OPTION D EXAM STYLE QUESTION

The `CookieOrder` class is as follows:

```
public class CookieOrder{

      private String variety;
      private int numBoxes;

      public CookieOrder(String variety, int numBoxes){

            this.variety = variety;
            this.numBoxes = numBoxes;
      }

      public String getVariety(){
            /*to be implemented in Part A*/
      }

      public int getNumBoxes(){
            /*to be implemented in Part B*/
      }

}
```

(a) Complete the implementation of the `getVariety()` method of the `CookieOrder` class

```
public String getVariety(){
```

(b) Complete the implementation of the `getNumBoxes()` method of the `CookieOrder` class

```
public int getNumBoxes(){
```

(c) Draw a UML diagram for the `CookieOrder` class. Keep it quite small - you will draw another one here later.

The `MasterOrder` class is defined as follows:

```
public class MasterOrder{

	private CookieOrder[] orders;

	public MasterOrder(){
		orders = new CookieOrder[100];
	}

	//method to add a CookieOrder to orders
	public void addCookieOrder(CookieOrder cookie){
		/*implementation not shown*/
	}

	public int getTotalBoxes(){
		/*to be implemented in Part D*/
	}

	public int removeVariety(String variety){
		/*to be implemented in Part E*/
	}

}
```

(d) The `getTotalBoxes` method of the `MasterOrder` class processes the `orders` array and determines the total number of boxes and returns the result.
/*pre-condition: any number of elements in orders could be `null`*/

```
public int getTotalBoxes(){
```

(e) The `removeVariety` method of the `MasterOrder` class searches the `orders` array for any `CookieOrder` with the same variety as the parameter `variety`. If it finds one, it replaces that `CookieOrder` object with `null`. It returns a count of how many `CookieOrder` objects were removed from `orders`.
/*pre-condition: any number of elements in orders could already be `null`*/

```
public int removeVariety(String variety){
```

(f) Return to (c) and draw the UML diagram of the `MasterOrder` class. If there is a relationship between the `MasterOrder` class and the `CookieOrder` class, illustrate that relationship appropriately.