

---

# Data Structures and Algorithms

**COMP 103**

**2019-20**


**Semester 2**

**Lecture 06a**

**Dr. Kerese Manueli**

**[kerese.manueli@ecs.vuw.ac.nz](mailto:kerese.manueli@ecs.vuw.ac.nz)**

**Victoria University of Wellington**



VICTORIA UNIVERSITY OF  
**WELLINGTON**  
TE HERENGA WAKA

School of  
**Engineering and Computer Science**  
Te Kura Mātai Pūkaha, Pūrorohiko

↑ XMUT103 home

Course Outline

Lecture Schedule

Weekly Timetable

Assignments

Submission

Your Marks

People

Java Resources

Java documentation

Tutor Space

Assignment Admin

Plagiarism Log

↑ [School of Engineering and Computer Science](#) ▶ [Courses/XMUT103\\_2020T1](#) ▶ Assignment2PartA

## Introduction to Data Structures and Algorithms

### Assignment 2 Part A: Using Collections

Not Available to Students

- Due 3 May , 7pm

#### Resources and links

- Download [zip file](#) containing the necessary code and data.
- Java [Documentation](#)
- [Submit](#) your answers
- [Marks and feedback](#)

#### What To Hand In

- `WellingtonTrains.java`
- `Station.java`
- `TrainLine.java`
- `TrainService.java`

Do not rename these files.

Download zip file

[https://ecs.wgtn.ac.nz/Courses/XMUT103\\_2020T1/Assignment2PartA](https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA)

## Files from the zip file:

Name	Date modified	Type	Size
data	15/04/2020 3:51 PM	File folder	
package.bluej	15/04/2020 3:51 PM	BlueJ Project File	2 KB
Station.java	15/04/2020 3:51 PM	JAVA File	2 KB
TrainLine.java	15/04/2020 3:51 PM	JAVA File	3 KB
TrainService.java	15/04/2020 3:51 PM	JAVA File	3 KB
WellingtonTrains.java	15/04/2020 3:51 PM	JAVA File	4 KB

## 25 files in the data folder:

Name	Date modified	Type	Size
fares.data	15/04/2020 3:51 PM	DATA File	1 KB
geographic-map.png	15/04/2020 3:51 PM	PNG File	91 KB
Johnsonville_Wellington-services.data	15/04/2020 3:51 PM	DATA File	2 KB
Johnsonville_Wellington-stations.data	15/04/2020 3:51 PM	DATA File	1 KB
Masterton_Wellington-services.data	15/04/2020 3:51 PM	DATA File	1 KB
Masterton_Wellington-stations.data	15/04/2020 3:51 PM	DATA File	1 KB
Melling_Wellington-services.data	15/04/2020 3:51 PM	DATA File	1 KB
Melling_Wellington-stations.data	15/04/2020 3:51 PM	DATA File	1 KB
stations.data	15/04/2020 3:51 PM	DATA File	1 KB
system-map.png	15/04/2020 3:51 PM	PNG File	70 KB
train-lines.data	15/04/2020 3:51 PM	DATA File	1 KB
Upper-Hutt_Wellington-services.data	15/04/2020 3:51 PM	DATA File	5 KB
Upper-Hutt_Wellington-stations.data	15/04/2020 3:51 PM	DATA File	1 KB
Waikanae_Wellington-services.data	15/04/2020 3:51 PM	DATA File	4 KB
Waikanae_Wellington-stations.data	15/04/2020 3:51 PM	DATA File	1 KB
Wellington_Johnsonville-services.data	15/04/2020 3:51 PM	DATA File	2 KB
Wellington_Johnsonville-stations.data	15/04/2020 3:51 PM	DATA File	1 KB
Wellington_Masterton-services.data	15/04/2020 3:51 PM	DATA File	1 KB
Wellington_Masterton-stations.data	15/04/2020 3:51 PM	DATA File	1 KB
Wellington_Melling-services.data	15/04/2020 3:51 PM	DATA File	1 KB
Wellington_Melling-stations.data	15/04/2020 3:51 PM	DATA File	1 KB

[https://ecs.wgtn.ac.nz/Courses/XMUT103\\_2020T1/Assignment2PartA](https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA)

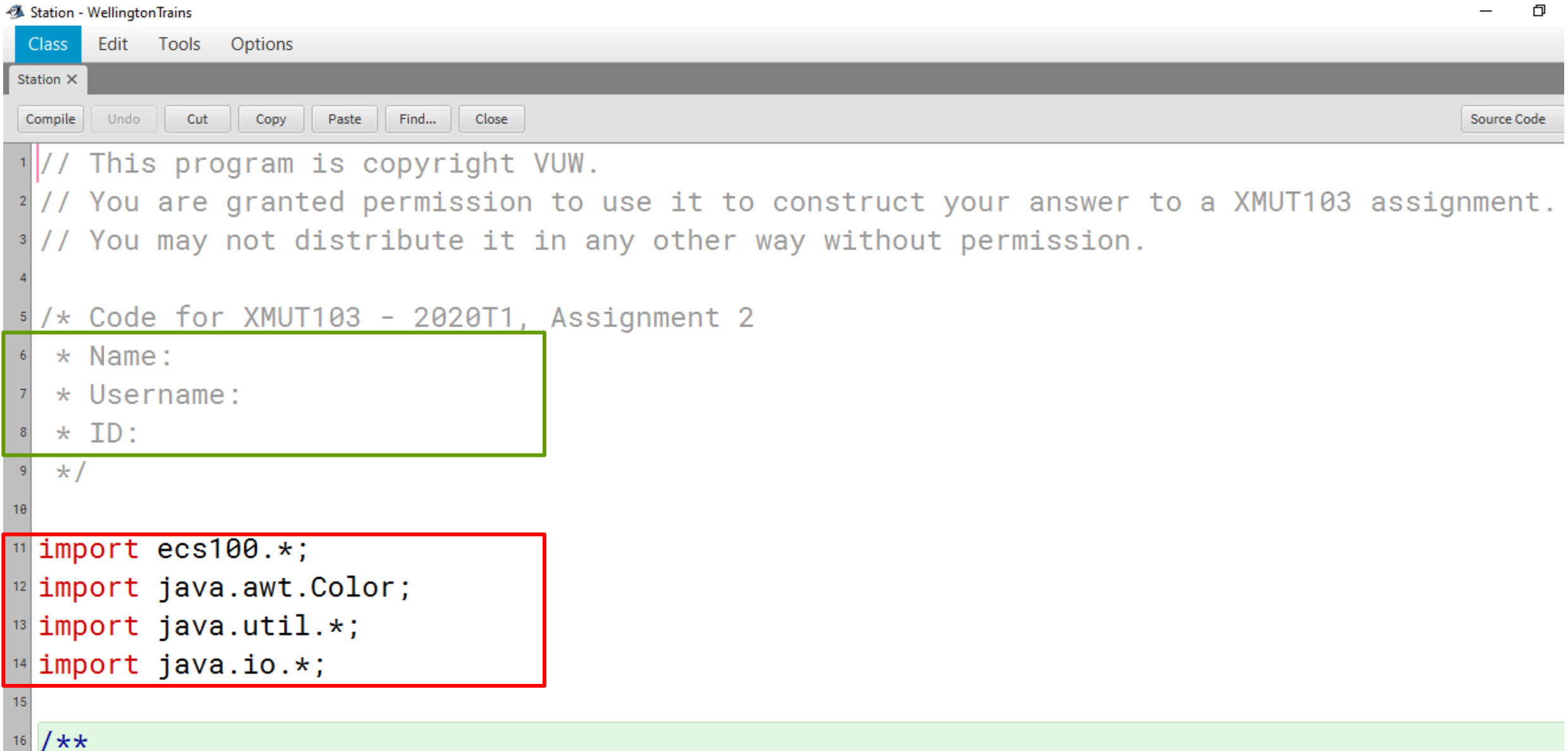
Name	Date modified	Type	Size
data	15/04/2020 3:51 PM	File folder	
package.bluej	15/04/2020 3:51 PM	BlueJ Project File	
Station.java	15/04/2020 3:51 PM	JAVA File	
TrainLine.java	15/04/2020 3:51 PM	JAVA File	
TrainService.java	15/04/2020 3:51 PM	JAVA File	
WellingtonTrains.java	15/04/2020 3:51 PM	JAVA File	

The screenshot shows the BlueJ IDE interface for a project named "WellingtonTrains". The main window displays a class hierarchy diagram with the following classes:

- WellingtonTrains** (top class, red checkered pattern)
- TrainLine** (middle class, diagonal stripes)
- Station** (bottom class, diagonal stripes)
- TrainService** (bottom class, diagonal stripes)

A red arrow originates from the `Station.java` file in the file explorer on the left and points to the `Station` class in the diagram. The IDE interface includes a menu bar (Project, Edit, Tools, View, Help), a toolbar with buttons for "New Class...", "Compile", and "Teamwork", and a "Testing" section with "Run Tests" and "End" buttons. The status bar at the bottom indicates "Initialising virtual machine... Done."

## Station.java file opened in BlueJ



```
1 // This program is copyright VUW.
2 // You are granted permission to use it to construct your answer to a XMUT103 assignment.
3 // You may not distribute it in any other way without permission.
4
5 /* Code for XMUT103 - 2020T1, Assignment 2
6  * Name:
7  * Username:
8  * ID:
9  */
10
11 import ecs100.*;
12 import java.awt.Color;
13 import java.util.*;
14 import java.io.*;
15
16 /**
```

## Station.java file opened in BlueJ

```
17 * Station
18 * Information about an individual station:
19 * - The name
20 * - The fare zone it is in (1 - 14)
21 * - The distance from the hub station (Wellington)
22 * - The set of TrainLines that go through that station.
23 * The constructor just takes the name, zone and distance;
24 * TrainLines must then be added to the station, one by one.
25 */
26
27 public class Station{
28     private String name;
29     private int zone;           // fare zone
30     private double distance;   // distance from Wellington
31     private Set<TrainLine> trainLines = new HashSet<TrainLine>();
32
33
34     public Station(String name, int zone, double dist){
35         this.name = name;
36         this.zone = zone;
37         this.distance = dist;
38     }
```

## Station.java file opened in BlueJ

```
40 public String getName(){
41     return this.name;
42 }
43
44 public int getZone(){
45     return this.zone;
46 }
47
48 /**
49  * Add a TrainLine to the station
50  */
51 public void addTrainLine(TrainLine line){
52     trainLines.add(line);
53 }
54
55 public Set<TrainLine> getTrainLines(){
56     return Collections.unmodifiableSet(trainLines); //Return an unmodifiable version of
57 }
58
59 /**
60  * toString is the station name plus zone, plus number of train lines
61  */
62 public String toString(){
63     return name+" (zone "+zone+", "+trainLines.size()+" lines)";
64 }
```

[https://ecs.wgtn.ac.nz/Courses/XMUT103\\_2020T1/Assignment2PartA](https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA)

Name	Date modified	Type	Size
data	15/04/2020 3:51 PM	File folder	
package.bluej	15/04/2020 3:51 PM	BlueJ Project File	
Station.java	15/04/2020 3:51 PM	JAVA File	
TrainLine.java	15/04/2020 3:51 PM	JAVA File	
TrainService.java	15/04/2020 3:51 PM	JAVA File	
WellingtonTrains.java	15/04/2020 3:51 PM	JAVA File	

The screenshot shows the BlueJ IDE interface for a project named "WellingtonTrains". The interface includes a menu bar with "Project", "Edit", "Tools", "View", and "Help". On the left side, there are buttons for "New Class...", "Compile", "Teamwork" (with a "Share..." button), and "Testing" (with "Run Tests", "recording", and "End" buttons). The main workspace displays four class icons: "WellingtonTrains" (top right, red checkered pattern), "TrainLine" (middle left, orange diagonal stripes), "Station" (middle right, orange diagonal stripes), and "TrainService" (bottom center, orange diagonal stripes). A red arrow originates from the "TrainLine.java" file in the file explorer on the left and points to the "TrainLine" class icon in the IDE. At the bottom of the IDE window, a status bar shows "Initialising virtual machine... Done." and a refresh button.



## TrainLine.java file opened in BlueJ

```
1 // This program is copyright VUW.
2 // You are granted permission to use it to construct your answer to a XMUT103 assignment.
3 // You may not distribute it in any other way without permission.
4
5 /* Code for XMUT103 - 2020T1, Assignment 2
6  * Name:
7  * Username:
8  * ID:
9  */
10
11 import ecs100.*;
12 import java.util.*;
13 import java.io.*;
```

## TrainLine.java file opened in BlueJ

```
16 * TrainLine
17 * Information about a Train Line.
18 * Note, we treat the outbound train line as a different from the inbound line.
19 * This means that the Johnsonville-Wellington line is a different train line from
20 * the Wellington-Johnsonville line.
21 * Although they have the same stations, the stations will be in opposite orders.
22 *
23 * A TrainLine contains
24 * - the name of the TrainLine (originating station - terminal station,
25 * - eg Wellington-Melling)
26 * - The list of stations on the line a list of TrainServices running on the line
27 * - (eg the 10:00 am service from Upper-Hutt to Wellington)
28 * - (in order of time - services earlier in the list are always earlier times
29 * - (at any station) than later services)
30 */
```

## TrainLine.java file opened in BlueJ

```
32 public class TrainLine{
33     //Fields
34     private String name;
35     private List<Station> stations = new ArrayList<Station>();           // list of stations on the line
36     private List<TrainService> trainServices = new ArrayList<TrainService>(); // set of TrainServices running on the line
37
38     //Constructor
39     public TrainLine(String name){
40         this.name = name;
41     }
42
43     // Methods to add values to the TrainLine
44     /**
45      * Add a TrainService to the set of TrainServices for this line
46      */
47     public void addTrainService(TrainService train){
48         trainServices.add(train);
49     }
50
51     /**
52      * Add a Station to the list of Stations on this line
53      */
54     public void addStation(Station station){
55         stations.add(station);
56     }
}
```

## TrainLine.java file opened in BlueJ

```
58 //Getters
59 public String getName(){
60     return name;
61 }
62
63 public List<Station> getStations(){
64     return Collections.unmodifiableList(stations); // an unmodifiable version of the list of stations
65 }
66
67 public List<TrainService> getTrainServices(){
68     return Collections.unmodifiableList(trainServices); // an unmodifiable version of the list of trainServices
69 }
70
71 /**
72  * String contains name of the train line name plus number of stations and number of services
73  */
74 public String toString(){
75     return (name+" ("+stations.size()+" stations, "+trainServices.size()+" services)");
76 }
77
78 }
```

[https://ecs.wgtn.ac.nz/Courses/XMUT103\\_2020T1/Assignment2PartA](https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA)

Name	Date modified	Type	Size
data	15/04/2020 3:51 PM	File folder	
package.bluej	15/04/2020 3:51 PM	BlueJ Project File	
Station.java	15/04/2020 3:51 PM	JAVA File	
TrainLine.java	15/04/2020 3:51 PM	JAVA File	
TrainService.java	15/04/2020 3:51 PM	JAVA File	
WellingtonTrains.java	15/04/2020 3:51 PM	JAVA File	

BlueJ: WellingtonTrains

Project Edit Tools View Help

New Class...  
→  
Compile

Teamwork  
Share...

Testing  
Run Tests  
recording  
End

WellingtonTrains

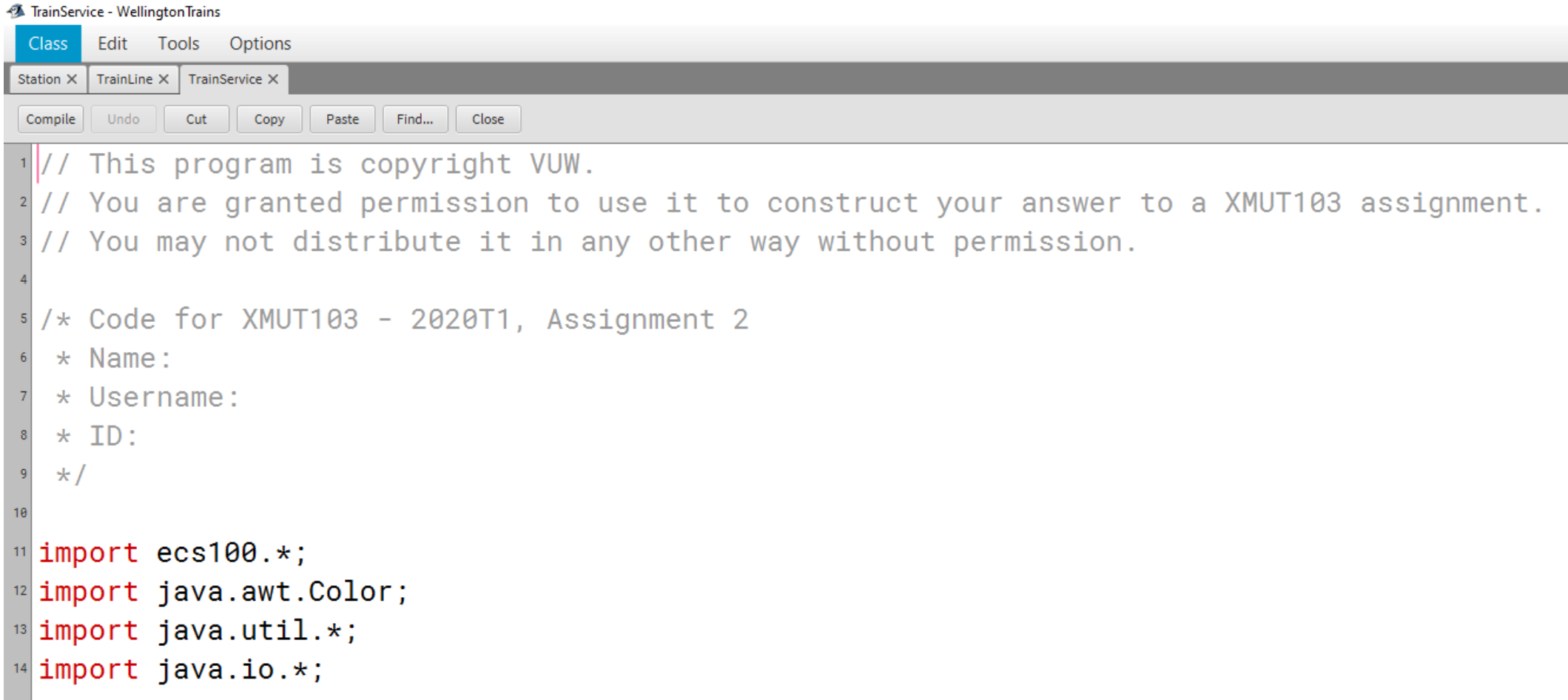
TrainLine

Station

TrainService

Initialising virtual machine... Done.

# TrainService.java file opened in BlueJ



```
1 // This program is copyright VUW.
2 // You are granted permission to use it to construct your answer to a XMUT103 assignment.
3 // You may not distribute it in any other way without permission.
4
5 /* Code for XMUT103 - 2020T1, Assignment 2
6  * Name:
7  * Username:
8  * ID:
9  */
10
11 import ecs100.*;
12 import java.awt.Color;
13 import java.util.*;
14 import java.io.*;
```

## TrainService.java file opened in BlueJ

```
16 /**
17  * TrainService
18  * A particular train service running on a train line.
19  * A train service is specified by a list of times that train leaves
20  * each station along the train line.
21  * (if the train does not stop at a station, then the corresponding time is -1)
22  * A TrainService object contains
23  * - The TrainLine that the service runs on
24  * - a ID of the train (the name of the line concatenated with the starting time of the train)
25  * - a list of times (integers representing 24-hour time, eg 1425 for 2:45pm), one for
26  * each station on the train line. A time is -1 if the train does not stop at the station.
27  * The getStart() method will return the first real time in the list of times
28  */
29
30 public class TrainService{
31     // Fields
32     private TrainLine trainLine;
33     private String trainID;    // train line name + starting time of the train
34     private List<Integer> times = new ArrayList<Integer>();
```

## TrainService.java file opened in BlueJ

```
36 //Constructor
37 /**
38  * Make a new TrainService on a particular train line.
39  */
40 public TrainService(TrainLine line){
41     trainLine = line;
42 }
43
44 //getters
45 public TrainLine getTrainLine(){
46     return trainLine;
47 }
48
49 public String getTrainID(){
50     return this.trainID;
51 }
52
53 public List<Integer> getTimes(){
54     return Collections.unmodifiableList(times); // unmodifiable version of the list of times.
55 }
```



## TrainService.java file opened in BlueJ

```
57 // Other methods.
```

```
58 /**
59  * Add the next time to the TrainService
60  */
```

```
61 public void addTime(int time){
62     times.add(time);
63     if (trainID==null && time != -1){
64         trainID = trainLine.getName()+"-"+time;
65     }
66 }
```

```
68 /**
69  * Return the start time of this Train Service
70  * -1 if no start times
71  */
```

```
72 public int getStart(){
73     for (int time : times){
74         if (time!=-1){return time;}
75     }
76     return -1;
77 }
```

```
79 /**
80  * ID plus number of stops
81  */
82 public String toString(){
83     if (trainID==null){return trainLine.getName()+"-unknownStart";}
84     int count = 0;
85     for (int time : times) {if (time!=-1) count++;}
86     return trainID+" (" +count+" stops)";
87 }
88
89 }
```