UML and Packages

Lesson 2.3

Learning Outcomes

LO 2.3.1 **Identify** valid elements in a UML diagram

LO 2.3.2 **Design** UML diagrams based on modelled classes

LO 2.3.3 **Translate** working UML diagrams into Java code

LO 2.3.4 **Place** related classes into a package

LO 2.3.5 **Import** packages to use segregated classes

Unified Modeling Language

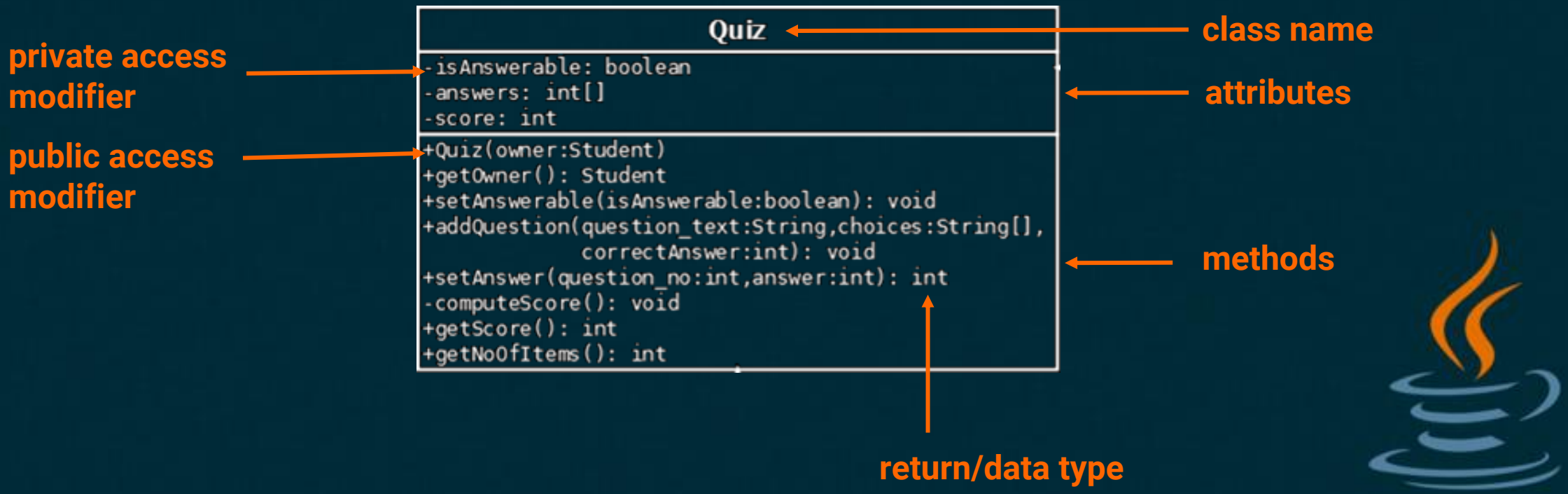
Unified Modeling Language, abbrev. UML, is a modeling language in the field of software engineering intending to provide way to visualize system designs.

In OOP, we use only the *class diagram* elements of UML.

Unified Modeling Language

The syntax for declarations:

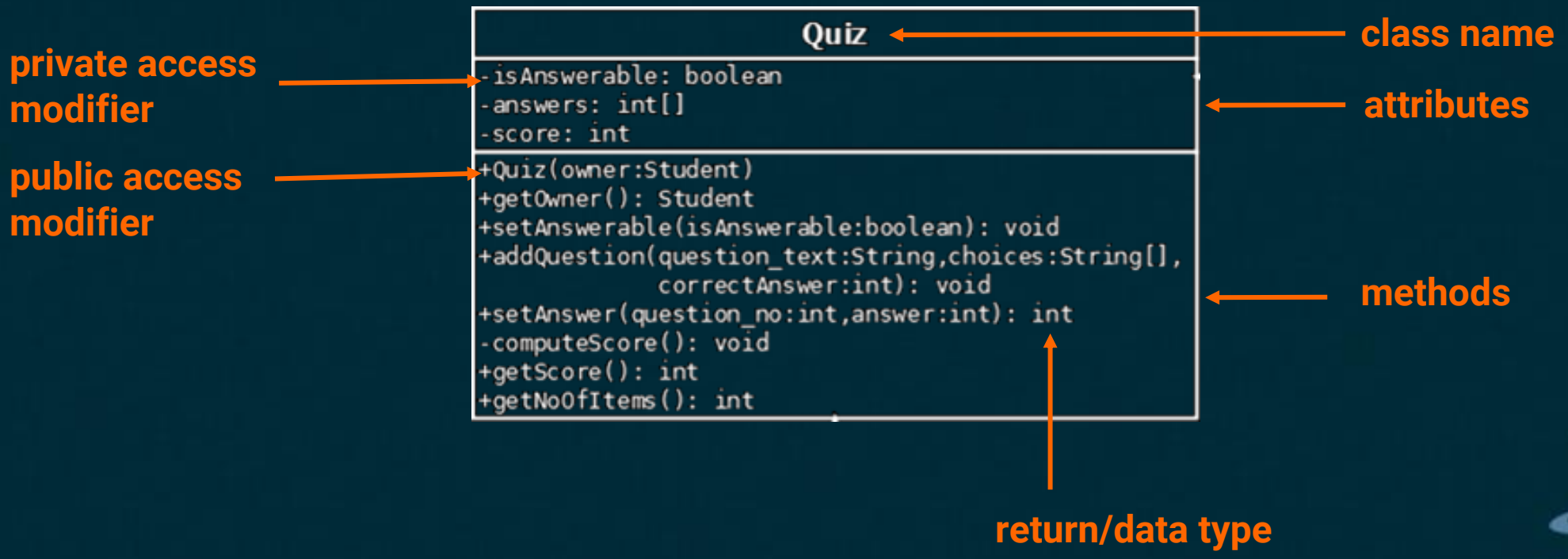
<acc_mod> <attrib_name> : <data_type> (*attribute/parameter*)



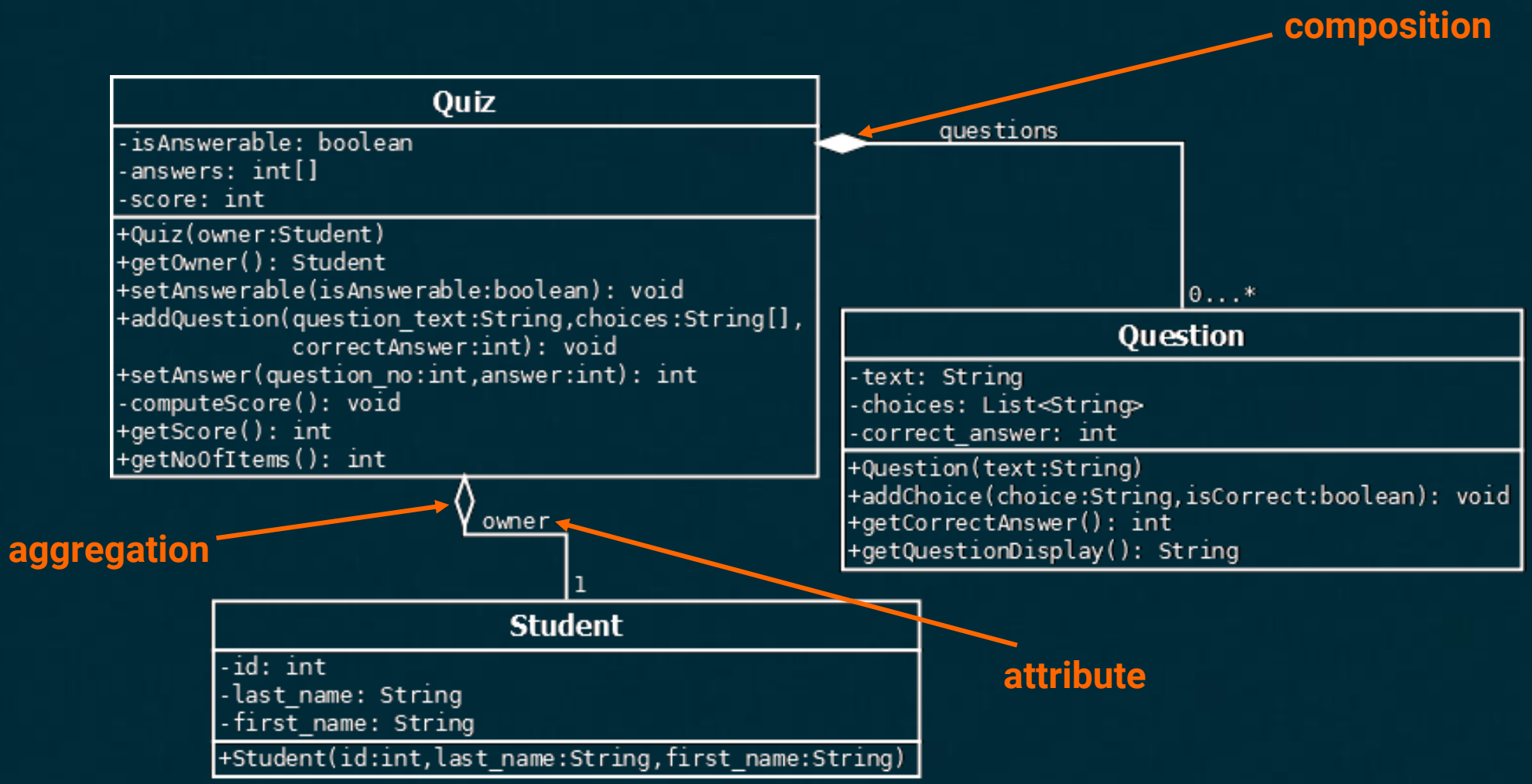
Unified Modeling Language

The syntax for declarations:

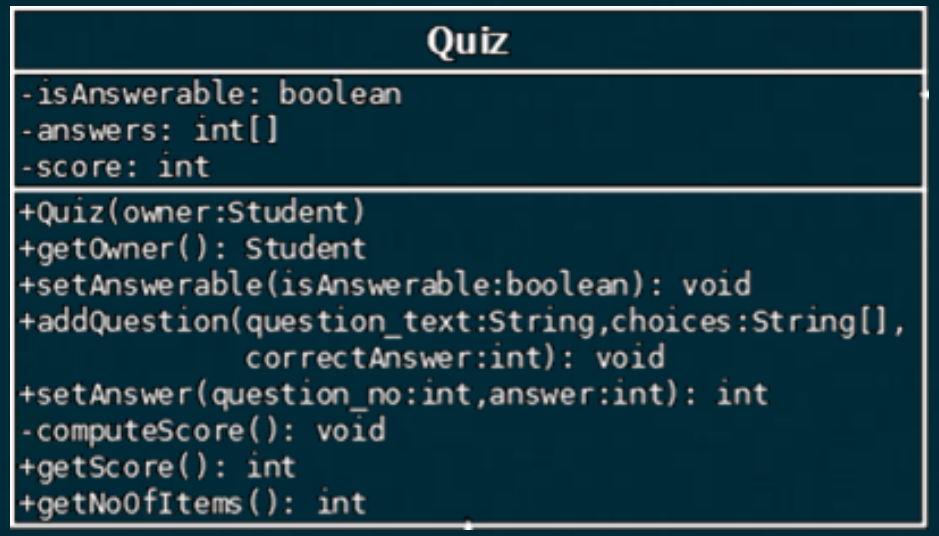
<acc_mod> <method_name> (<params>) : <ret_type> (*methods*)



Unified Modeling Language



LO 2.3.1 Identify valid elements in a UML diagram



How many attributes does the class Quiz have?

Can you create a Quiz object with a default constructor? Why or why not?

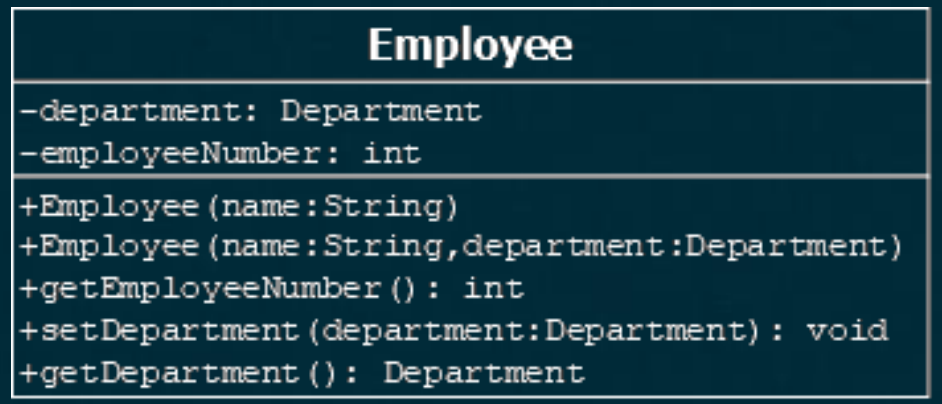
How many methods can you call from a Quiz object outside of class Quiz?

LO 2.3.1 Identify valid elements in a UML diagram

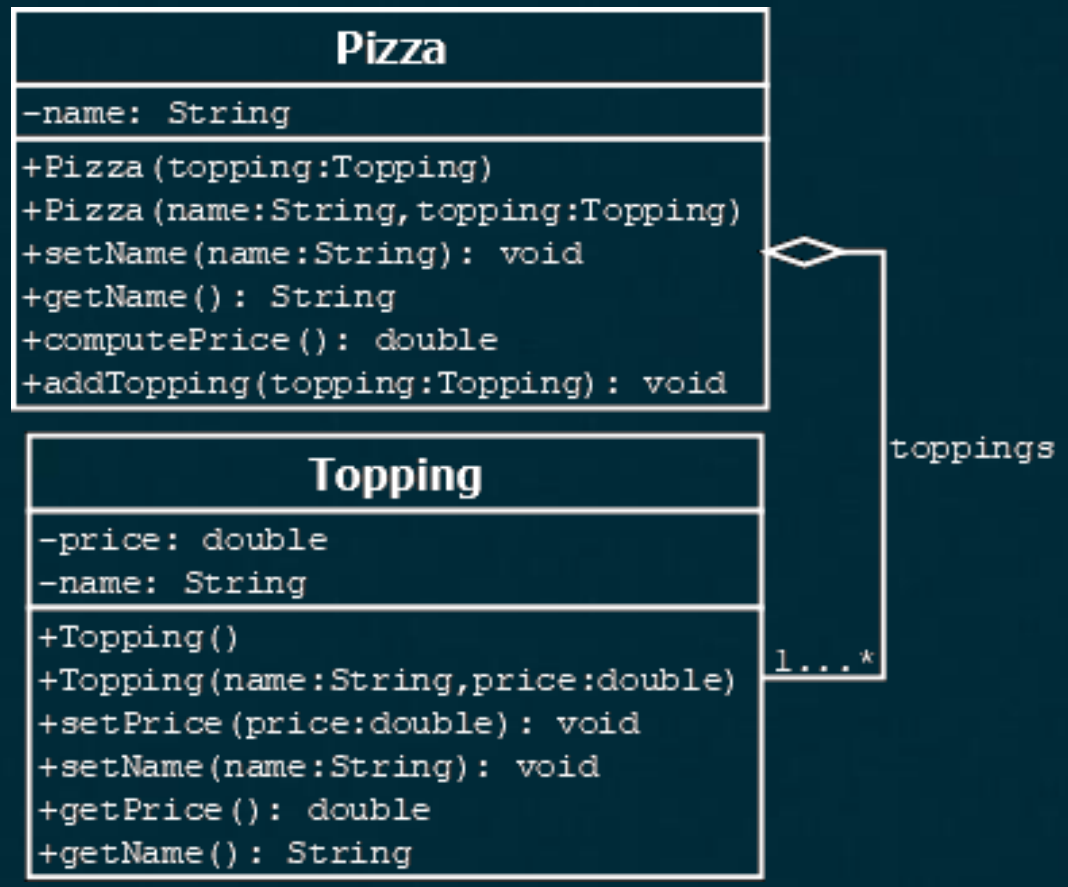
What is the name of the class?

How many accessors can you utilize in this class?

What is the datatype of the return value of the mutator of the attribute department of class Employee?



LO 2.3.1 Identify valid elements in a UML diagram



How many attributes does the class Pizza have?

What is the data type of the attribute toppings of class Pizza?

What relationship is class Topping to class Pizza?

LO 2.3.2 Design UML diagrams based on modelled classes

Design class definitions for the computing problem below and create UML class diagrams for that design.

A vehicle can contain 1 or more wheels and can have an engine or not. A wheel/tire has a volume of air while if the value of it is zero, it is considered a flat tire. A vehicle can not run if there's at least 1 flat tire.

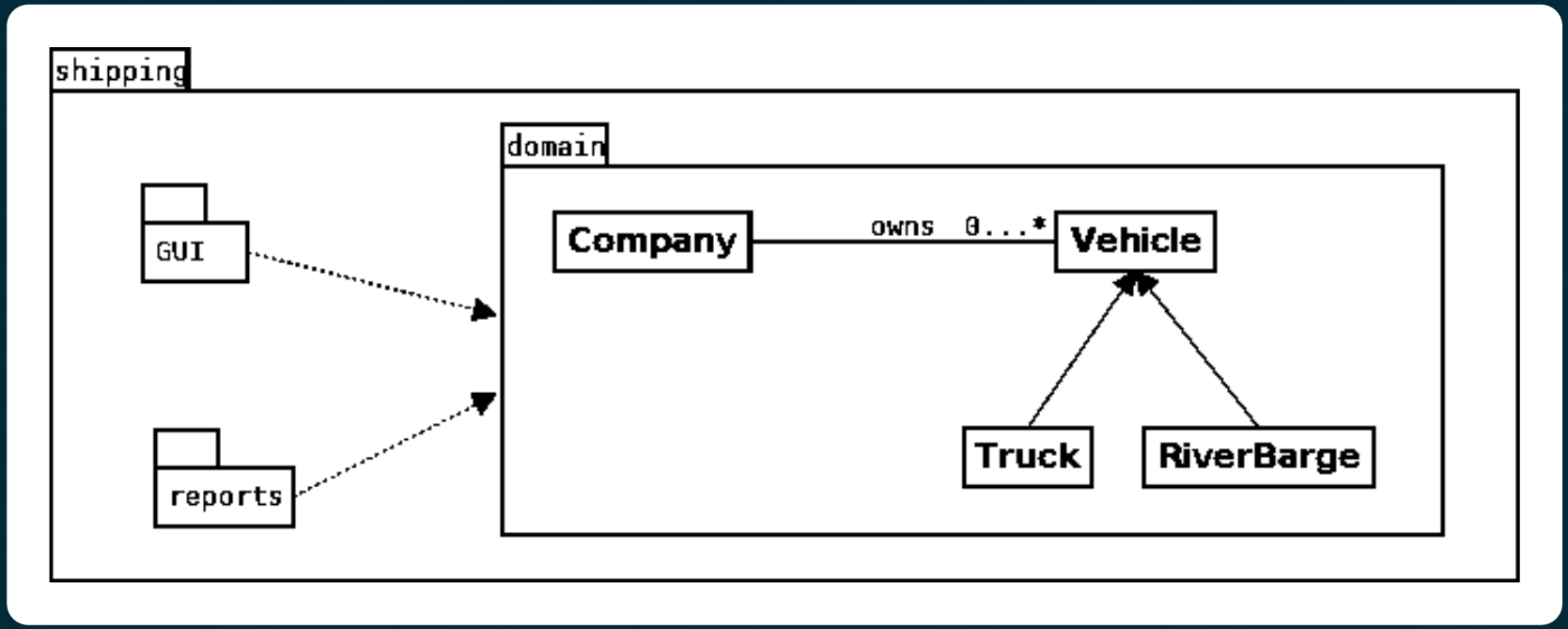
LO 2.3.3 Translate working UML diagrams into Java code

Implement the UML diagram you have just designed for the problem below in Java code.

A vehicle can contain 1 or more wheels and can have an engine or not. A wheel/tire has a volume of air while if the value of it is zero, it is considered a flat tire. A vehicle can not run if there's at least 1 flat tire.

Packages

A **software package** is where individual files, resources, and/or subpackages are packed together as a software collection that provides certain functionality as part of a larger system.



Packages

The basic syntax of assigning a package for a file:

package <top_pkg_name> [.<sub_pkg_name>]*;

Example:

```
package shipping.reports;
```

Specify the package declaration at the beginning of the source file.
One package can only be assigned per source file.
Package names must be hierarchical and separated by dot notation.

Packages

A package must be **imported** so that classes within the specified package can be used.

The basic syntax of importing an existing package:

import <package directory>.[* or <specific file name>];

Example:

```
import shipping.domain.*;
import java.util.List;
import shipping.*;
```

LO 2.3.4 Place related classes into a package

Organize the classes below into different appropriate packages of your own design.

- User
- Profile
- Message
- Notification
- Post
- Comment
- Analytics
- Setting
- ContentModeration
- Group

LO 2.3.5 Import packages to use segregated classes

With your designed packages that grouped several different classes, how do you import:

1. Post into ContentModeration
2. User into Group
3. User into Analytics
4. Setting into Profile