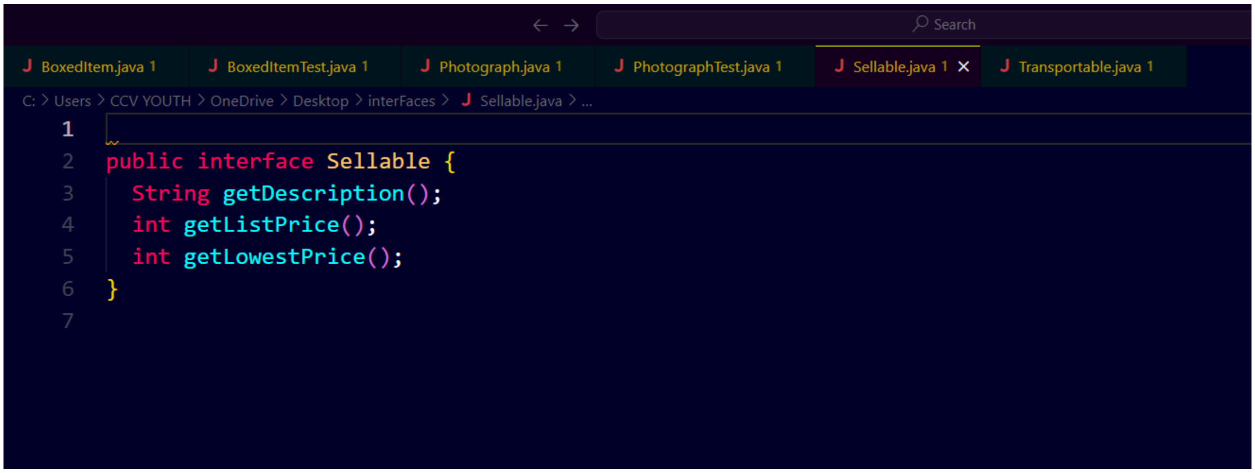
Documentation for Sellable, Transportable, Photograph, and BoxedItem Java Project

File: `Sellable.java`

Purpose: Defines the `Sellable` interface for items that can be sold.



```java

public interface Sellable {
  String getDescription();
  int getListPrice();
  int getLowestPrice();
}

```

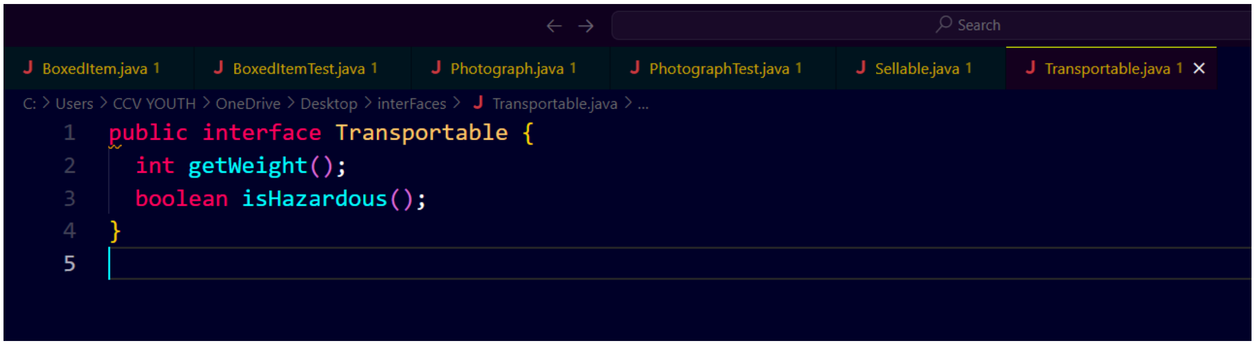
Methods:

- `String getDescription()`: Returns a description of the item.
- `int getListPrice()`: Returns the list price of the item in cents.
- `int getLowestPrice()`: Returns the lowest possible price of the item in cents.

Usage: Classes implementing `Sellable` must define these methods, ensuring they have a description and two price points.

File: Transportable.java

Purpose: Defines the Transportable interface for items that can be transported.



```java
public interface Transportable {
  int getWeight();
  boolean isHazardous();
}

```

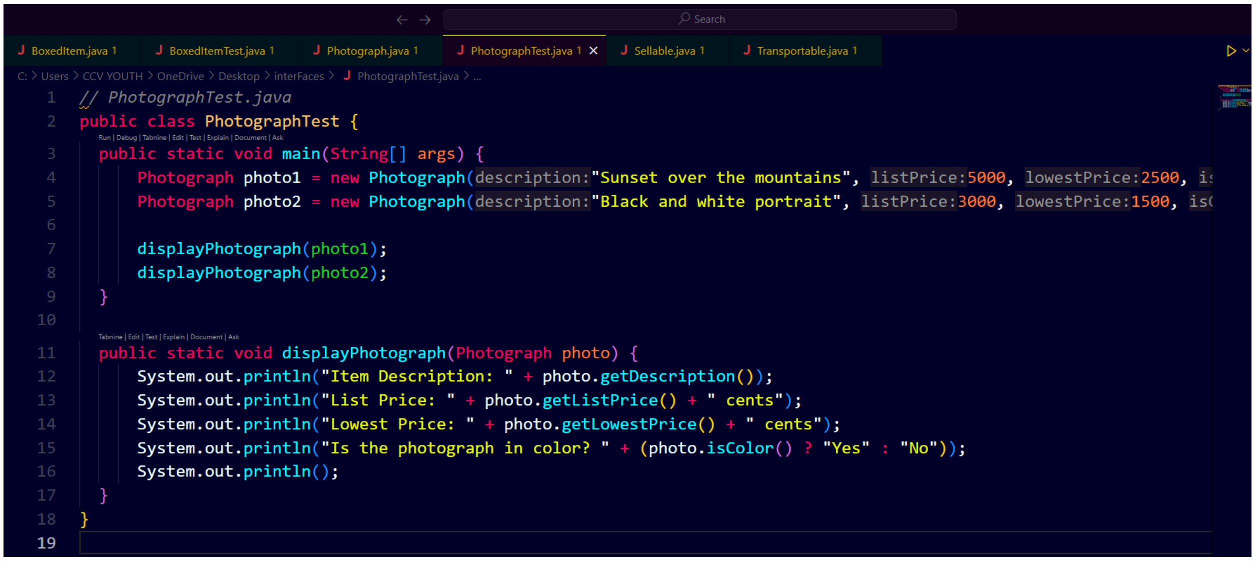
Methods:

- int getWeight(): Returns the weight of the item in grams.
- boolean isHazardous(): Returns true if the item is hazardous, false otherwise.

Usage: Classes implementing Transportable must define these methods, specifying weight and hazard status for transportable items.

File: Photograph.java

Purpose: Represents a Photograph that can be sold. Implements the Sellable interface.


Search
BoxedItem.java 1
BoxedItemTest.java 1
Photograph.java 1
PhotographTest.java 1
Sellable.java 1
Transportable.java 1
C: > Users > CCV YOUTH > OneDrive > Desktop > interFaces > PhotographTest.java > ...
Run | Debug | Tabnine | Edit | Test | Explain | Document | Ask
Tabnine | Edit | Test | Explain | Document | Ask

```java
// PhotographTest.java
public class PhotographTest {
  public static void main(String[] args) {
    Photograph photo1 = new Photograph(description:"Sunset over the mountains", listPrice:5000, lowestPrice:2500, is
    Photograph photo2 = new Photograph(description:"Black and white portrait", listPrice:3000, lowestPrice:1500, is(

    displayPhotograph(photo1);
    displayPhotograph(photo2);
  }

  public static void displayPhotograph(Photograph photo) {
    System.out.println("Item Description: " + photo.getDescription());
    System.out.println("List Price: " + photo.getListPrice() + " cents");
    System.out.println("Lowest Price: " + photo.getLowestPrice() + " cents");
    System.out.println("Is the photograph in color? " + (photo.isColor() ? "Yes" : "No"));
    System.out.println();
  }
}

```

Attributes:

- `description`: Description of the photograph (e.g., "Sunset over the mountains").
- `listPrice`: The photograph's list price in cents.
- `lowestPrice`: The lowest possible price for the photograph in cents.
- `isColor`: Boolean indicating if the photograph is in color.

Constructor:

- `Photograph(String description, int listPrice, int lowestPrice, boolean isColor)`: Initializes the photograph with its description, prices, and color status.

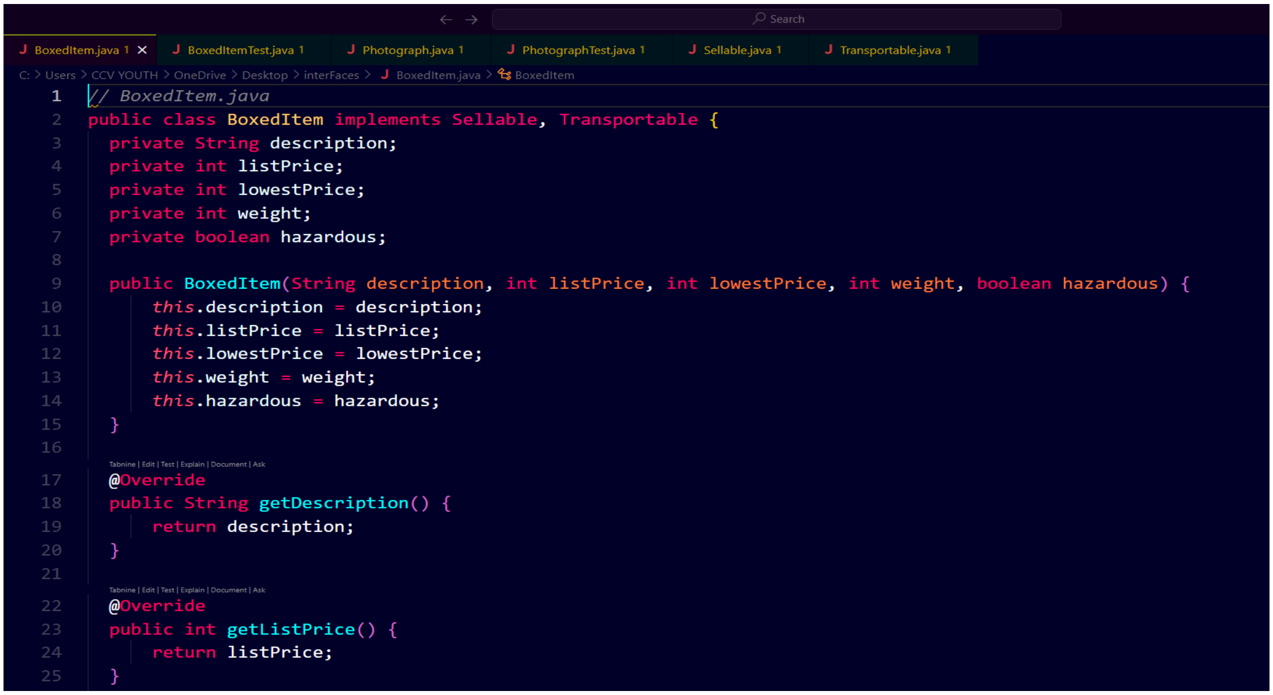
Methods:

- `String getDescription()`: Returns the photograph's description.
- `int getListPrice()`: Returns the photograph's list price.
- `int getLowestPrice()`: Returns the photograph's lowest price.
- `boolean isColor()`: Returns `true` if the photograph is in color, `false` otherwise.

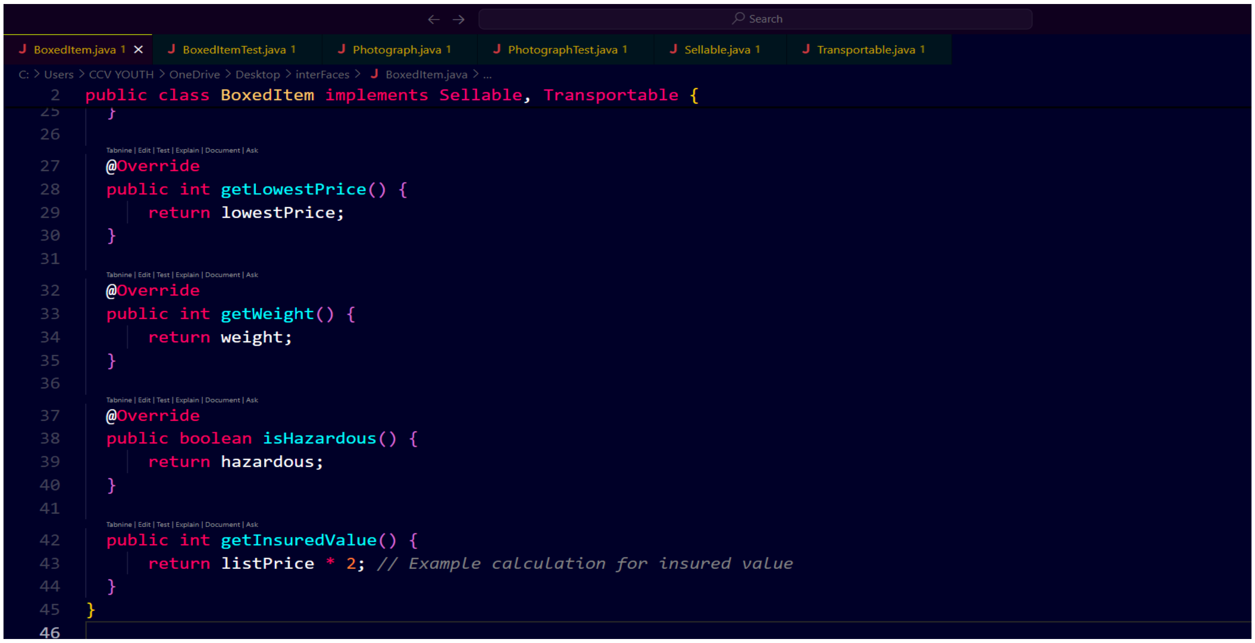
Usage: Used to create and manage photographs that are sellable, with color information.

File: `BoxedItem.java`

Purpose: Represents an item that is boxed and transportable. Implements both `Sellable` and `Transportable` interfaces.


Search
BoxedItem.java 1
BoxedItemTest.java 1
Photograph.java 1
PhotographTest.java 1
Sellable.java 1
Transportable.java 1
C: > Users > CCV YOUTH > OneDrive > Desktop > interFaces > BoxedItem.java > BoxedItem
Tabnine | Edit | Test | Explain | Document | Ask
Tabnine | Edit | Test | Explain | Document | Ask

```java
// BoxedItem.java
public class BoxedItem implements Sellable, Transportable {
  private String description;
  private int listPrice;
  private int lowestPrice;
  private int weight;
  private boolean hazardous;

  public BoxedItem(String description, int listPrice, int lowestPrice, int weight, boolean hazardous) {
      this.description = description;
      this.listPrice = listPrice;
      this.lowestPrice = lowestPrice;
      this.weight = weight;
      this.hazardous = hazardous;
  }

  @Override
  public String getDescription() {
      return description;
  }

  @Override
  public int getListPrice() {
      return listPrice;
  }
```


Search
BoxedItem.java 1
BoxedItemTest.java 1
Photograph.java 1
PhotographTest.java 1
Sellable.java 1
Transportable.java 1
C: > Users > CCV YOUTH > OneDrive > Desktop > interFaces > BoxedItem.java > ...
Tabnine | Edit | Test | Explain | Document | Ask

```java
public class BoxedItem implements Sellable, Transportable {
    }

    @Override
    public int getLowestPrice() {
        return lowestPrice;
    }

    @Override
    public int getWeight() {
        return weight;
    }

    @Override
    public boolean isHazardous() {
        return hazardous;
    }

    public int getInsuredValue() {
        return listPrice * 2; // Example calculation for insured value
    }
}
```

Attributes:

- `description`: Description of the boxed item (e.g., "Smartphone").
- `listPrice`: List price of the item in cents.
- `lowestPrice`: Lowest price of the item in cents.
- `weight`: Weight of the item in grams.
- `hazardous`: Boolean indicating if the item is hazardous.

Constructor:

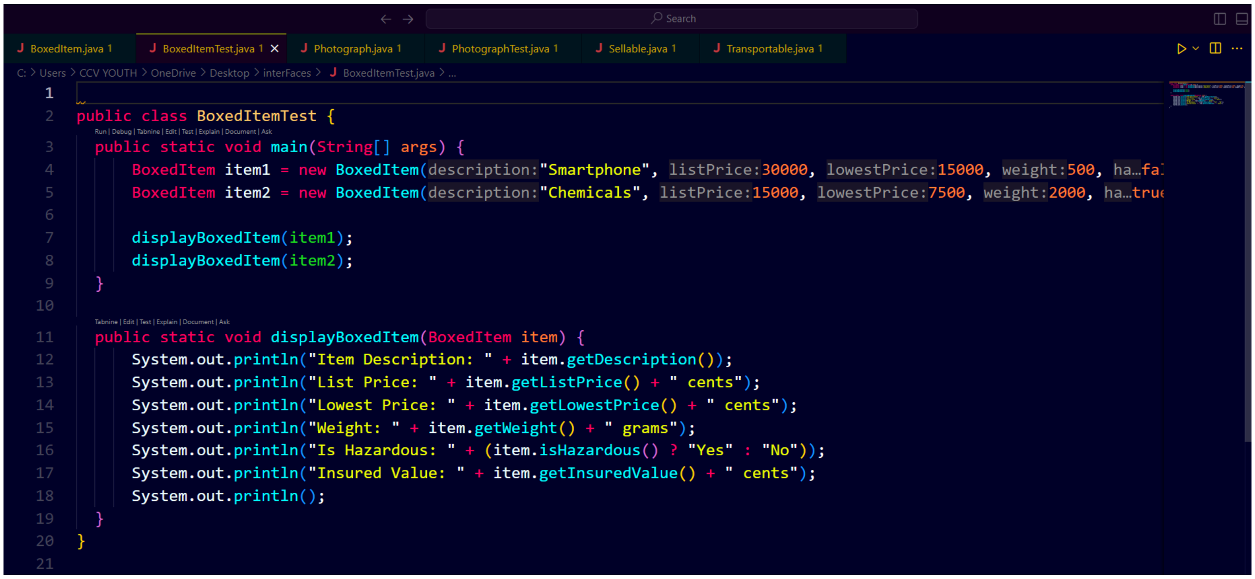
`BoxedItem(String description, int listPrice, int lowestPrice, int weight, boolean hazardous)`: Initializes the boxed item with its description, prices, weight, and hazard status.

Methods:

- `String getDescription()`: Returns the item's description.
- `int getListPrice()`: Returns the item's list price.
- `int getLowestPrice()`: Returns the item's lowest price.
- `int getWeight()`: Returns the item's weight in grams.
- `boolean isHazardous()`: Returns `true` if the item is hazardous, `false` otherwise.
- `int getInsuredValue()`: Returns the insured value of the item, calculated as twice the list price.
  Usage: Used to manage boxed items with details for both selling and transporting, including their hazardous status and insured value.

File: PhotographTest.java

Purpose: Tests the functionality of the Photograph class by creating instances and displaying their details.

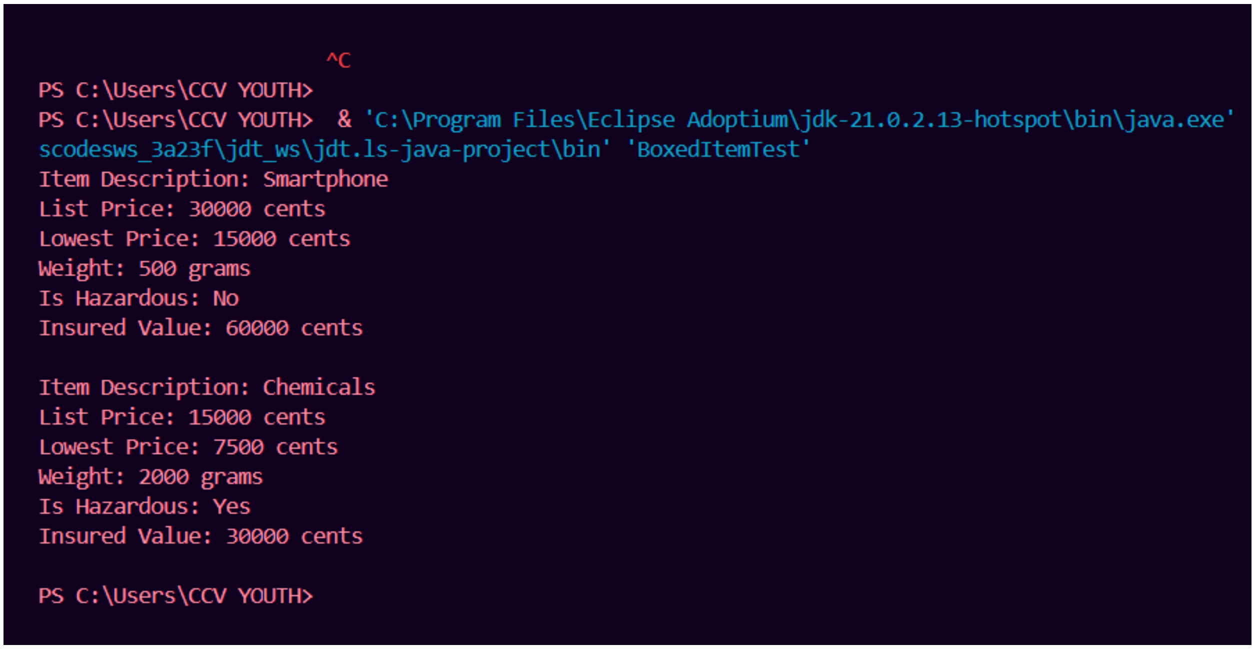

Search
BoxedItem.java 1
BoxedItemTest.java 1
Photograph.java 1
PhotographTest.java 1
Sellable.java 1
Transportable.java 1
C: > Users > CCV YOUTH > OneDrive > Desktop > interFaces > BoxedItemTest.java > ...
Run | Debug | Tabnine | Edit | Test | Explain | Document | Ask
Tabnine | Edit | Test | Explain | Document | Ask

```java

public class BoxedItemTest {
  public static void main(String[] args) {
      BoxedItem item1 = new BoxedItem(description:"Smartphone", listPrice:30000, lowestPrice:15000, weight:500, ha…fa
      BoxedItem item2 = new BoxedItem(description:"Chemicals", listPrice:15000, lowestPrice:7500, weight:2000, ha…true

      displayBoxedItem(item1);
      displayBoxedItem(item2);
  }

  public static void displayBoxedItem(BoxedItem item) {
      System.out.println("Item Description: " + item.getDescription());
      System.out.println("List Price: " + item.getListPrice() + " cents");
      System.out.println("Lowest Price: " + item.getLowestPrice() + " cents");
      System.out.println("Weight: " + item.getWeight() + " grams");
      System.out.println("Is Hazardous: " + (item.isHazardous() ? "Yes" : "No"));
      System.out.println("Insured Value: " + item.getInsuredValue() + " cents");
      System.out.println();
  }
}

```

Methods:

- public static void main(String[] args): Main method that creates two Photograph objects and calls displayPhotograph for each.
- public static void displayPhotograph(Photograph photo): Prints the description, list price, lowest price, and color status of the photograph.

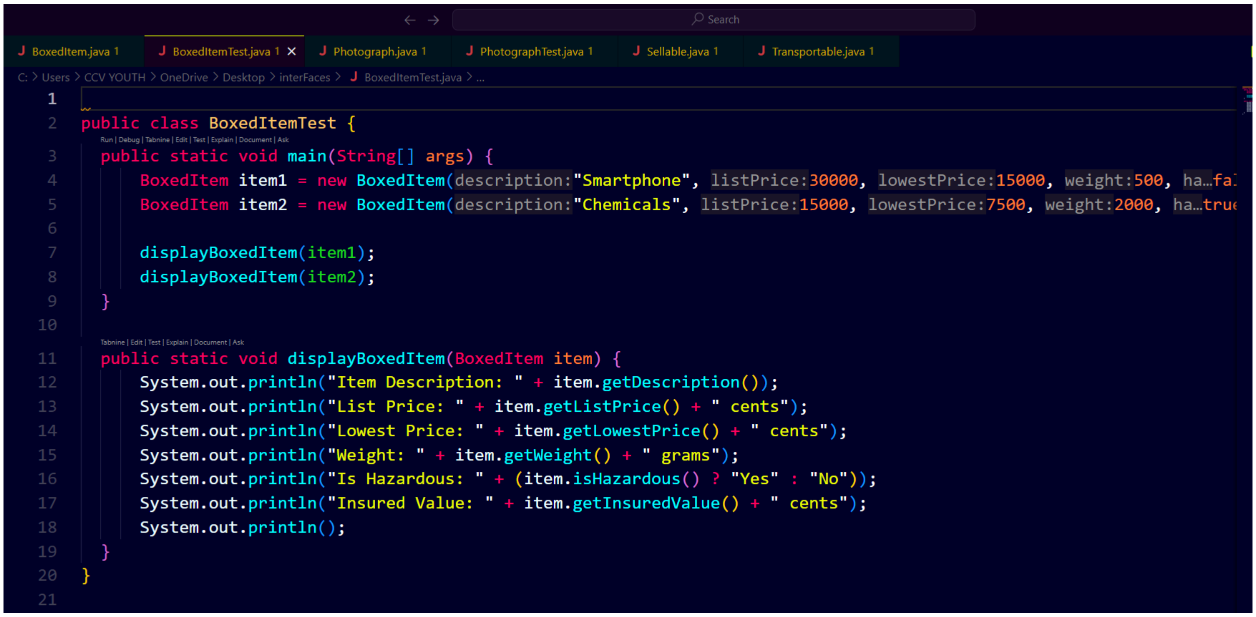
Usage: To verify that Photograph objects are created and display their attributes correctly.



```
                        ^C
PS C:\Users\CCV YOUTH>
PS C:\Users\CCV YOUTH>  & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.2.13-hotspot\bin\java.exe'
scodesws_3a23f\jdt_ws\jdt.ls-java-project\bin' 'BoxedItemTest'
Item Description: Smartphone
List Price: 30000 cents
Lowest Price: 15000 cents
Weight: 500 grams
Is Hazardous: No
Insured Value: 60000 cents

Item Description: Chemicals
List Price: 15000 cents
Lowest Price: 7500 cents
Weight: 2000 grams
Is Hazardous: Yes
Insured Value: 30000 cents

PS C:\Users\CCV YOUTH>
```

File: BoxedItemTest.java

Purpose: Tests the functionality of the BoxedItem class by creating instances and displaying their details.

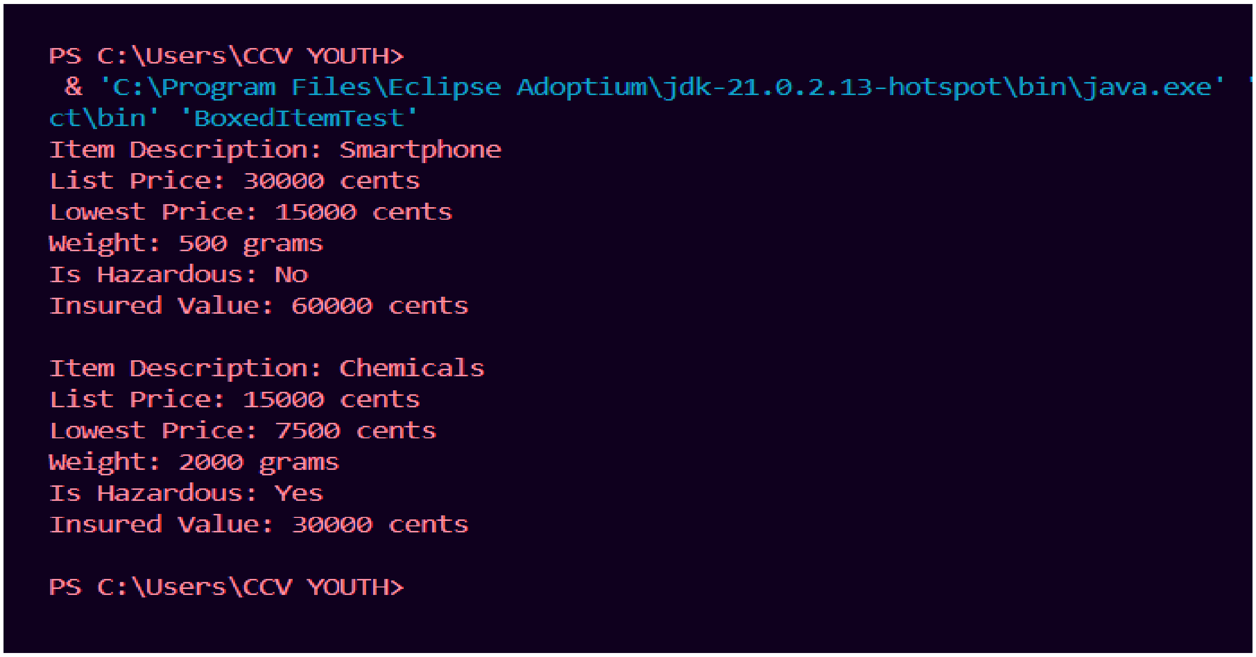

Search
BoxedItem.java 1
BoxedItemTest.java 1
Photograph.java 1
PhotographTest.java 1
Sellable.java 1
Transportable.java 1
C: > Users > CCV YOUTH > OneDrive > Desktop > interFaces > BoxedItemTest.java > ...
Run | Debug | Tabnine | Edit | Test | Explain | Document | Ask
Tabnine | Edit | Test | Explain | Document | Ask

```java

public class BoxedItemTest {
    public static void main(String[] args) {
        BoxedItem item1 = new BoxedItem(description:"Smartphone", listPrice:30000, lowestPrice:15000, weight:500, ha…fa
        BoxedItem item2 = new BoxedItem(description:"Chemicals", listPrice:15000, lowestPrice:7500, weight:2000, ha…true

        displayBoxedItem(item1);
        displayBoxedItem(item2);
    }

    public static void displayBoxedItem(BoxedItem item) {
        System.out.println("Item Description: " + item.getDescription());
        System.out.println("List Price: " + item.getListPrice() + " cents");
        System.out.println("Lowest Price: " + item.getLowestPrice() + " cents");
        System.out.println("Weight: " + item.getWeight() + " grams");
        System.out.println("Is Hazardous: " + (item.isHazardous() ? "Yes" : "No"));
        System.out.println("Insured Value: " + item.getInsuredValue() + " cents");
        System.out.println();
    }
}

```

- public static void main(String[] args): Main method that creates two BoxedItem objects and calls displayBoxedItem for each.
- public static void displayBoxedItem(BoxedItem item): Prints the description, list price, lowest price, weight, hazardous status, and insured value of the item.

Usage: To verify that BoxedItem objects are created and display their attributes correctly.

Output Example:



```
PS C:\Users\CCV YOUTH>
 & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.2.13-hotspot\bin\java.exe' '
ct\bin' 'BoxedItemTest'
Item Description: Smartphone
List Price: 30000 cents
Lowest Price: 15000 cents
Weight: 500 grams
Is Hazardous: No
Insured Value: 60000 cents

Item Description: Chemicals
List Price: 15000 cents
Lowest Price: 7500 cents
Weight: 2000 grams
Is Hazardous: Yes
Insured Value: 30000 cents

PS C:\Users\CCV YOUTH>
```