

Java Application Design

Fall 2020 - Week 3

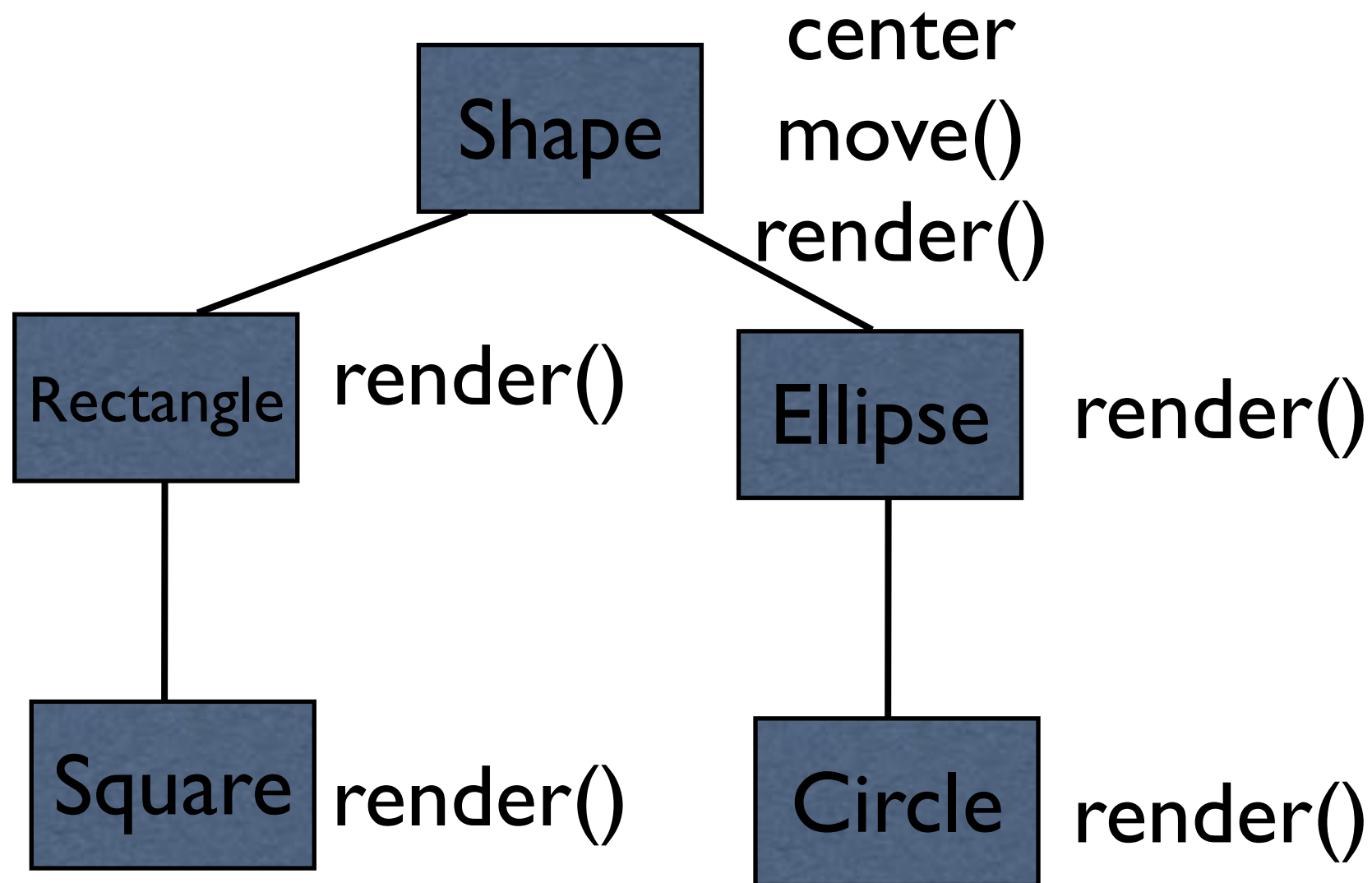
Containers

Weng Kai

Upcast

- is taking an object handle and treating it as the handle of its base type.
- The relationship between sub and base classes is :The new class is a type of the existing class.
- Case:Wind.java

Conceptual model



Method call binding

- is connecting a method call to a method body:
Which function is to be called?
- Static binding: call the function as the code
- Dynamic binding: call the function of the object
- Case: Shapes.java

Override

- Methods in super and sub classes have same name AND same parameters.
- NOT for fields, ctors. and private methods.
- Case: Override.java

Abstract classes and methods

- Some class is to create a common interface for all the classes derived from it.
- An abstract method is incomplete. It has only a declaration and no method body.
- A class containing abstract methods is called an abstract class.
- Case : Music4.java

Interface

totally abstract class

- All methods in interface are public.
- All data members in interface are public static final.

Interface

- To define an interface:

```
public interface InterfaceName extends BaseInterfaces
```

- To inherit from an interface:

```
class ClassName implements interfaces
```

- Case: Music5.java

Use of interface

- An interface can be used as a class

Case: Store.java

- An interface can inherit from more than one interfaces, but not class.

Case: HorrorShow.java

- A class can implement more than one interfaces.

Case: Adventure.java

Inner classes

- Since Java 1.1, it is possible to place a class definition within another class definition.
- Case: Parcel1.java
- Case: Parcel3.java

Where are inner classes?

- A class defined within a method.
Destination.java Contents.java Wrapping.java
Parcel4.java
- A class defined within a scope inside a method. Parcel5.java
- An anonymous class implementing an interface. Parcel6.java

- An anonymous class extending a class that has a non-default constructor. Parcel7.java
- An anonymous class that perform fields initialization. Parcel8.java
- An anonymous class that perform construction using instance initialization. Parcel9.java

Outter class?

- As a member, the inner class can access everything of its outter class.
- case: Sequence.java

Override inner?

- case: BigEgg.java

Enums

- `enum Season { WINTER, SPRING, SUMMER, FALL }`
- Sample: Deal.java

\$ java Deal 4 5

[FOUR of HEARTS, NINE of DIAMONDS, QUEEN of SPADES, ACE
of SPADES, NINE of SPADES]

[DEUCE of HEARTS, EIGHT of SPADES, JACK of DIAMONDS, TEN
of CLUBS, SEVEN of SPADES]

[FIVE of HEARTS, FOUR of DIAMONDS, SIX of DIAMONDS, NINE
of CLUBS, JACK of CLUBS]

[SEVEN of HEARTS, SIX of CLUBS, DEUCE of DIAMONDS,
THREE of SPADES, EIGHT of CLUBS]

Enum with operations

- Example: Planet.java

```
$ java Planet 175
```

```
Your weight on MERCURY is 66.107583
```

```
Your weight on VENUS is 158.374842
```

```
Your weight on EARTH is 175.000000
```

```
...
```

More enum

- Example: Operation.java

Static Import

- `double r = Math.cos(Math.PI * theta);`
- `import static java.lang.Math.PI;`
- `import static java.lang.Math.*;`
- `double r = cos(PI * theta);`

number format in 1.7

- 145_345.23
- 0b11001011