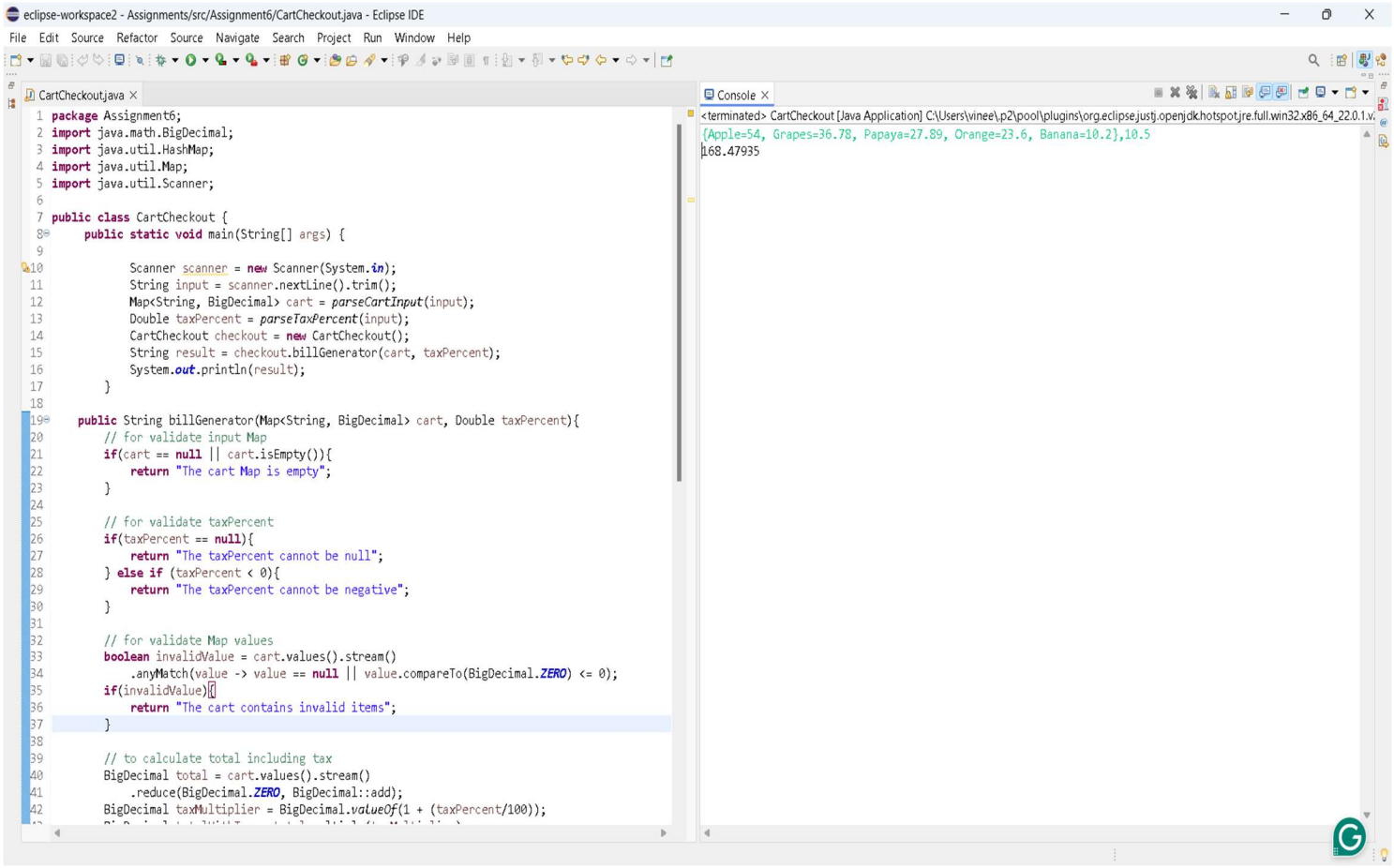
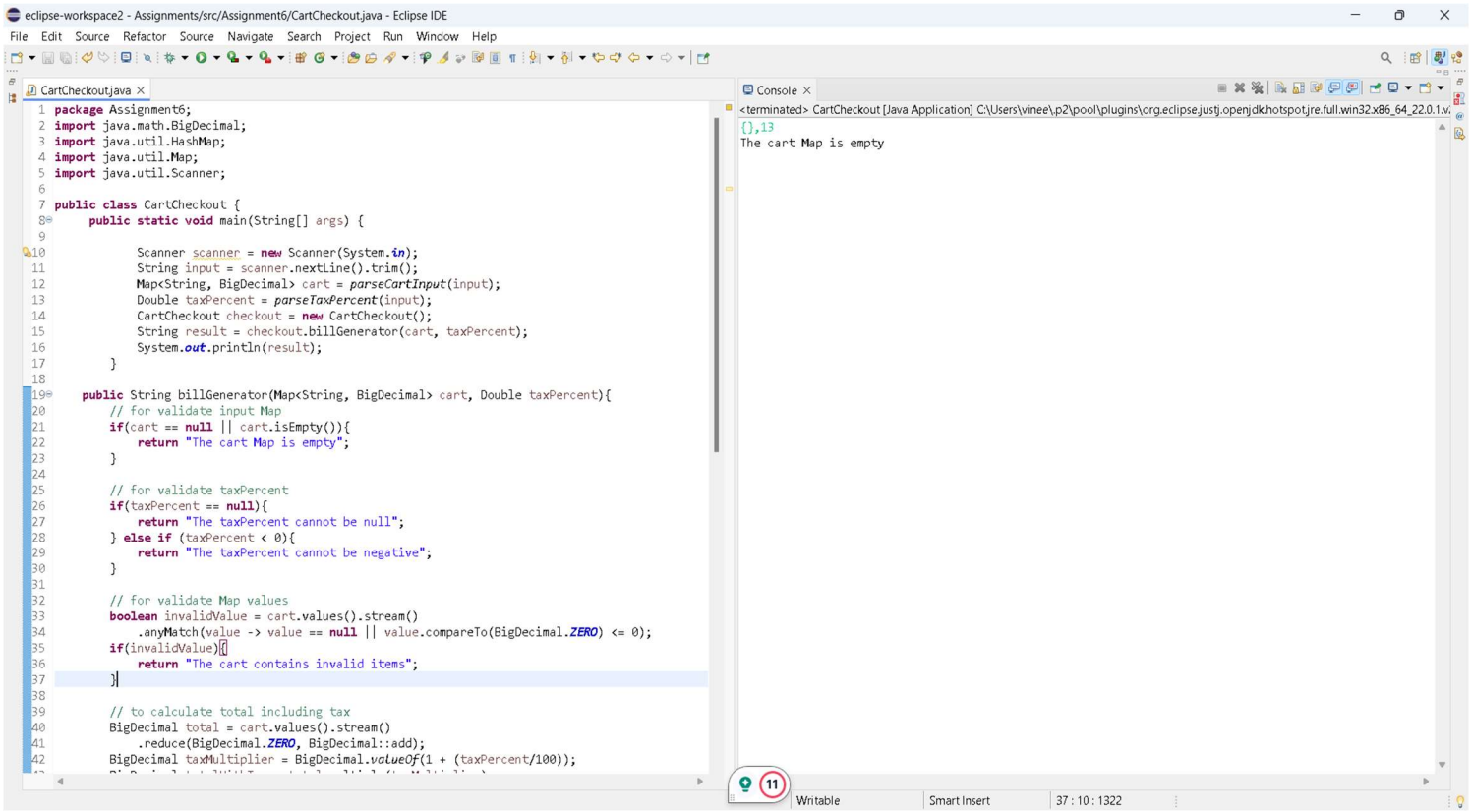
```java
package Assignment6;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class CartCheckout {
        public static void main(String[] args) {

            Scanner scanner = new Scanner(System.in);
            String input = scanner.nextLine().trim();
            Map<String, BigDecimal> cart = parseCartInput(input);
            Double taxPercent = parseTaxPercent(input);
            CartCheckout checkout = new CartCheckout();
            String result = checkout.billGenerator(cart, taxPercent);
            System.out.println(result);
        }

    public String billGenerator(Map<String, BigDecimal> cart, Double taxPercent){
        // for validate input Map
        if(cart == null || cart.isEmpty()){
            return "The cart Map is empty";
        }

        // for validate taxPercent
        if(taxPercent == null){
            return "The taxPercent cannot be null";
        } else if (taxPercent < 0){
            return "The taxPercent cannot be negative";
        }
```

```
    // for validate Map values
    boolean invalidValue = cart.values().stream()
        .anyMatch(value -> value == null || value.compareTo(BigDecimal.ZERO) <= 0);
    if(invalidValue){
        return "The cart contains invalid items";
    }

    // to calculate total including tax
    BigDecimal total = cart.values().stream()
        .reduce(BigDecimal.ZERO, BigDecimal::add);
    BigDecimal taxMultiplier = BigDecimal.valueOf(1 + (taxPercent/100));
    BigDecimal totalWithTax = total.multiply(taxMultiplier);
    return totalWithTax.toPlainString();
}

private static Double parseTaxPercent(String input){
    // to extract the tax percentage part from the input
    int splitIndex = input.lastIndexOf('}');
    String taxPercentString = input.substring(splitIndex + 2).trim();

    // to handle the case where taxPercent is null
    if ("null".equalsIgnoreCase(taxPercentString)) {
        return null;
    }

    return Double.parseDouble(taxPercentString);
}
```

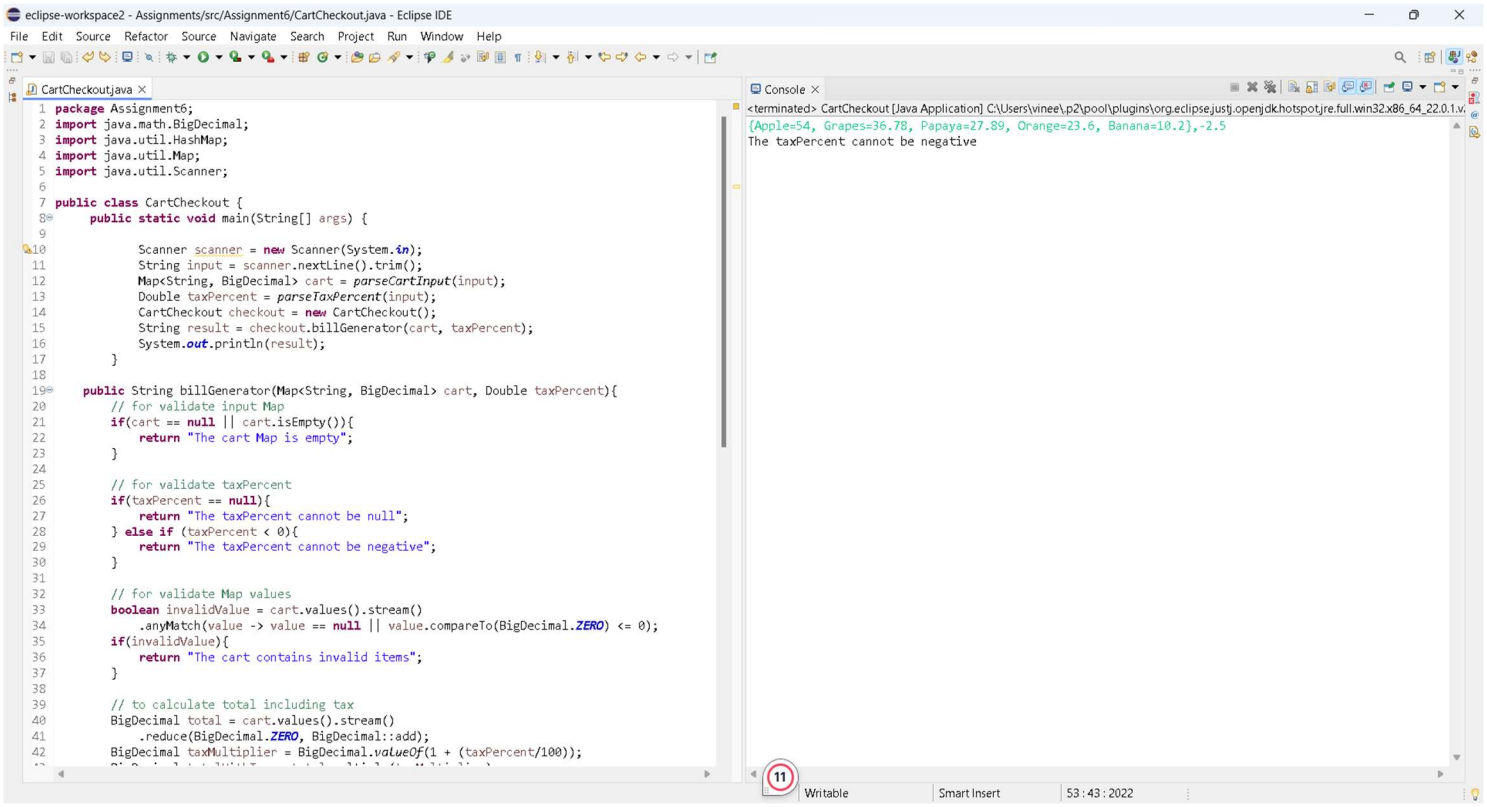
```
  private static Map<String, BigDecimal> parseCartInput(String input){
      Map<String, BigDecimal> cart = new HashMap<>();

      // to extract cart part from the input
      int splitIndex = input.lastIndexOf('}');
      String cartInput = input.substring(1, splitIndex).trim();

      // to handle the case where the cart is empty
      if(cartInput.isEmpty()){
         return cart;
      }
      String[] items = cartInput.split(",");
      for(String item : items){
         String[] keyValue = item.split("=");
         String itemName = keyValue[0].trim();
         BigDecimal itemPrice = new BigDecimal(keyValue[1].trim());
         cart.put(itemName, itemPrice);
      }
      return cart;
   }
}
```

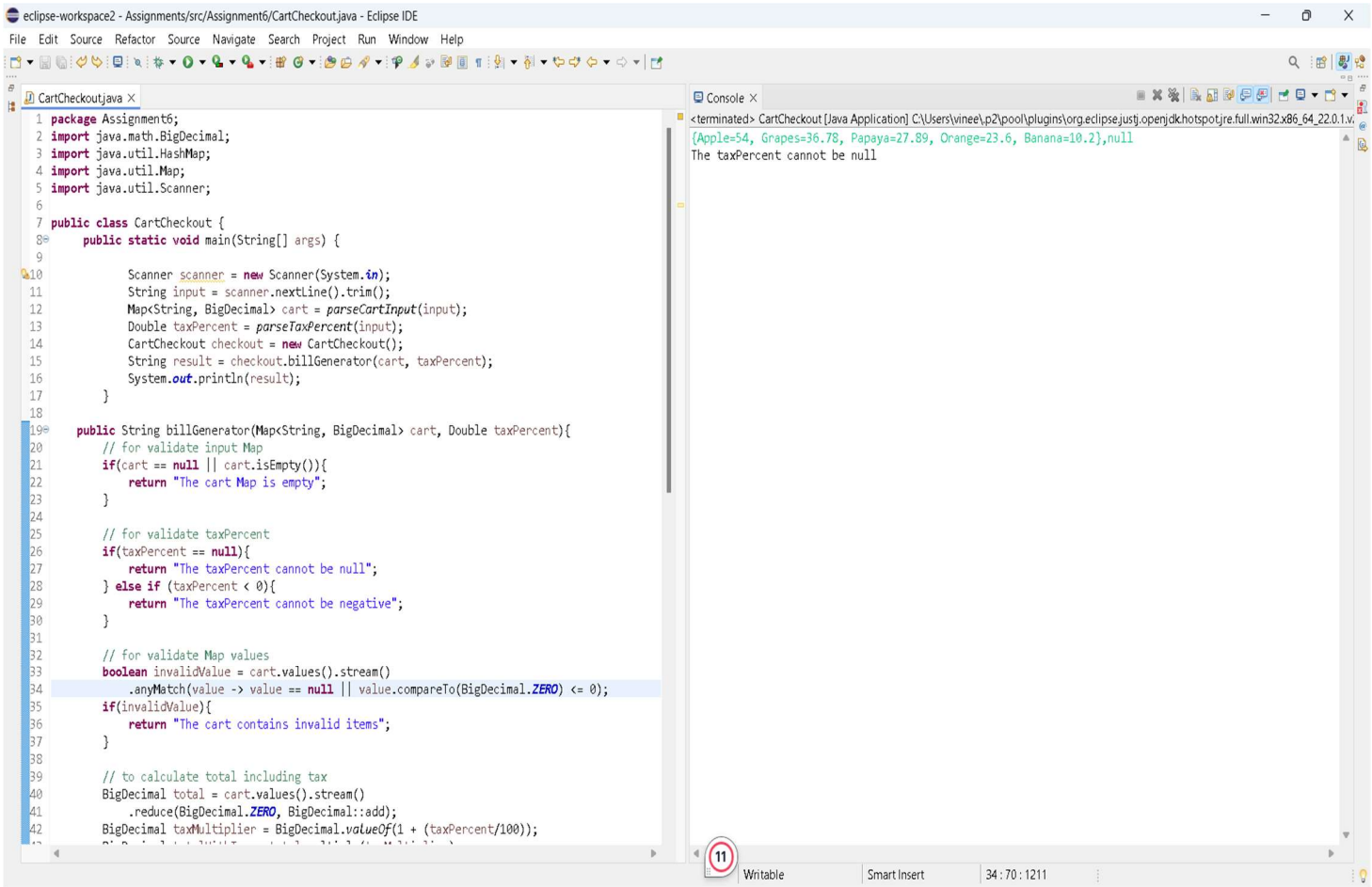



```java
package Assignment6;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class CartCheckout {
    public static void main(String[] args) {

            Scanner scanner = new Scanner(System.in);
            String input = scanner.nextLine().trim();
            Map<String, BigDecimal> cart = parseCartInput(input);
            Double taxPercent = parseTaxPercent(input);
            CartCheckout checkout = new CartCheckout();
            String result = checkout.billGenerator(cart, taxPercent);
            System.out.println(result);
        }

    public String billGenerator(Map<String, BigDecimal> cart, Double taxPercent){
        // for validate input Map
        if(cart == null || cart.isEmpty()){
            return "The cart Map is empty";
        }

        // for validate taxPercent
        if(taxPercent == null){
            return "The taxPercent cannot be null";
        } else if (taxPercent < 0){
            return "The taxPercent cannot be negative";
        }

        // for validate Map values
        boolean invalidValue = cart.values().stream()
            .anyMatch(value -> value == null || value.compareTo(BigDecimal.ZERO) <= 0);
        if(invalidValue){
            return "The cart contains invalid items";
        }

        // to calculate total including tax
        BigDecimal total = cart.values().stream()
            .reduce(BigDecimal.ZERO, BigDecimal::add);
        BigDecimal taxMultiplier = BigDecimal.valueOf(1 + (taxPercent/100));
```




```java
package Assignment6;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class CartCheckout {
    public static void main(String[] args) {

            Scanner scanner = new Scanner(System.in);
            String input = scanner.nextLine().trim();
            Map<String, BigDecimal> cart = parseCartInput(input);
            Double taxPercent = parseTaxPercent(input);
            CartCheckout checkout = new CartCheckout();
            String result = checkout.billGenerator(cart, taxPercent);
            System.out.println(result);
        }

    public String billGenerator(Map<String, BigDecimal> cart, Double taxPercent){
        // for validate input Map
        if(cart == null || cart.isEmpty()){
            return "The cart Map is empty";
        }

        // for validate taxPercent
        if(taxPercent == null){
            return "The taxPercent cannot be null";
        } else if (taxPercent < 0){
            return "The taxPercent cannot be negative";
        }

        // for validate Map values
        boolean invalidValue = cart.values().stream()
            .anyMatch(value -> value == null || value.compareTo(BigDecimal.ZERO) <= 0);
        if(invalidValue){
            return "The cart contains invalid items";
        }

        // to calculate total including tax
        BigDecimal total = cart.values().stream()
            .reduce(BigDecimal.ZERO, BigDecimal::add);
        BigDecimal taxMultiplier = BigDecimal.valueOf(1 + (taxPercent/100));
```


eclipse-workspace2 - Assignments/src/Assignment6/CartCheckout.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
CartCheckout.java
Console
<terminated> CartCheckout [Java Application] C:\Users\vinee\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v.
{Apple=54, Grapes=36.78, Papaya=27.89, Orange=23.6, Banana=10.2},-2.5
The taxPercent cannot be negative
Writable
Smart Insert
53 : 43 : 2022


eclipse-workspace2 - Assignments/src/Assignment6/CartCheckout.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
CartCheckout.java
Console
<terminated> CartCheckout [Java Application] C:\Users\vinee\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v.
{Apple=54, Grapes=36.78, Papaya=27.89, Orange=23.6, Banana=10.2},null
The taxPercent cannot be null
Writable
Smart Insert
34 : 70 : 1211