Introduction to computer programming A
LAB10

贾艳红 Jana
Email:jiayh@mail.sustech.edu.cn

LAB OBJECTIVES

- **Learn inheritance.**
- **Learn protected keyword.**
- **Learn to use enum types**

knowledge points

Enumerated Types

- ✓ **Enumeration** means **a list of named constant**.**It is created using enum keyword**.
- ✓ **Enumerated type is type-safe, meaning that an attempt to assign a value other than one of the enumerated values or null will result in a compile error.**
- ✓ **Each enumeration constant is public, static and final by default.**
- ✓ you can use the following methods on an enumerated object:
  - public String name(); Returns a name of the value for the object
  - public int ordinal(); Returns the ordinal value associated with the enumerated value.
- ✓ In Java, enumeration defines a **class type** and have constructors, you **do not instantiate an enum using new.**
- ✓ The set of every element can be get by the method(**values()**),

```
public class EnumeratedTypeDemo {
    public static void main(String[] args) {
        EnumDay x = EnumDay.MONDAY;
        switch (x) {
        case TUESDAY:
            System.out.println("learning in class");
            break;
        case WEDNESDAY:
        case THURSDAY:
            System.out.println("practice in class ");
            break;
        default:
            System.out.println("practice after class");
        }
        for (EnumDay day : EnumDay.values()) {
            System.out.println("day's name is " + day.name());
            System.out.println("day's ordinal is " + day.ordinal());
        }
        EnumDay day1 = EnumDay.FRIDAY;
        EnumDay day2 = EnumDay.THURSDAY;
        System.out.println("day1.equals(day2) returns " + day1.equals(day2));
        System.out.println("day1.toString() returns " + day1.toString());
        System.out.println("day1.compareTo(day2) returns " + day1.compareTo(day2));
    }
}
enum EnumDay {
    SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
};
```

Usingswitch Statements with an Enumerated Variable

Variables of Enumeration can be defined directly without any new keyword

Defined simpley by creating a list of enum variable

- ✓ An enumerated type is **a subclass of the Object class** and **the Comparable interface**, you can invoke the methods **equals, toString, and compareTo** from an enumerated object reference variable

Enumerated Types

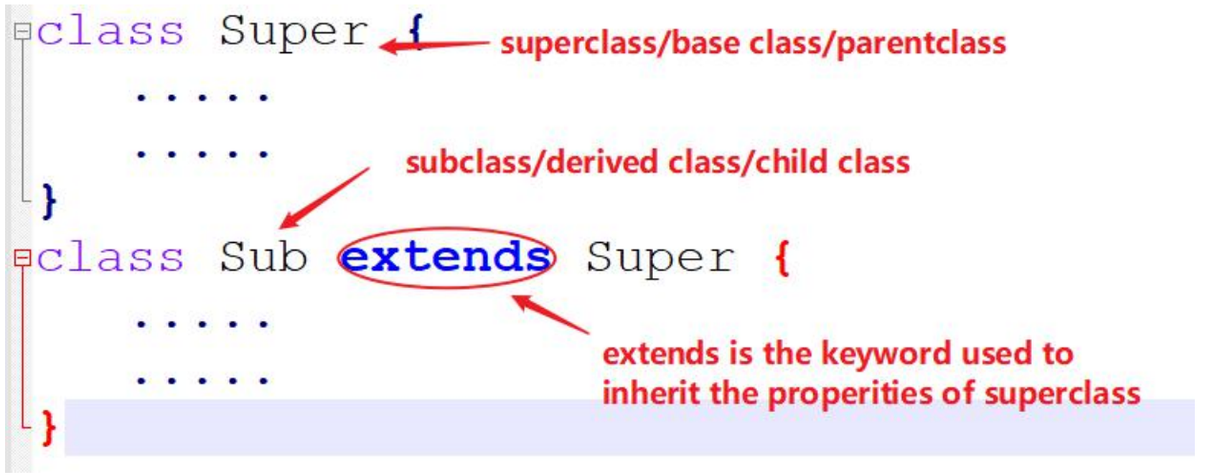
- ✓ **In Java, enumeration defines a class type. so it can have constructors, methods and instance variables.**
  - **An private (by default) constructor** :While there is attribute defined for enum, there must be a private constructor with parameters for initialization on attributes
  - **There could be also get method , toString()** could also be override, other normal method could be nonstatic or static depends on the design.

```
public class EnumeratedTypeDemo2 {
    public static void main(String[] args) {
        TrafficLight light = TrafficLight.RED;
        System.out.println(light.getDescription());
    }
}

enum TrafficLight {
    RED ("Please stop"), GREEN ("Please go"),
    YELLOW ("Please caution");
                          Remember to end with a semicolon
     private String description;

     private TrafficLight(String description) {
     this.description = description;
     }

      public String getDescription() {
      return description;
      }
};
```

Inheritance

Inheritance in Java is an important concept of OOP(Object Oriented Programming). it is the mechanism in java by which one class is allow to inherit the features(fields and methods) of another class.

Inheritance Syntax:



Why use inheritance?

- For Code Reusability.
- For Method Overriding (so runtime polymorphism can be achieved).

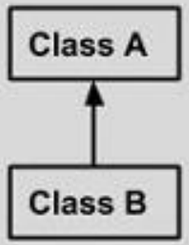
Inheritance

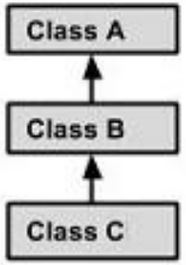
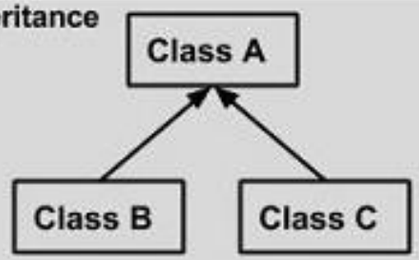
```java
package LAB10;


public abstract class Shape {
    protected double x;
    protected double y;
    protected ShapeColor color = ShapeColor.GRAY;
    private static int screenSize = 10;

    public Shape(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public double getX() {

        return x;
    }

    public void setX(double x) { this.x = x; }

    public double getY() { return y; }

    public void setY(double y) { this.y = y; }
```

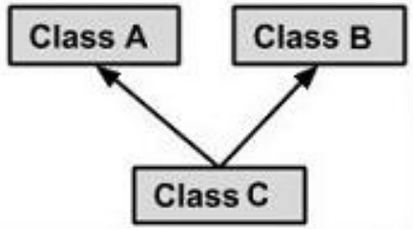
```java
import LAB10.ShapeColor;
import LAB10.StdDraw;
import LAB10.Shape;

public class Circle extends Shape {
    private double radius;
    static final int DEFAULT_RADIUS = 5;


    public Circle(double radius, double x, double y) {
        super(x,y);
        this.radius = radius;

    }

    public Circle(double radius) {
        super( x: 0, y: 0);
        this.radius = radius;

    }

    public Circle(double x, double y) {
        super(x,y);
        this.radius = DEFAULT_RADIUS;
```

The keyword **extends** tells the compiler the **Circle** class extends the **Shape** class, thus inheriting the methods **getX, setX, getY,setY, and toString etc.**

Types of Inheritance

Single Inheritance
 
	public class A {
	.....
	}
	public class B **extends** A {
	.....
	}

Multi Level Inheritance 	public class A { .....} public class B **extends** A {.....} public class C **extends** B {..... }
Hierarchical Inheritance 	public class A { .....} public class B **extends** A {.....} public class C **extends** A {..... }

Multiple Inheritance
 
	public class A { .....}
	public class B {.....}
	public class C **extends** A,B {
	.....
	} // Java does not support mutiple Inheritance

Why multiple inheritance is not supported?

Suppose, Class C inherits Class A and Class B. If there is the same method in both of these classes , it will create an ambiguity whenever we call the standard method using child class object

Inheritance

Super class vs sub class

- If a class **is not defined extends with a super class**, it inherit **java.lang.Object**
- **all the non-private data-filed and methods**（ except constructor ） of super class are inherited by the sub class from its super class.
- **sub class can also define its own data-filed and methods**
- **Superclass can only be one.**

Exercises

Complete the exercises in the **2020S-Java-A-Lab-10.pdf** and submit to the blackboard as required.

贾艳红 Jana
Email:jiayh@mail.sustech.edu.cn