

```
Compile Messages | JGRASP Messages | Run I/O | Interactions
Stop
Clear
Copy
A
----jGRASP exec: javac -g Vehicle.java
----jGRASP: operation complete.
----jGRASP exec: javac -g VehicleManagement.java
----jGRASP: operation complete.
----jGRASP exec: javac -g Vehicle.java
----jGRASP: operation complete.
----jGRASP exec: javac -g VehicleManagement.java
----jGRASP: operation complete.
```

```
----jGRASP exec: java VehicleManagement
Menu:
1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 1
>> Enter registration number: ABC123
>> Enter make: Toyota
>> Enter year of manufacture: 2018
>> Enter value: 25000
Vehicle added successfully!

Menu:
1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 1
>> Enter registration number: XYZ456
>> Enter make: Honda
>> Enter year of manufacture: 2020
>> Enter value: 18000
Vehicle added successfully!

Menu:
1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 1
>> Enter registration number: DEF789
>> Enter make: Ford
>> Enter year of manufacture: 2015
>> Enter value: 30000
Vehicle added successfully!
```

```
Menu:
1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 2
List of vehicle details:
Reg No: ABC123, Make: Toyota, Year of Manufacture: 2018, Value: Php25000.0
Reg No: XYZ456, Make: Honda, Year of Manufacture: 2020, Value: Php18000.0
Reg No: DEF789, Make: Ford, Year of Manufacture: 2015, Value: Php30000.0

Menu:
1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 4
Sort Menu:
1. Sort by age (ascending)
2. Sort by age (descending)
3. Back to main menu
>> Enter your choice: 1
List of vehicle details sorted by age (ascending):
Reg No: DEF789, Make: Ford, Year of Manufacture: 2015, Value: Php30000.0
Reg No: ABC123, Make: Toyota, Year of Manufacture: 2018, Value: Php25000.0
Reg No: XYZ456, Make: Honda, Year of Manufacture: 2020, Value: Php18000.0

Sort Menu:
1. Sort by age (ascending)
2. Sort by age (descending)
3. Back to main menu
>> Enter your choice: 2
List of vehicle details sorted by age (descending):
Reg No: XYZ456, Make: Honda, Year of Manufacture: 2020, Value: Php18000.0
Reg No: ABC123, Make: Toyota, Year of Manufacture: 2018, Value: Php25000.0
Reg No: DEF789, Make: Ford, Year of Manufacture: 2015, Value: Php30000.0
```

```
Sort Menu:
1. Sort by age (ascending)
2. Sort by age (descending)
3. Back to main menu
>> Enter your choice: 3
Returning to main menu...

Menu:
1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 3
>> Enter the index of the vehicle to delete: 1
Vehicle deleted successfully!

Menu:
1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 2
List of vehicle details:
Reg No: XYZ456, Make: Honda, Year of Manufacture: 2020, Value: Php18000.0
Reg No: DEF789, Make: Ford, Year of Manufacture: 2015, Value: Php30000.0

Menu:
1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 5
Exiting program...

----jGRASP: operation complete.
```

github.com/ManiscanBSIT1A/Kyla-Jamela-V.-Maniscan

ManiscanBSIT1A / Kyla-Jamela-V.-Maniscan

Q Type to search

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Kyla-Jamela-V.-Maniscan

Public

Pin

Unwatch 1

Fork 0

Star 0

main 1 Branch 0 Tags

Go to file

Add file

<> Code

About

ManiscanBSIT1A

Add files via upload

89e477a · 1 minute ago

2 Commits

README.md

Initial commit

2 minutes ago

Vehicle.java

Add files via upload

1 minute ago

VehicleManagement.java

Add files via upload

1 minute ago

README

Kyla-Jamela-V.-Maniscan

Programming exercise - Vehicles

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Compile Messages

jGRASP Messages

Run I/O

Interactions

End

Clear

Help



```
----jGRASP exec: java VehicleManagement
```

Menu:

1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit

```

>>> Enter your choice: 1

```

```
Enter registration number: ABC123
```

```

Enter make: Toyota

```

Enter year of manufacture: 2018

Enter value: 25000

Vehicle added successfully!

Menu:

1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit

```

Enter your choice: 1

```

```

Enter registration number: XYZ456

```

Enter make: Honda

Enter year of manufacture: 2020

Enter value: 18000

Vehicle added successfully!

Menu:

1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit

```

Enter your choice: 1

```

```

Enter registration number: DEF789

```

Enter make: Ford

Enter year of manufacture: 2015

Enter value: 30000

Vehicle added successfully!

```
1 // Maniscan, Kyla Jamela V.
2
3 import java.util.*;
4 public class VehicleManagement{
5     public static void main(String[] args) {
6         ArrayList <Vehicle> vehicleList = new ArrayList<>();
7         Scanner scanner = new Scanner(System.in);
8     }
```

Compile Messages

jGRASP Messages

Run I/O

Interactions

End

Clear

Help



Menu:

1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit

▶▶ Enter your choice: 2

List of vehicle details:

Reg No: ABC123, Make: Toyota, Year of Manufacture: 2018, Value: Php25000.0
Reg No: XYZ456, Make: Honda, Year of Manufacture: 2020, Value: Php18000.0
Reg No: DEF789, Make: Ford, Year of Manufacture: 2015, Value: Php30000.0

Menu:

1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit

▶▶ Enter your choice: 4

Sort Menu:

1. Sort by age (ascending)
2. Sort by age (descending)
3. Back to main menu

▶▶ Enter your choice: 1

List of vehicle details sorted by age (ascending):

Reg No: DEF789, Make: Ford, Year of Manufacture: 2015, Value: Php30000.0
Reg No: ABC123, Make: Toyota, Year of Manufacture: 2018, Value: Php25000.0
Reg No: XYZ456, Make: Honda, Year of Manufacture: 2020, Value: Php18000.0

The screenshot displays the jGRASP IDE interface. The top menu bar includes File, Edit, View, Build, Project, Settings, Tools, Window, and Help. Below the menu is a toolbar with various icons for file operations and development tools. The left sidebar shows the file explorer with 'All Files' and 'This PC' views, and the 'Acer (C:)' drive is selected. The main editor window shows the code for 'VehicleManagement.java'.

```
1 // Maniscan, Kyla Jamela V.
2
3 import java.util.*;
4 public class VehicleManagement{
5     public static void main(String[] args) {
6         ArrayList <Vehicle> vehicleList = new ArrayList<>();
7         Scanner scanner = new Scanner(System.in);
8     }
```

Below the editor, the 'jGRASP Messages' tab is active, showing the execution output. The output includes a 'Sort Menu' with three options, a prompt for a choice, and a list of vehicle details sorted by age (descending). The output also shows a 'Menu' with five options, a prompt for a choice, and a confirmation message for deleting a vehicle.

End

Clear

Help

Sort Menu:

1. Sort by age (ascending)
2. Sort by age (descending)
3. Back to main menu

Enter your choice: 2

List of vehicle details sorted by age (descending):

Reg No: XYZ456, Make: Honda, Year of Manufacture: 2020, Value: Php18000.0

Reg No: ABC123, Make: Toyota, Year of Manufacture: 2018, Value: Php25000.0

Reg No: DEF789, Make: Ford, Year of Manufacture: 2015, Value: Php30000.0

Sort Menu:

1. Sort by age (ascending)
2. Sort by age (descending)
3. Back to main menu

Enter your choice: 3

Returning to main menu...

Menu:

1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit

Enter your choice: 3

Enter the index of the vehicle to delete: 1

Vehicle deleted successfully!

File Edit View Build Project Settings Tools Window Help

All Files

This PC

Acer (C:)

VehicleManagement.java

Vehicle.java

```

1 // Maniscan, Kyla Jamela V.
2
3 import java.util.*;
4 public class VehicleManagement{
5     public static void main(String[] args) {
6         ArrayList <Vehicle> vehicleList = new ArrayList<>();
7         Scanner scanner = new Scanner(System.in);
8

```

Compile Messages jGRASP Messages Run I/O Interactions

End

Clear

Help

```

1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 3
>> Enter the index of the vehicle to delete: 1
Vehicle deleted successfully!

Menu:
1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 2
List of vehicle details:
Reg No: XYZ456, Make: Honda, Year of Manufacture: 2020, Value: Php18000.0
Reg No: DEF789, Make: Ford, Year of Manufacture: 2015, Value: Php30000.0

Menu:
1. Add a vehicle
2. Display a list of vehicle details
3. Delete a vehicle
4. Sort vehicle list by age
5. Quit
>> Enter your choice: 5
Exiting program...

----jGRASP: operation complete.

```

Browse Find


Debug

Compile MessagesjGRASP MessagesRun I/OInteractions

Stop

Clear

Copy



[----jGRASP exec: javac -g Vehicle.java

[----jGRASP: operation complete.

[----jGRASP exec: javac -g VehicleManagement.java

[----jGRASP: operation complete.

[----jGRASP exec: javac -g Vehicle.java

[----jGRASP: operation complete.

[----jGRASP exec: javac -g VehicleManagement.java

[----jGRASP: operation complete.