

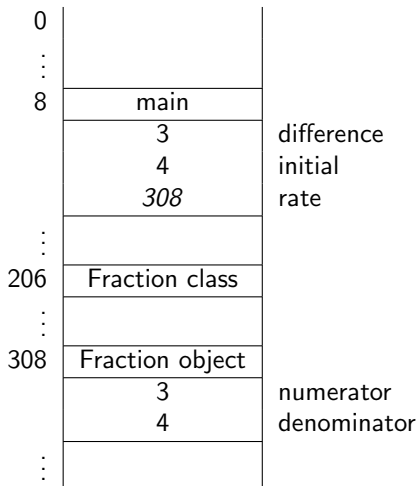
Aggregation (Chapter 8)

CSE 5910

www.cse.yorku.ca/course/5910

Classes and Objects

```
long difference = 3;  
long initial = 4;  
Fraction rate = new Fraction(difference, initial);
```



Classes and Objects

A `Fraction` object encapsulates two pieces of data: the numerator and denominator. Note that both are of primitive type.

Question

Can we combine data of non-primitive type into an object?

Classes and Objects

A `Fraction` object encapsulates two pieces of data: the numerator and denominator. Note that both are of primitive type.

Question

Can we combine data of non-primitive type into an object?

Answer

Yes. This is known as **aggregation**.

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

Examples

- 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).

She appeared on Late Night with David Letterman.



Grace Murray Hopper

(1906–1992)

Source: James S. Davis

Definition

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

Examples

- 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

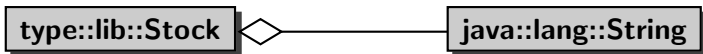
Aggregation is a binary relation on classes. The pair (A, P) of classes is in the aggregation relation if class A (aggregate) has a non-static attribute of type P (part).

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 .

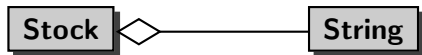
Examples

- Stock has-a String.
- Investment has-a Stock.

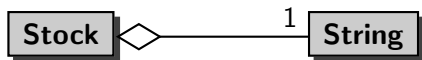
UML Diagrams



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 stock
	200	
	300	
200	String object	value
	"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	
	<i>200</i>	symbol
	<i>300</i>	stock
	3	number
	10.00	value
	<i>400</i>	investment
200	String object	
	"HR.A"	value
300	Stock object	
	<i>200</i>	symbol
400	Investment object	
	<i>300</i>	stock
	3	quantity
	10.00	bookValue

Question

What is an accessor?

Question

What is an accessor?

Answer

A method which returns the value of an attribute.

Question

What is an accessor?

Answer

A method which returns the value of an attribute.

Question

What is the name of the accessor for the attribute `symbol1`?

Question

What is an accessor?

Answer

A method which returns the value of an attribute.

Question

What is the name of the accessor for the attribute `symbol1`?

Answer

`getSymbol1`.

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

Question

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

Answer

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

Question

What is a mutator?

Mutators

Question

What is a mutator?

Answer

A method which changes the value of an attribute.

Mutators

Question

What is a mutator?

Answer

A method which changes the value of an attribute.

Question

What is the name of the mutator for the attribute `symbol`?

Mutators

Question

What is a mutator?

Answer

A method which changes the value of an attribute.

Question

What is the name of the mutator for the attribute `symbol`?

Answer

`setSymbol`.

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	
	<i>400</i>	investment
	<i>300</i>	stock
200	String object	
	"HR.Z"	value
300	Stock object	
	<i>200</i>	symbol
400	Investment object	
	<i>300</i>	stock
	8	quantity
	25.50	bookValue

Question

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

Mutators

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

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.

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

Question

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

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

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

Question

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

Shallow Copy

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

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	
	<i>400</i>	investment deepCopy
200	String object	
	"HR.Z"	value
300	Stock object	
	<i>200</i>	symbol
400	Investment object	
	<i>300</i>	stock
	8	quantity
	25.50	bookValue

Question

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

Deep Copy

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

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 (revisited)

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 (revisited)

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.



CreditCard Object

Question

Create a `CreditCard` object with number 123456 and name Virginia Kaarthouer.

CreditCard Object

Question

Create a CreditCard object with number 123456 and name Virginia Kaarthouer.

Answer

```
int number = 123456;  
String name = "Virginia Kaarthouer";  
CreditCard card = new CreditCard(number, name);
```

CreditCard Object

Question

Create a CreditCard object with number 123456 and name Virginia Kaarhouer.

Answer

```
int number = 123456;  
String name = "Virginia Kaarhouer";  
CreditCard card = new CreditCard(number, name);
```

Question

Draw the memory diagram depicting memory at the end of the second line. (To save space, do not include the attributes balance and limit.)

CreditCard Object

100	main invocation	
	123456	number
	200	name
	500	card
200	String object	
	"Virginia Kaarthouer"	value
300	Date object	
	1415637359054	time
400	Date object	
	1478795881318	time
500	CreditCard object	
	123456	number
	200	name
	300	issueDate
	400	expiryDate