**Classes**

Main - the main class is responsible for the flow of the program. It is used to add volunteers, team leaders. It acts as the backbone of the program. As it provides all the data, and all the methods and classes are called from here.

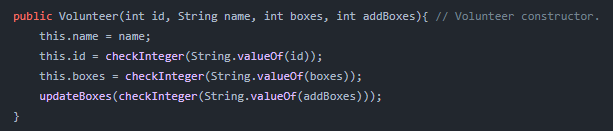
.

Team - this class's responsibility is to create a team and store all the volunteers into that team. The team class also provides methods to collect volunteer data to display team statistics, it also provides methods to sort the team alphabetical, by wage, by boxes delivered.

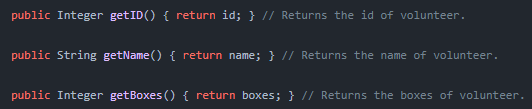
Volunteer - Volunteers are essential to the program. This class is responsible for storing volunteers data and calculating volunteer wages and boxes. It also displays volunteers using the format function within java to display the volunteers neatly.

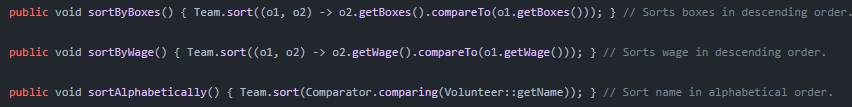
Team leader - Because the team leader class is basically the same as the volunteer class, we can inherit the methods from the volunteer class. However, team leaders get a 20% increase in the wage. So, we use the override function and the super function to get the wage and then add 20% onto the previous wage. We can also use the override function to display the team leader.

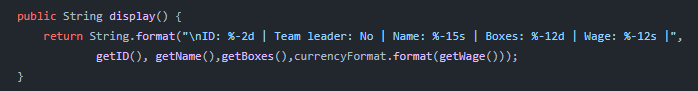
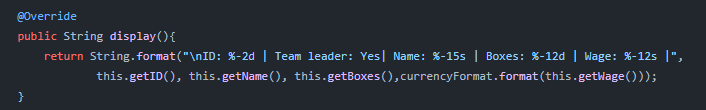
**Methods used**

Constructor - A constructor initializes the classes; they are used to add volunteer’s values when a new volunteer object is created. Each class has a constructor to get the values from previous classes.

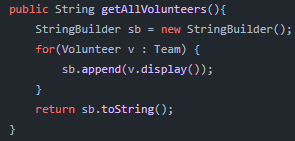
ImageUpdate boxes - To update boxes we must add the current value of the boxes to the add box value.

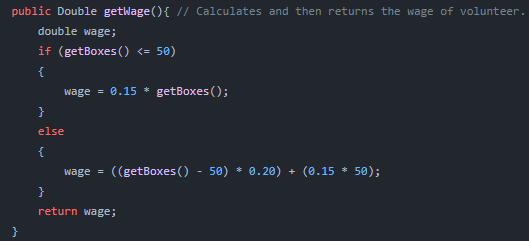
Getting Values - There are a variety of get methods these methods are used to return the value of the specified object.

Sorting volunteers - To sort volunteers we get the team Array List and use the compare to function which creates 2 new objects of the volunteer and compares both of them together.

Display - This method is used to display volunteer data and team leader data in a neat format to then be displayed as a table later. This method can be overridden to be used in multiple classes.

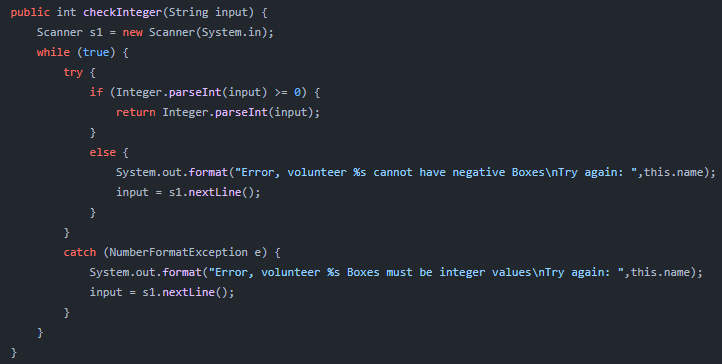
Team table - The team table loops through all the volunteer objects in the team Array List and uses the display method to print out the volunteer once this is done it will then move on to the next volunteer in the Array List on a new line.



Calculations - The brief specified that a volunteer gets 0.15p for the first 50 boxes and 0.20p for the rest. If the volunteer is a team leader, they get a 20% increase. We must first check if the volunteer box amount is equal to or less than 50 if this is true, we \* the boxes by 0.15p to get the value. If there are more than 50 boxes, we take away 50 boxes from the volunteer then we \* 0.20p by the rest of the boxes and the \* 50 by 0.15 once both of these calculations have been performed, we then add them up to get the wage of the volunteer.

Image

**Testing phase**

During this stage I came across some issues that could occur if the user enters values that are not suitable, anything below 0 is classified as an unsuitable value. To combat this, I created a method within the volunteer class that checks if the value is less than or equal to 0 if this value is unsuitable, it will give an error and ask the user to enter this value again.

Image

