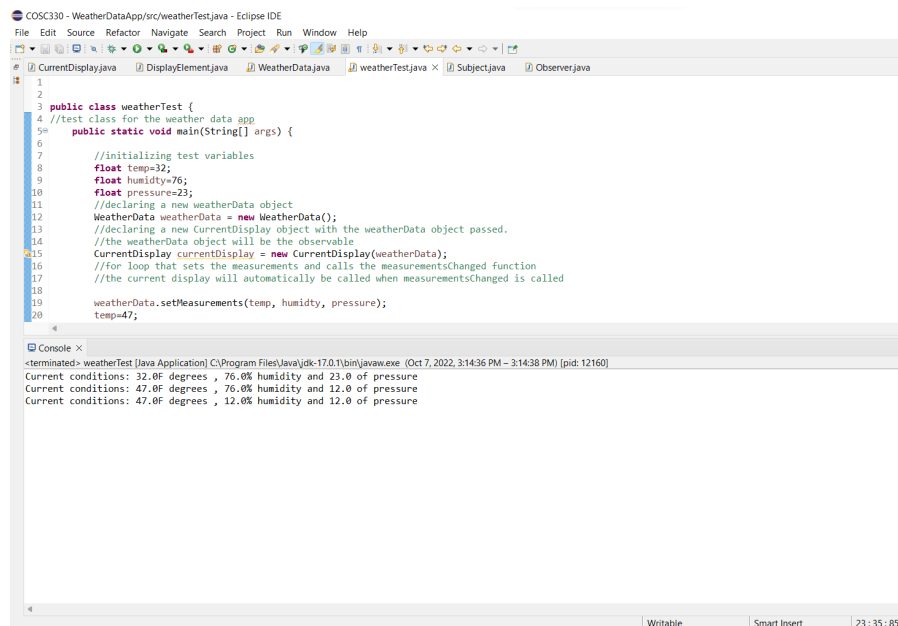


## Weather Data Application Output Screenshot:



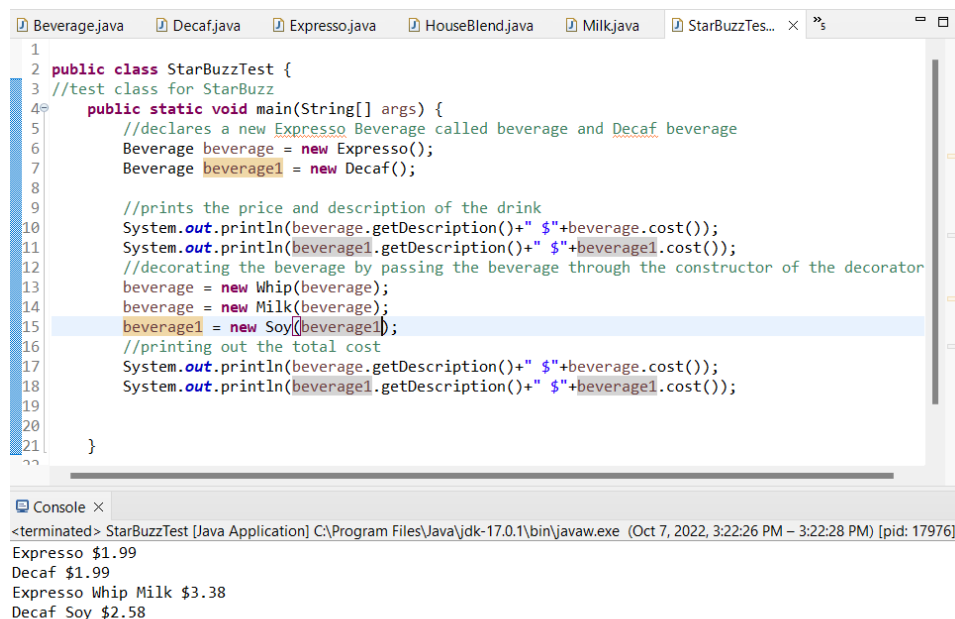
The screenshot shows the Eclipse IDE with the file `WeatherDataApp/src/weatherTest.java` open. The code is a test class for the weather data application. The console output shows the results of the `main` method, which prints the current conditions (temperature, humidity, and pressure) at three different points in time.

```
1
2
3 public class weatherTest {
4     //test class for the weather data app
5     public static void main(String[] args) {
6
7         //initializing test variables
8         float temp=32;
9         float humidity=76;
10        float pressure=23;
11        //declaring a new weatherData object
12        WeatherData weatherData = new WeatherData();
13        //declaring a new CurrentDisplay object with the weatherData object passed.
14        //the weatherData object will be the observable
15        CurrentDisplay currentDisplay = new CurrentDisplay(weatherData);
16        //for loop that sets the measurements and calls the measurementsChanged function
17        //the current display will automatically be called when measurementsChanged is called
18
19        weatherData.setMeasurements(temp, humidity, pressure);
20        temp=47;
21    }
22 }
```

Console Output:

```
<terminated> weatherTest [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Oct 7, 2022, 3:14:36 PM - 3:14:38 PM) [pid: 12160]
Current conditions: 32.0F degrees , 76.0% humidity and 23.0 of pressure
Current conditions: 47.0F degrees , 76.0% humidity and 12.0 of pressure
Current conditions: 47.0F degrees , 12.0% humidity and 12.0 of pressure
```

## StarBuzzCafe Output ScreenShot:



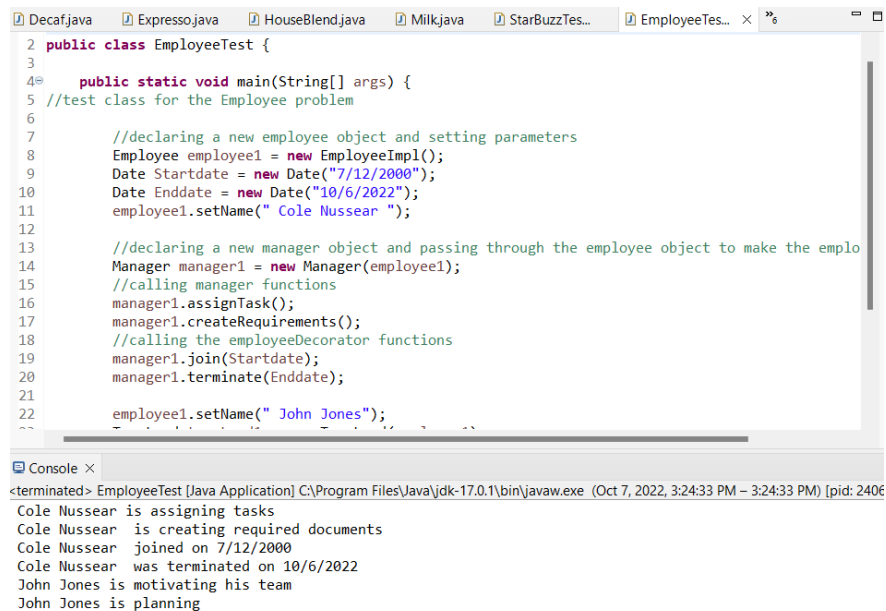
The screenshot shows the Eclipse IDE with the file `StarBuzzTest.java` open. The code is a test class for the StarBuzzCafe application. The console output shows the results of the `main` method, which prints the price and description of the drinks (Espresso, Decaf, Espresso Whip, and Decaf Soy) at three different points in time.

```
1
2 public class StarBuzzTest {
3     //test class for StarBuzz
4     public static void main(String[] args) {
5         //declares a new Espresso Beverage called beverage and Decaf beverage
6         Beverage beverage = new Espresso();
7         Beverage beverage1 = new Decaf();
8
9         //prints the price and description of the drink
10        System.out.println(beverage.getDescription()+" $"+beverage.cost());
11        System.out.println(beverage1.getDescription()+" $"+beverage1.cost());
12        //decorating the beverage by passing the beverage through the constructor of the decorator
13        beverage = new Whip(beverage);
14        beverage = new Milk(beverage);
15        beverage1 = new Soy(beverage1);
16        //printing out the total cost
17        System.out.println(beverage.getDescription()+" $"+beverage.cost());
18        System.out.println(beverage1.getDescription()+" $"+beverage1.cost());
19
20    }
21 }
```

Console Output:

```
<terminated> StarBuzzTest [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Oct 7, 2022, 3:22:26 PM - 3:22:28 PM) [pid: 17976]
Espresso $1.99
Decaf $1.99
Espresso Whip $3.38
Decaf Soy $2.58
```

## Employee Output Screenshot:



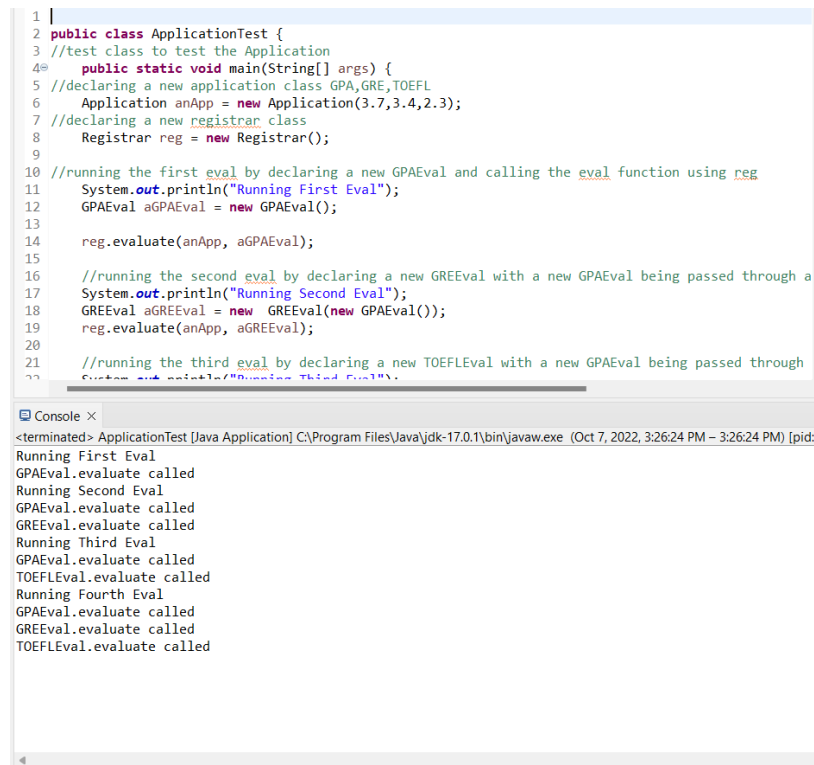
The screenshot shows an IDE with a tab for EmployeeTest.java. The code defines an EmployeeTest class with a main method that creates an EmployeeImpl object (Cole Nussear) and a Manager object, then calls various methods like assignTask, createRequirements, join, and terminate. The console output shows the execution of these methods.

```
2 public class EmployeeTest {
3
4     public static void main(String[] args) {
5         //test class for the Employee problem
6
7         //declaring a new employee object and setting parameters
8         Employee employee1 = new EmployeeImpl();
9         Date Startdate = new Date("7/12/2000");
10        Date Enddate = new Date("10/6/2022");
11        employee1.setName(" Cole Nussear ");
12
13        //declaring a new manager object and passing through the employee object to make the emplo
14        Manager manager1 = new Manager(employee1);
15        //calling manager functions
16        manager1.assignTask();
17        manager1.createRequirements();
18        //calling the employeeDecorator functions
19        manager1.join(Startdate);
20        manager1.terminate(Enddate);
21
22        employee1.setName(" John Jones");
23    }
24 }
```

Console Output:

```
<terminated> EmployeeTest [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Oct 7, 2022, 3:24:33 PM - 3:24:33 PM) [pid: 2406]
Cole Nussear is assigning tasks
Cole Nussear is creating required documents
Cole Nussear joined on 7/12/2000
Cole Nussear was terminated on 10/6/2022
John Jones is motivating his team
John Jones is planning
```

## Application Output Screenshot:



The screenshot shows an IDE with a tab for ApplicationTest.java. The code defines an ApplicationTest class with a main method that creates an Application object, a Registrar object, and then evaluates the application using GPAEval, GREval, and TOEFL Eval objects. The console output shows the execution of these evaluations.

```
1 |
2 public class ApplicationTest {
3     //test class to test the Application
4     public static void main(String[] args) {
5         //declaring a new application class GPA,GRE,TOEFL
6         Application anApp = new Application(3.7,3.4,2.3);
7         //declaring a new registrar class
8         Registrar reg = new Registrar();
9
10        //running the first eval by declaring a new GPAEval and calling the eval function using reg
11        System.out.println("Running First Eval");
12        GPAEval aGPAEval = new GPAEval();
13
14        reg.evaluate(anApp, aGPAEval);
15
16        //running the second eval by declaring a new GREval with a new GPAEval being passed through a
17        System.out.println("Running Second Eval");
18        GREval aGREval = new GREval(new GPAEval());
19        reg.evaluate(anApp, aGREval);
20
21        //running the third eval by declaring a new TOEFL Eval with a new GPAEval being passed through
22        System.out.println("Running Third Eval");
23    }
24 }
```

Console Output:

```
<terminated> ApplicationTest [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Oct 7, 2022, 3:26:24 PM - 3:26:24 PM) [pid: 2406]
Running First Eval
GPAEval.evaluate called
Running Second Eval
GPAEval.evaluate called
GREval.evaluate called
Running Third Eval
GPAEval.evaluate called
TOEFL Eval.evaluate called
Running Fourth Eval
GPAEval.evaluate called
GREval.evaluate called
TOEFL Eval.evaluate called
```

## Singleton Pattern Output:

```
1
2 public class SingletonTest {
3
4     public static void main(String[] args) {
5         //test class for the singleton
6
7         //declaring a Singleton object by calling the get instance function
8         Singleton s = Singleton.getInstance();
9
10    }
11
12 }
13
```

Problems @ Javadoc Declaration Console × Coverage

<terminated> SingletonTest [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Oct 7, 2022, 10:27:15 AM – 10:27:15 AM) [pid: 1292]  
instance is null, a new Singleton is created  
<terminated> SingletonTest [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Oct 7, 2022

## SingletonGUI output:

```
1 import java.awt.*;
2
3 //class MyFrame extends JFrame
4
5 public class MyFrame extends JFrame {
6
7     //declaring two buttons for the frame
8     JButton jButton1 = new JButton();
9     JButton jButton2 = new JButton();
10
11     //constructor for the myFrame class, just calls the init function
12     public MyFrame() {
13         init();
14     }
15
16     //test class that declares a new MyFrame and sets the visibility to true
17     public static void main(String[] args) {
18         MyFrame frame = new MyFrame();
19         frame.setSize(300,250);
20         frame.setVisible(true);
21     }
22
23     //init function sets the buttons up and then adds them to the content pane
24 }
```

Console ×

MyFrame [Java Appli] Singleton Frame. From Button 1 TimeStamp: 16651708534...

Show Singleton Frame

Show the same Singleton Frame

