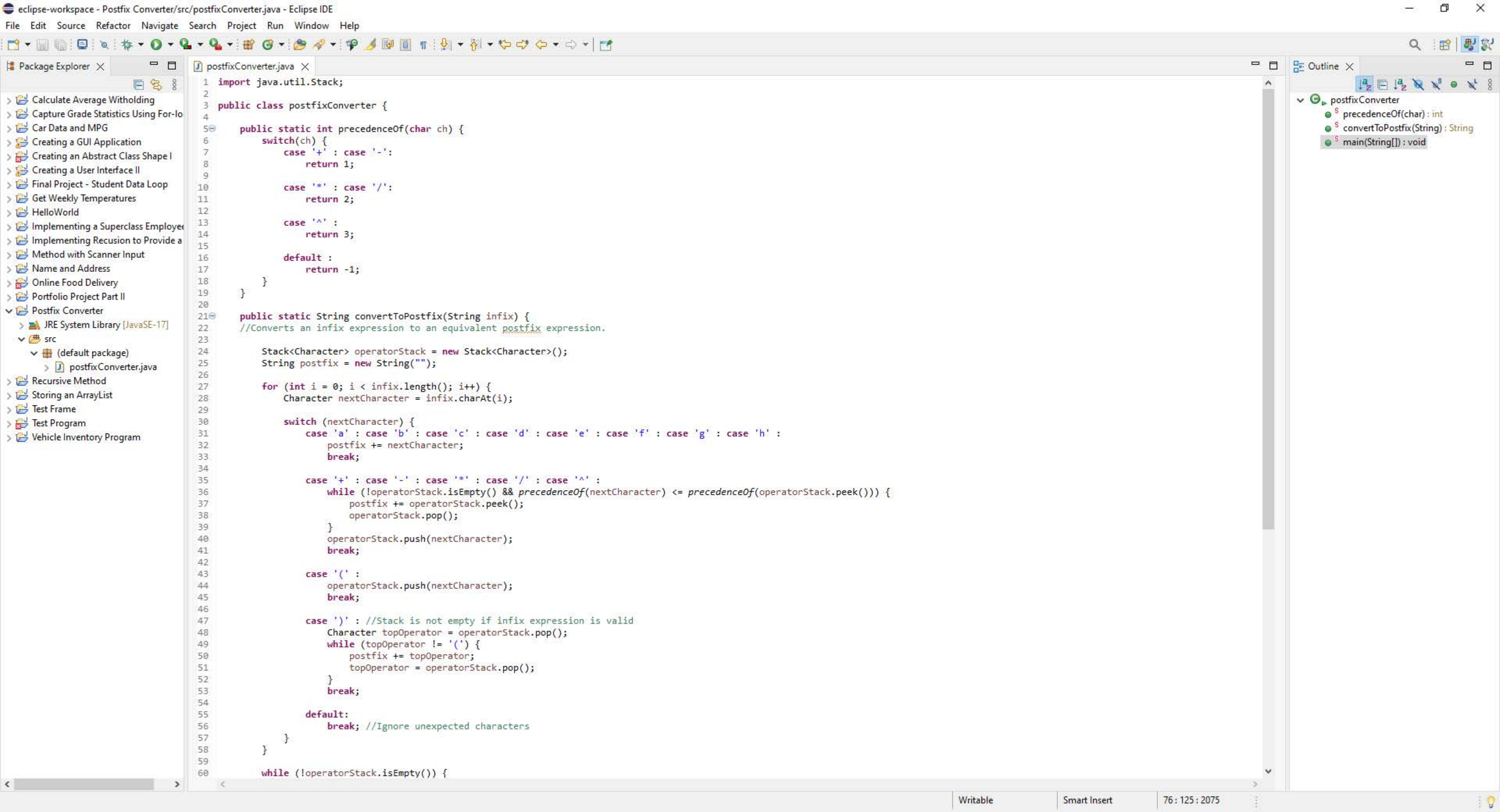
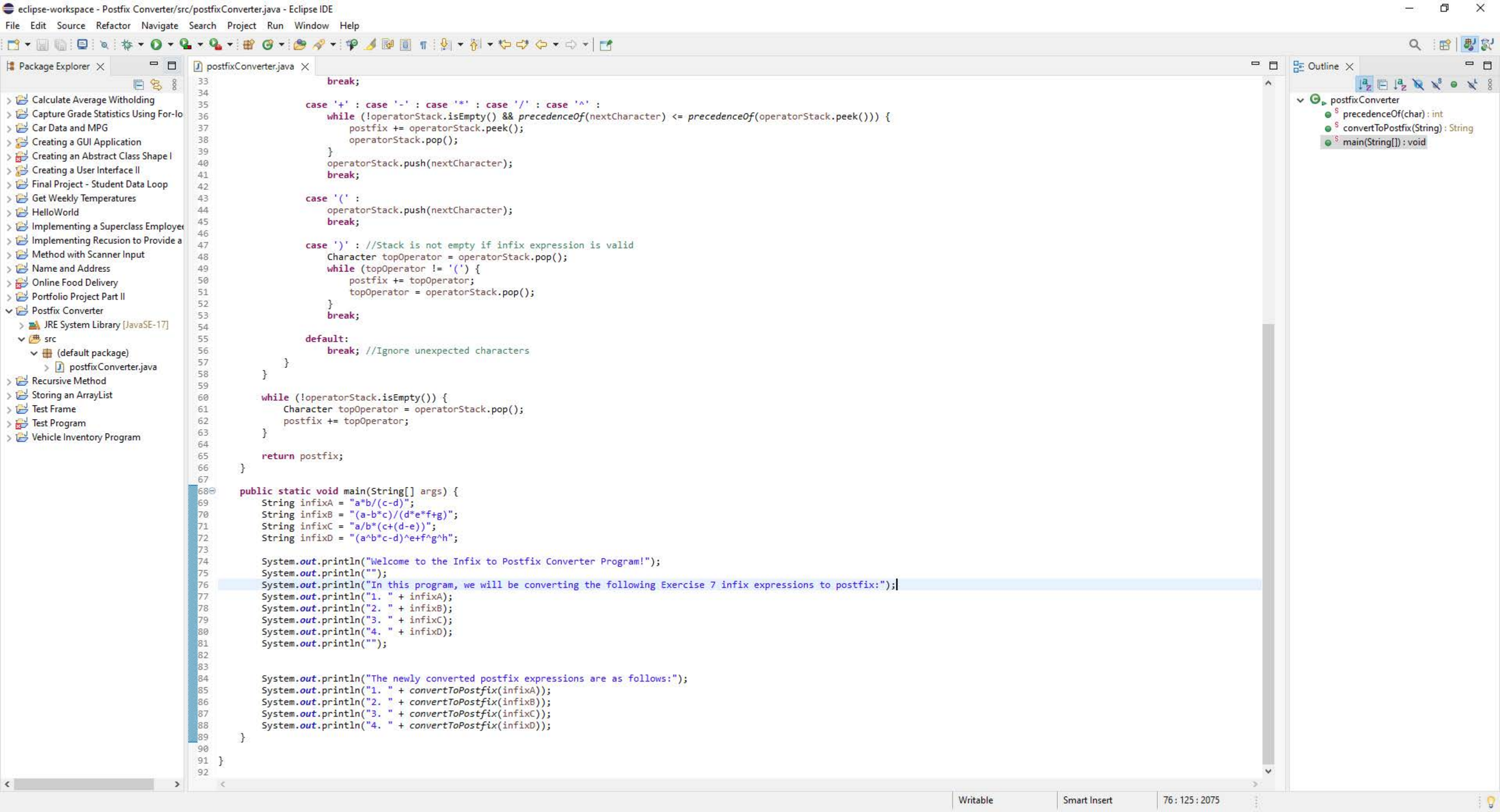
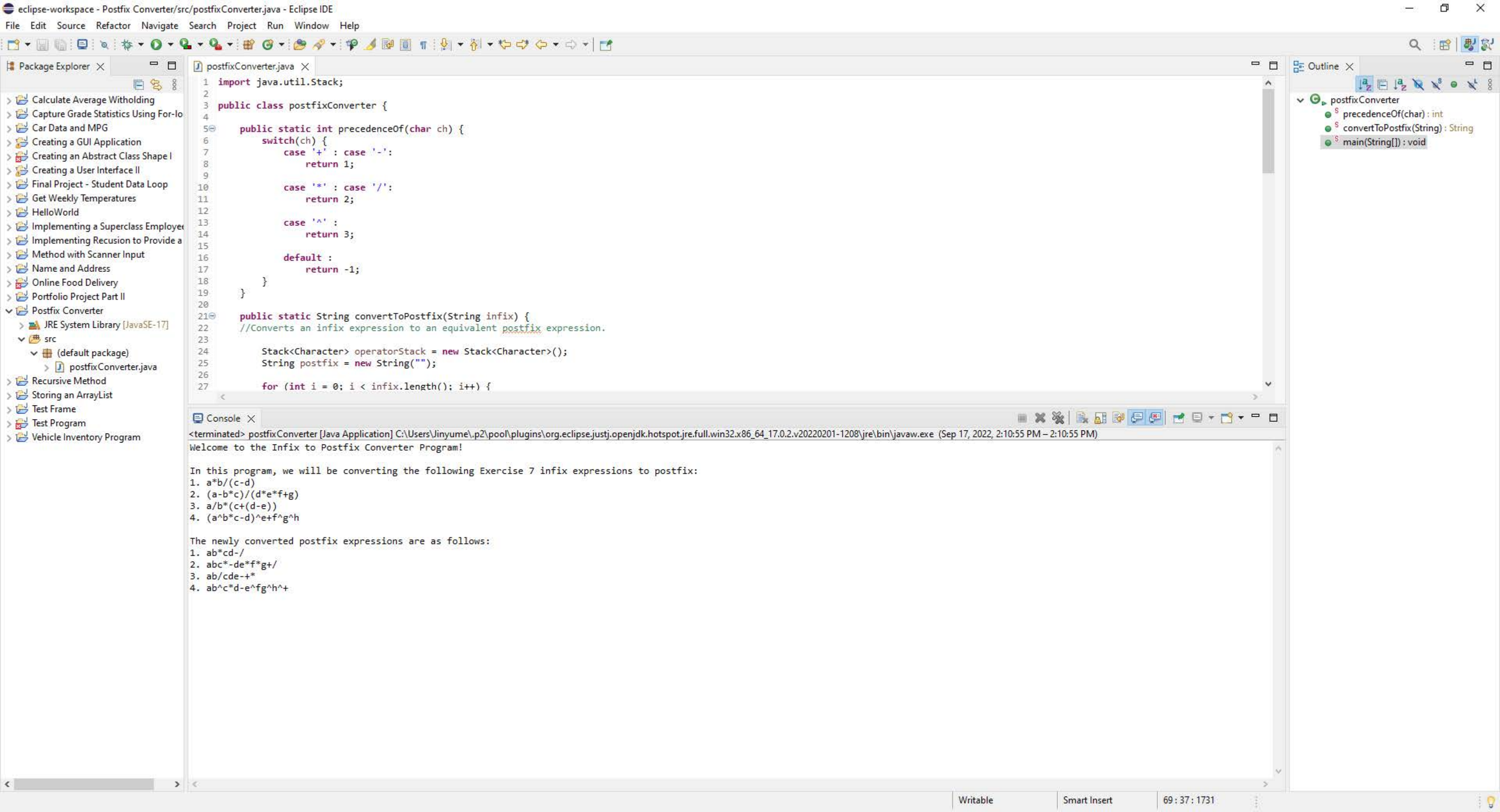

eclipse-workspace - Postfix Converter/src/postfixConverter.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Calculate Average Witholding
Capture Grade Statistics Using For-lo
Car Data and MPG
Creating a GUI Application
Creating an Abstract Class Shape I
Creating a User Interface II
Final Project - Student Data Loop
Get Weekly Temperatures
HelloWorld
Implementing a Superclass Employee
Implementing Recusion to Provide a
Method with Scanner Input
Name and Address
Online Food Delivery
Portfolio Project Part II
Postfix Converter
JRE System Library [JavaSE-17]
src
(default package)
postfixConverter.java
Recursive Method
Storing an ArrayList
Test Frame
Test Program
Vehicle Inventory Program
postfixConverter.java
1 import java.util.Stack;
2
3 public class postfixConverter {
4
5 public static int precedenceOf(char ch) {
6 switch(ch) {
7 case '+' : case '-':
8 return 1;
9
10 case '*' : case '/':
11 return 2;
12
13 case '^' :
14 return 3;
15
16 default :
17 return -1;
18 }
19 }
20
21 public static String convertToPostfix(String infix) {
22 //Converts an infix expression to an equivalent postfix expression.
23
24 Stack<Character> operatorStack = new Stack<Character>();
25 String postfix = new String("");
26
27 for (int i = 0; i < infix.length(); i++) {
28 Character nextCharacter = infix.charAt(i);
29
30 switch (nextCharacter) {
31 case 'a' : case 'b' : case 'c' : case 'd' : case 'e' : case 'f' : case 'g' : case 'h' :
32 postfix += nextCharacter;
33 break;
34
35 case '+' : case '-' : case '*' : case '/' : case '^' :
36 while (!operatorStack.isEmpty() && precedenceOf(nextCharacter) <= precedenceOf(operatorStack.peek())) {
37 postfix += operatorStack.peek();
38 operatorStack.pop();
39 }
40 operatorStack.push(nextCharacter);
41 break;
42
43 case '(' :
44 operatorStack.push(nextCharacter);
45 break;
46
47 case ')' : //Stack is not empty if infix expression is valid
48 Character topOperator = operatorStack.pop();
49 while (topOperator != '(') {
50 postfix += topOperator;
51 topOperator = operatorStack.pop();
52 }
53 break;
54
55 default:
56 break; //Ignore unexpected characters
57 }
58 }
59
60 while (!operatorStack.isEmpty()) {
Outline
postfixConverter
precedenceOf(char) : int
convertToPostfix(String) : String
main(String[]) : void
Writable
Smart Insert
76 : 125 : 2075


eclipse-workspace - Postfix Converter/src/postfixConverter.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Calculate Average Witholding
Capture Grade Statistics Using For-lo
Car Data and MPG
Creating a GUI Application
Creating an Abstract Class Shape I
Creating a User Interface II
Final Project - Student Data Loop
Get Weekly Temperatures
HelloWorld
Implementing a Superclass Employee
Implementing Recusion to Provide a
Method with Scanner Input
Name and Address
Online Food Delivery
Portfolio Project Part II
Postfix Converter
JRE System Library [JavaSE-17]
src
(default package)
postfixConverter.java
Recursive Method
Storing an ArrayList
Test Frame
Test Program
Vehicle Inventory Program
postfixConverter.java
33 break;
34
35 case '+' : case '-' : case '*' : case '/' : case '^' :
36 while (!operatorStack.isEmpty() && precedenceOf(nextCharacter) <= precedenceOf(operatorStack.peek())) {
37 postfix += operatorStack.peek();
38 operatorStack.pop();
39 }
40 operatorStack.push(nextCharacter);
41 break;
42
43 case '(' :
44 operatorStack.push(nextCharacter);
45 break;
46
47 case ')' : //Stack is not empty if infix expression is valid
48 Character topOperator = operatorStack.pop();
49 while (topOperator != '(') {
50 postfix += topOperator;
51 topOperator = operatorStack.pop();
52 }
53 break;
54
55 default:
56 break; //Ignore unexpected characters
57 }
58 }
59
60 while (!operatorStack.isEmpty()) {
61 Character topOperator = operatorStack.pop();
62 postfix += topOperator;
63 }
64
65 return postfix;
66 }
67
68 public static void main(String[] args) {
69 String infixA = "a*b/(c-d)";
70 String infixB = "(a-b*c)/(d*e*f+g)";
71 String infixC = "a/b*(c+(d-e))";
72 String infixD = "(a^b*c-d)^e+f^g^h";
73
74 System.out.println("Welcome to the Infix to Postfix Converter Program!");
75 System.out.println("");
76 System.out.println("In this program, we will be converting the following Exercise 7 infix expressions to postfix:");
77 System.out.println("1. " + infixA);
78 System.out.println("2. " + infixB);
79 System.out.println("3. " + infixC);
80 System.out.println("4. " + infixD);
81 System.out.println("");
82
83
84 System.out.println("The newly converted postfix expressions are as follows:");
85 System.out.println("1. " + convertToPostfix(infixA));
86 System.out.println("2. " + convertToPostfix(infixB));
87 System.out.println("3. " + convertToPostfix(infixC));
88 System.out.println("4. " + convertToPostfix(infixD));
89 }
90
91 }
92
Outline
postfixConverter
precedenceOf(char) : int
convertToPostfix(String) : String
main(String[]) : void
Writable
Smart Insert
76 : 125 : 2075


eclipse-workspace - Postfix Converter/src/postfixConverter.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Calculate Average Witholding
Capture Grade Statistics Using For-lo
Car Data and MPG
Creating a GUI Application
Creating an Abstract Class Shape I
Creating a User Interface II
Final Project - Student Data Loop
Get Weekly Temperatures
HelloWorld
Implementing a Superclass Employee
Implementing Recusion to Provide a
Method with Scanner Input
Name and Address
Online Food Delivery
Portfolio Project Part II
Postfix Converter
JRE System Library [JavaSE-17]
src
(default package)
postfixConverter.java
Recursive Method
Storing an ArrayList
Test Frame
Test Program
Vehicle Inventory Program
postfixConverter.java
1 import java.util.Stack;
2
3 public class postfixConverter {
4
5 public static int precedenceOf(char ch) {
6 switch(ch) {
7 case '+' : case '-':
8 return 1;
9
10 case '*' : case '/':
11 return 2;
12
13 case '^' :
14 return 3;
15
16 default :
17 return -1;
18 }
19 }
20
21 public static String convertToPostfix(String infix) {
22 //Converts an infix expression to an equivalent postfix expression.
23
24 Stack<Character> operatorStack = new Stack<Character>();
25 String postfix = new String("");
26
27 for (int i = 0; i < infix.length(); i++) {
Outline
postfixConverter
precedenceOf(char) : int
convertToPostfix(String) : String
main(String[]) : void
Console
<terminated> postfixConverter [Java Application] C:\Users\Jinyume\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (Sep 17, 2022, 2:10:55 PM – 2:10:55 PM)
Welcome to the Infix to Postfix Converter Program!
In this program, we will be converting the following Exercise 7 infix expressions to postfix:
1. a*b/(c-d)
2. (a-b*c)/(d*e*f+g)
3. a/b*(c+(d-e))
4. (a^b*c-d)^e+f^g^h
The newly converted postfix expressions are as follows:
1. ab*cd-/
2. abc*-de*f*g+/
3. ab/cde-+*
4. ab^c*d-e^fg^h^+
Writable
Smart Insert
69 : 37 : 1731

Exercise 7:
1. a*b/(c-d)
2. (a-b*c)/(d*e*f+g)
3. a/b*(c+(d-e))
4. (a^b*c-d)^e+f^g^h

Exercise 7 after conversions:
1. ab*cd-/
2. abc*-de*f*g+/
3. ab/cde-+*
4. ab^c*d-e^fg^h^+