

Aggregation (Chapter 8)

CSE 5910

Introduction to Aggregation

Most real-life objects are compound. That is, the objects themselves are made up of other objects.

A university consists of various departments and each department has a number of professors.

A creditcard contains the name of the holder and the expiry date.

An investment consists of a stock and each stock has a stock symbol.

Introduction to Aggregation

Combine simple data into more complex data.

1959 COBOL

1972 C structures

1979 ML records

1995 Java classes

Roots of Aggregation

The notion aggregation can be traced back to the notion of records that could already be found in the programming language COBOL (COmmon Business-Oriented Language) in 1959.

In 1997, 80 percent of the world's businesses ran on COBOL and 310 billion lines of COBOL were in use.

Grace Murray Hopper

COBOL was based on the philosophy of Grace Murray Hopper that programs could be written in a language that was close to English.

The annual “Grace Murray Hopper award for outstanding young computer professionals” was established in 1971 by the Association for Computing Machinery (ACM).



Grace Murray Hopper

(1906–1992)

Definition

A class is called an *aggregate* if it has at least one attribute whose type is not primitive.

Example

The class `Stock` of the package `type.lib` is an aggregate because it has an attribute named `symbol` of type `String`.

The class `Investment` of the package `type.lib` is an aggregate because it has an attribute named `stock` of type `Stock`.

The class `Fraction` of the package `type.lib` is **not** an aggregate because all its attributes are of primitive type.

Definition

Aggregation is a binary relation on classes. The pair (A, P) of classes is in the aggregation relation if class A (aggregate) has an attribute of type P (part).

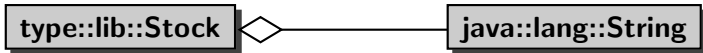
The aggregation relation is also known as the **has-a** relation. Instead of saying that (A, P) is in the aggregation relation, we often simply say that A has-a P .

Example

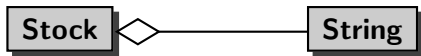
Stock has-a String.

Investment has-a Stock.

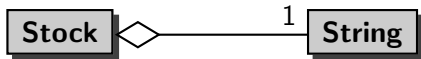
UML Diagrams



UML Diagrams



UML Diagrams



UML Diagrams



Question

How do you create a Stock object with symbol "HR.A"?

Question

How do you create a Stock object with symbol "HR.A"?

Answer

```
String symbol = new String("HR.A"); // "HR.A"  
Stock stock = new Stock(symbol);
```

Question

How do you create a Stock object with symbol "HR.A"?

Answer

```
String symbol = new String("HR.A"); // "HR.A"  
Stock stock = new Stock(symbol);
```

Question

Draw the memory diagram depicting memory at the end of the second line.

Stock Object

Answer

100	main invocation
symbol	200
stock	300
200	String object
	"HR.A"
300	Stock object
symbol	200

Investment Object

Question

How do you create an `Investment` object with three shares of HR.A stock, each of value 10.00?

Investment Object

Question

How do you create an Investment object with three shares of HR.A stock, each of value 10.00?

Answer

```
String symbol = new String("HR.A"); // "HR.A"  
Stock stock = new Stock(symbol);  
int number = 3;  
double value = 10.00;  
Investment investment = new Investment(stock, number,  
    value);
```

Investment Object

Question

How do you create an Investment object with three shares of HR.A stock, each of value 10.00?

Answer

```
String symbol = new String("HR.A"); // "HR.A"  
Stock stock = new Stock(symbol);  
int number = 3;  
double value = 10.00;  
Investment investment = new Investment(stock, number,  
    value);
```

Question

Draw the memory diagram depicting memory at the end of the fifth line.

Investment Object

100	main invocation
symbol	200
stock	300
number	3
value	10.00
investment	400
200	String object
	"HR.A"
300	Stock object
symbol	200
400	Investment object
stock	300
quantity	3
bookValue	10.00

Question

Create a random Investment object and print its stock symbol.

Question

Create a random Investment object and print its stock symbol.

Answer

```
Investment investment = Investment.getRandom();  
Stock stock = investment.getStock();  
String symbol = stock.getSymbol();  
output.println(symbol);
```

Question

Create a random Investment object and print its stock symbol.

Answer

```
Investment investment = Investment.getRandom();  
Stock stock = investment.getStock();  
String symbol = stock.getSymbol();  
output.println(symbol);
```

Answer (shorter)

```
Investment investment = Investment.getRandom();  
output.println(investment.getStock().getSymbol());
```

Question

Create a random Investment object and print its stock symbol.

Answer

```
Investment investment = Investment.getRandom();  
Stock stock = investment.getStock();  
String symbol = stock.getSymbol();  
output.println(symbol);
```

Answer (shorter)

```
Investment investment = Investment.getRandom();  
output.println(investment.getStock().getSymbol());
```

Question

Draw the memory diagram depicting memory at the end of the first line.

Answer

100	main invocation
investment	400
stock	
symbol	
200	String object
	"HR.Z"
300	Stock object
symbol	200
400	Investment object
stock	300
quantity	8
bookValue	25.50

Question

Draw the memory diagram depicting memory at the end of the third line.

Answer

100	main invocation
investment	400
stock	300
symbol	200
200	String object
	"HR.Z"
300	Stock object
symbol	200
400	Investment object
stock	300
quantity	8
bookValue	25.50

Question

Create a random Investment object and set its stock symbol "HR.B".

Mutators

Question

Create a random Investment object and set its stock symbol "HR.B".

Answer

```
Investment investment = Investment.getRandom();  
Stock stock = investment.getStock();  
stock.setSymbol("HR.B");
```

Mutators

Question

Create a random Investment object and set its stock symbol "HR.B".

Answer

```
Investment investment = Investment.getRandom();  
Stock stock = investment.getStock();  
stock.setSymbol("HR.B");
```

Answer (shorter)

```
Investment investment = Investment.getRandom();  
investment.getStock().setSymbol("HR.B");
```

Mutators

Question

Create a random Investment object and set its stock symbol "HR.B".

Answer

```
Investment investment = Investment.getRandom();  
Stock stock = investment.getStock();  
stock.setSymbol("HR.B");
```

Answer (shorter)

```
Investment investment = Investment.getRandom();  
investment.getStock().setSymbol("HR.B");
```

Question

Draw the memory diagram depicting memory at the end of the second line (of the longer answer).

Answer

100	main invocation
investment	400
stock	300
200	String object
	"HR.Z"
300	Stock object
symbol	200
400	Investment object
stock	300
quantity	8
bookValue	25.50

Question

Draw the memory diagram depicting memory at the end of the third line.

Mutators

100	main invocation
investment	400
stock	300
200	String object
	"HR.Z"
300	Stock object
symbol	500
400	Investment object
stock	300
quantity	8
bookValue	25.50
500	String object
	"HR.B"

How to Copy an Object?

We will show three ways to copy an object:

- create an alias,
- create a shallow copy, and
- create a deep copy.

The created copies are fundamentally different.

How to Create an Alias?

Question

How to create an alias of the following Investment object?

```
Investment investment = Investment.getRandom();
```

How to Create an Alias?

Question

How to create an alias of the following Investment object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Investment alias = investment;
```

How to Create an Alias?

Question

How to create an alias of the following Investment object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Investment alias = investment;
```

Question

Draw the memory diagram depicting memory at the end of the first line.

Alias

100	main invocation
investment	400
alias	
200	String object
	"HR.Z"
300	Stock object
symbol	200
400	Investment object
stock	300
quantity	8
bookValue	25.50

Question

Draw the memory diagram depicting memory at the end of the second line.

Alias

100	main invocation
investment	400
alias	400
200	String object
	"HR.Z"
300	Stock object
symbol	200
400	Investment object
stock	300
quantity	8
bookValue	25.50

How to Create a Shallow Copy?

Question

How to create a shallow copy of the following Investment object?

```
Investment investment = Investment.getRandom();
```

How to Create a Shallow Copy?

Question

How to create a shallow copy of the following Investment object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Stock stock = investment.getStock();  
int quantity = investment.getQty();  
double bookValue = investment.getBookValue();  
Investment shallowCopy = new Investment(stock,  
quantity, bookValue);
```

How to Create a Shallow Copy?

Question

How to create a shallow copy of the following Investment object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Stock stock = investment.getStock();  
int quantity = investment.getQty();  
double bookValue = investment.getBookValue();  
Investment shallowCopy = new Investment(stock,  
quantity, bookValue);
```

Question

Draw the memory diagram depicting memory at the end of the first line.

Shallow Copy

100	main invocation
investment	400
stock	
quantity	
bookValue	
shallowCopy	
200	String object
	"HR.Z"
300	Stock object
symbol	200
400	Investment object
stock	300
quantity	8
bookValue	25.50

Question

Draw only those blocks of the memory diagram that change when reaching the end of the fifth line.

Shallow Copy

100	main invocation
investment	400
stock	300
quantity	8
bookValue	25.50
shallowCopy	500
500	Investment object
stock	300
quantity	8
bookValue	25.50

How to Create a Deep Copy?

Question

How to create a deep copy of the following Investment object?

```
Investment investment = Investment.getRandom();
```

How to Create a Deep Copy?

Question

How to create a deep copy of the following Investment object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Stock stock = investment.getStock();  
String symbol = stock.getSymbol();  
int quantity = investment.getQty();  
double bookValue = investment.getBookValue();  
String symbolCopy = new String(symbol);  
Stock stockCopy = new Stock(symbolCopy);  
Investment deepCopy = new Investment(stockCopy,  
quantity, bookValue);
```


How to Create a Deep Copy?

Question

How to create a deep copy of the following Investment object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Stock stock = investment.getStock();  
String symbol = stock.getSymbol();  
int quantity = investment.getQty();  
double bookValue = investment.getBookValue();  
String symbolCopy = new String(symbol);  
Stock stockCopy = new Stock(symbolCopy);  
Investment deepCopy = new Investment(stockCopy,  
quantity, bookValue);
```

Question

Draw the memory diagram depicting memory at the end of the first line.

Deep Copy

100	main invocation
investment	400
deepCopy	
200	String object
"HR.Z"	
300	Stock object
symbol	200
400	Investment object
stock	300
quantity	8
bookValue	25.50

Question

Draw only those blocks of the memory diagram that change when reaching the end of the fifth line.

Deep Copy

100	main invocation
investment	400
deepCopy	500
500	Investment object
stock	600
quantity	8
bookValue	25.50
600	Stock object
symbol	700
700	String object
	"HR.Z"

Question

Recall that `String` objects are immutable. Is there any point of having two identical `String` objects in memory?

Deep Copy

Question

Recall that `String` objects are immutable. Is there any point of having two identical `String` objects in memory?

Answer

No. It only wastes memory.

Deep Copy

Question

Recall that `String` objects are immutable. Is there any point of having two identical `String` objects in memory?

Answer

No. It only wastes memory.

Question (revisted)

How to create a deep copy of the following `Investment` object?
`Investment investment = Investment.getRandom();`

Deep Copy

Question

Recall that `String` objects are immutable. Is there any point of having two identical `String` objects in memory?

Answer

No. It only wastes memory.

Question (revisted)

How to create a deep copy of the following `Investment` object?
`Investment investment = Investment.getRandom();`

Answer (improved)

```
Investment deepCopy = new Investment(  
    new Stock(investment.getStock().getSymbol()),  
    investment.getQty(),  
    investment.getBookValue());
```

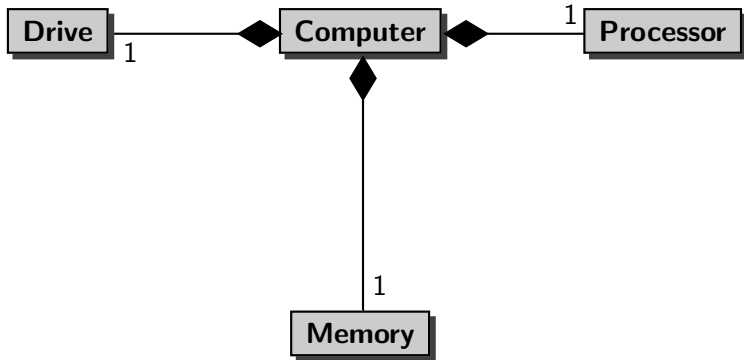

Composition

Composition is a special type of aggregation. The aggregate A and its part P form a composition if “ A owns P ”, that is, each object of type A has exclusive access to its attribute of type P .

The designer and the implementer of a class determine whether an aggregation is a composition.

Java does not provide any special language constructs for implementing compositions. The constructors, accessors and mutators are implemented in a particular way.

UML Diagrams



Question

Create a Computer object and set its processor to Intel Core i5.

Question

Create a Computer object and set its processor to Intel Core i5.

Answer

```
Computer computer = new Computer();  
Processor processor = new Processor("Intel Core i5");  
computer.setProcessor(processor);
```

Question

Create a Computer object and set its processor to Intel Core i5.

Answer

```
Computer computer = new Computer();  
Processor processor = new Processor("Intel Core i5");  
computer.setProcessor(processor);
```

Question

Draw the memory diagram depicting memory at the end of the second line.

Mutators

100	main invocation
computer	500
processor	600
200	Drive object
300	Memory object
400	Processor object
500	Computer object
drive	200
memory	300
processor	400
600	Processor object

Question

Draw the memory diagram depicting memory at the end of the third line.

Mutators

100	main invocation
computer	500
processor	600
200	Drive object
300	Memory object
400	Processor object
500	Computer object
drive	200
memory	300
processor	700
600	Processor object
700	Processor object

matlabcontrol allows us to use MATLAB in our Java code. All we need is

- the jar file matlabcontrol-4.1.0.jar, and
- its API.

Using MATLAB in Java

- Connect with MATLAB;
- Convert data from Java to MATLAB;
- Execute MATLAB code;
- Convert data from MATLAB to Java.

Connect with MATLAB

```
import matlabcontrol.MatlabProxyFactory;
import matlabcontrol.MatlabProxy;
import matlabcontrol.MatlabConnectionException;
import matlabcontrol.MatlabInvocationException;

public class Test {
    public static void main(String[] args) throws
        MatlabConnectionException, MatlabInvocationException
    {
        MatlabProxyFactory factory = new MatlabProxyFactory();
        MatlabProxy proxy = factory.getProxy();

        // use MATLAB

        proxy.disconnect();
    }
}
```

Execute MATLAB code

```
proxy.eval("fprintf('hello world');");
```

Convert data from Java to MATLAB: simple

```
proxy.setVariable("birthYear", 1995);
```

Execute MATLAB code

```
proxy.eval("now = clock;");  
proxy.eval("currentYear = now(1);");  
proxy.eval("age = currentYear - birthYear;");
```

Convert data from MATLAB to Java: simple

```
Object result = proxy.getVariable("age");
```

Convert data from MATLAB to Java: simple

```
double[] result = (double[]) proxy.getVariable("age");  
double age = result[0];  
output.printf("%.0f\n", age);
```


Convert data from Java to MATLAB: arrays

```
double[][] matrix = {{ 1, 2, 3 },  
                     { 4, 5, 6 },  
                     { 7, 8, 9 }};  
MatlabTypeConverter converter =  
    new MatlabTypeConverter(proxy);  
MatlabNumericArray array =  
    new MatlabNumericArray(matrix, null);  
converter.setNumericArray("matrix", array);
```

Convert data from Java to MATLAB:arrays

```
double[] [] matrix = {{ 1, 2, 3 },
                       { 4, 5, 6 },
                       { 7, 8, 9 }};
MatlabTypeConverter converter =
    new MatlabTypeConverter(proxy);
MatlabNumericArray array =
    new MatlabNumericArray(matrix, null);
converter.setNumericArray("matrix", array);
proxy.eval("matrix = transpose(matrix);");
```

Convert data from MATLAB to Java: array

```
matrix =  
    converter.getNumericArray("matrix").getRealArray2D();  
  
for (int r = 0; r < matrix.length; r++) {  
    for (int c = 0; c < matrix[r].length; c++) {  
        output.printf("%.0f ", matrix[r][c]);  
    }  
    output.println();  
}
```