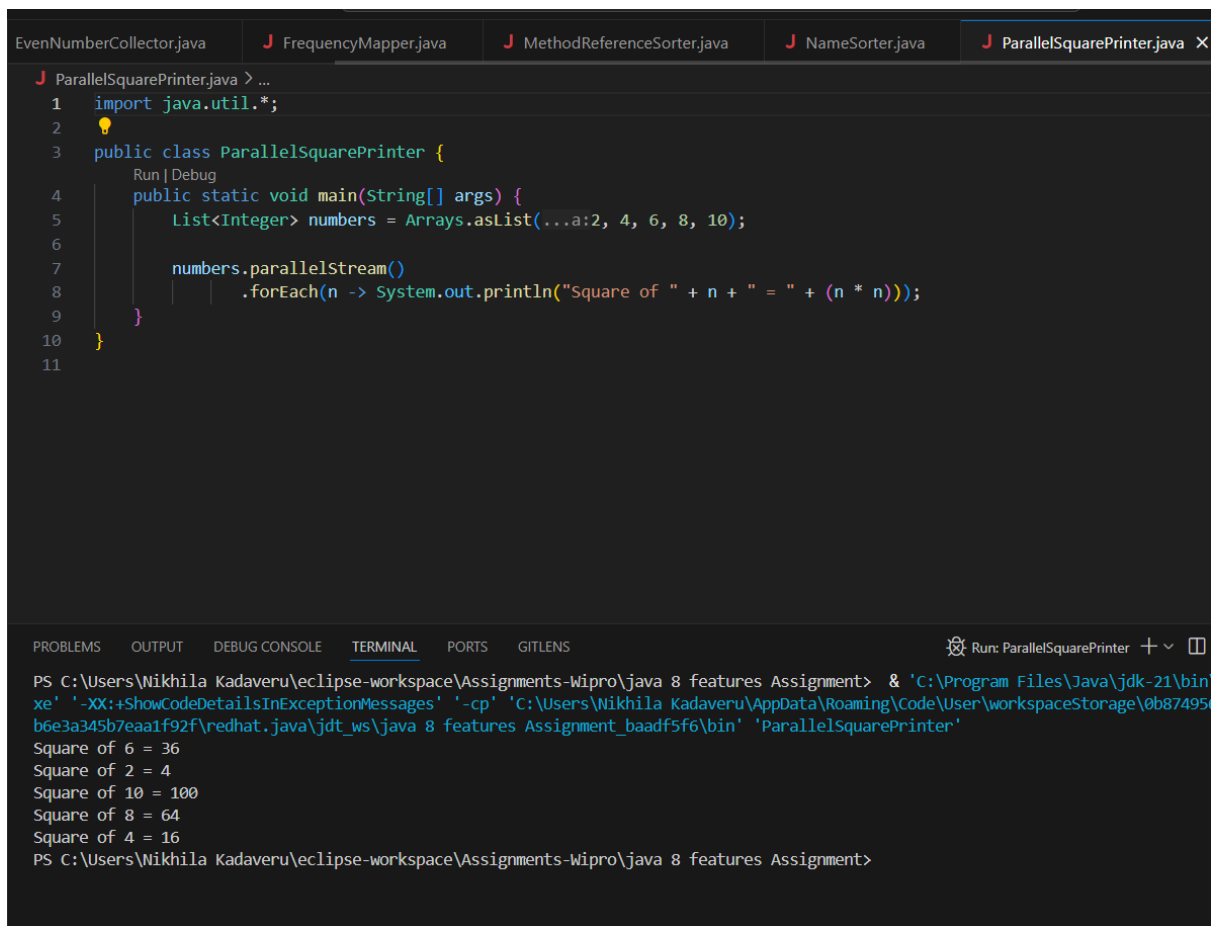
```
1 import java.util.*;
2
3 public class NameSorter {
4     public static void main(String[] args) {
5         List<String> names = Arrays.asList("Akki", "Anu", "Ram", "Rashmitha", "Geetanjali");
6         names.sort((name1, name2) -> name1.compareTo(name2));
7         names.forEach(System.out::println);
8     }
9 }
10
```

Below the code editor, the TERMINAL tab is selected, showing the output of running the NameSorter class. The output lists the names in sorted order: Ram, Rashmitha, Akki, Anu, Geetanjali. The terminal also shows the command prompt and the execution command used to run the program.

```
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment> ^C
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment>
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment> c:: cd 'c:\Users\Nikhila Kadaveru\workspace\Assignments-Wipro\java 8 features Assignment'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\workspaceStorage\0b87495676935eb6e3a345b7eaa1f92f\redhat\java 8 features Assignment_baadf5f6\bin' 'NameSorter'
Ram
Rashmitha
Akki
Anu
Geetanjali
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-Wipro\java 8 features Assignment>
```

Figure 8 NameSorter





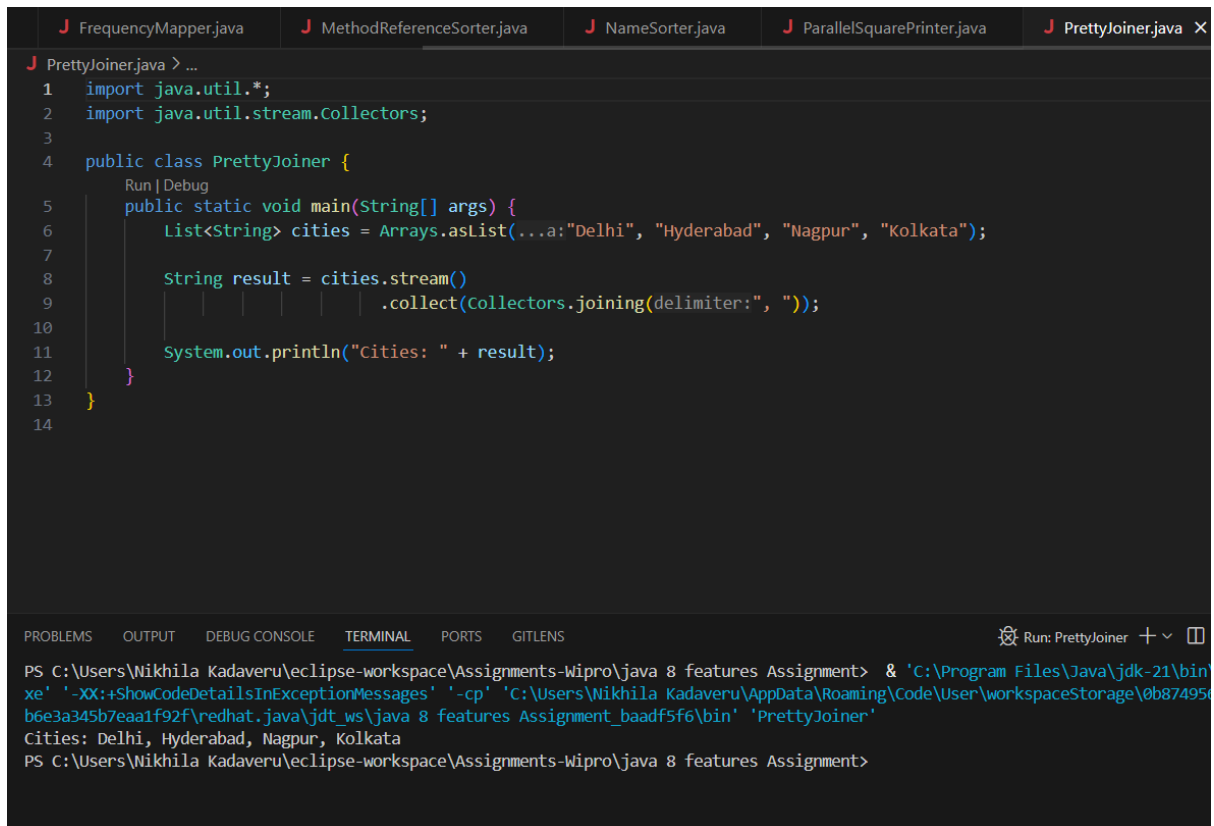
The screenshot displays the Eclipse IDE interface. The top tab bar shows several Java files: EvenNumberCollector.java, FrequencyMapper.java, MethodReferenceSorter.java, NameSorter.java, and ParallelSquarePrinter.java. The ParallelSquarePrinter.java file is currently open and shows the following code:

```
1 import java.util.*;
2
3 public class ParallelSquarePrinter {
4     public static void main(String[] args) {
5         List<Integer> numbers = Arrays.asList(...a:2, 4, 6, 8, 10);
6
7         numbers.parallelStream()
8             .forEach(n -> System.out.println("Square of " + n + " = " + (n * n)));
9     }
10 }
11
```

Below the code editor, the terminal window is active, showing the command used to run the program and its output:

```
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-wipro\java 8 features Assignment> & 'C:\Program Files\Java\jdk-21\bin\
xe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\workspaceStorage\0b87495
b6e3a345b7eaa1f92f\redhat.java\jdt_ws\java 8 features Assignment_baadf5f6\bin' 'ParallelSquarePrinter'
Square of 6 = 36
Square of 2 = 4
Square of 10 = 100
Square of 8 = 64
Square of 4 = 16
PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-wipro\java 8 features Assignment>
```

*Figure 9 ParallelSquarePrinter*



The screenshot shows the Eclipse IDE with the `PrettyJoiner.java` file open. The code defines a `main` method that takes an array of city names, converts it to a `List`, and uses `Stream` and `Collectors.joining` to join the elements with a comma and space. The terminal output shows the command to run the program and the resulting output: `Cities: Delhi, Hyderabad, Nagpur, Kolkata`.

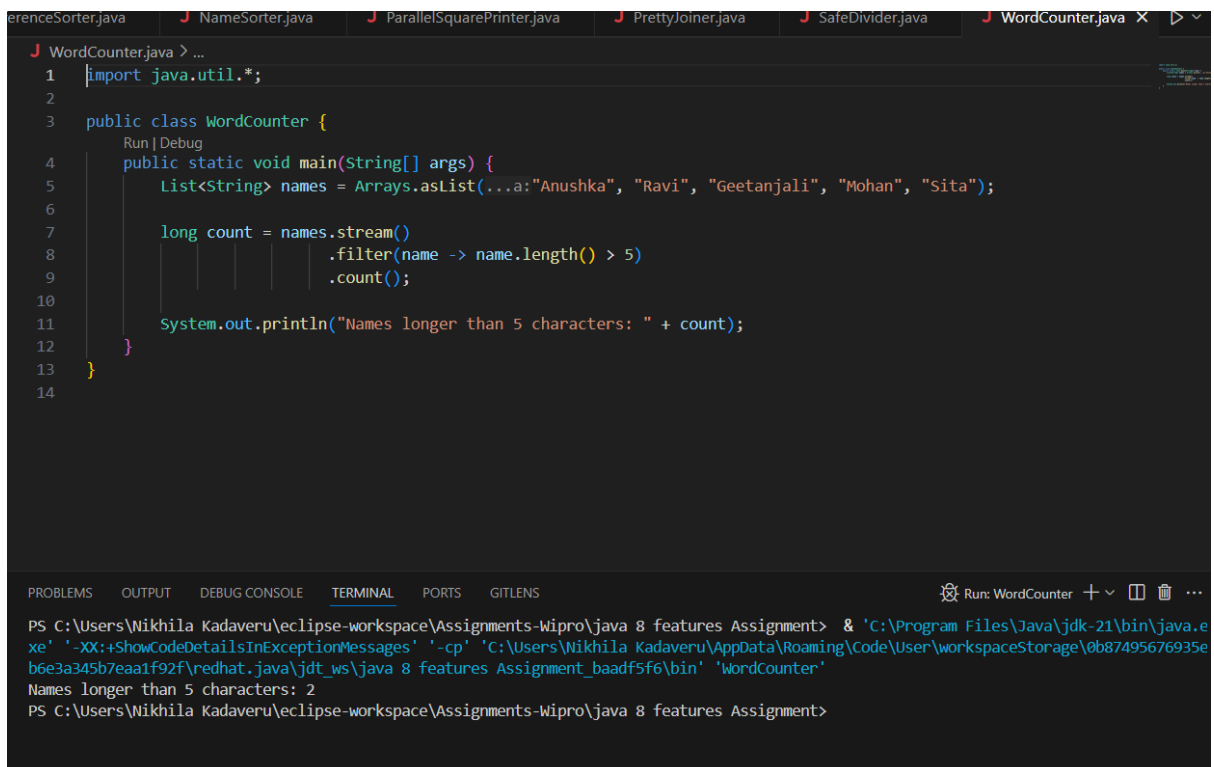
```
1 import java.util.*;
2 import java.util.stream.Collectors;
3
4 public class PrettyJoiner {
5     public static void main(String[] args) {
6         List<String> cities = Arrays.asList("Delhi", "Hyderabad", "Nagpur", "Kolkata");
7
8         String result = cities.stream()
9             .collect(Collectors.joining(", "));
10
11         System.out.println("Cities: " + result);
12     }
13 }
14
```

PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-wipro\java 8 features Assignment> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\workspaceStorage\0b874956b6e3a345b7eaa1f92f\redhat.java\jdt\_ws\java 8 features Assignment\_baadf5f6\bin' 'PrettyJoiner'

Cities: Delhi, Hyderabad, Nagpur, Kolkata

PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-wipro\java 8 features Assignment>

Figure 10 *PrettyJoiner*



The screenshot shows the Eclipse IDE with the `WordCounter.java` file open. The code defines a `main` method that takes an array of names, filters those with a length greater than 5, and counts them. The terminal output shows the command to run the program and the resulting output: `Names longer than 5 characters: 2`.

```
1 import java.util.*;
2
3 public class WordCounter {
4     public static void main(String[] args) {
5         List<String> names = Arrays.asList("Anushka", "Ravi", "Geetanjali", "Mohan", "Sita");
6
7         long count = names.stream()
8             .filter(name -> name.length() > 5)
9             .count();
10
11         System.out.println("Names longer than 5 characters: " + count);
12     }
13 }
14
```

PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-wipro\java 8 features Assignment> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Nikhila Kadaveru\AppData\Roaming\Code\User\workspaceStorage\0b874956b6e3a345b7eaa1f92f\redhat.java\jdt\_ws\java 8 features Assignment\_baadf5f6\bin' 'WordCounter'

Names longer than 5 characters: 2

PS C:\Users\Nikhila Kadaveru\eclipse-workspace\Assignments-wipro\java 8 features Assignment>

Figure 11 *WordCounter*