TECHNISCHE
UNIVERSITÄT
WIEN

Welcome.TU.code

OOP: Inheritance

Recap Questions

What are enums? Define an enum for the 4 seasons and a code snippet that prints a different greeting for each season.

Recap Question

Describe the difference between classes and objects.
Which Java keywords are used when defining a class and creating an object?

Recap Question

```
public class Circle {
    private int diameter = 1;
    String color = "Red";
    public int id;
    private static final double PI = 3.1415926;

    private void access() {
        // which variables can we access here?
    }

    private static void access2() {
        // which variables can we access here?
    }
}
```

Recap Question

```
public class Circle {
    private int diameter = 1;
    private int a = 2;

    private void access() {
     int a = 1;
        final int b = 1;
    // which variables can we access here?
        for(int i=0; i<10; i++) {
            double c = 1.0;
            // which variables can we access here?
    }
    // which variables can we access here?
    }
}
```

Recap Question

```
public class Circle {
      private static Circle c3;

    public static void main(String[] args) {
          Circle c1 = new Circle();
          Circle c2;
          c1.access(); // What happens here?
          c2.access(); // What happens here?
          c3.access(); // What happens here?
    }

        private void access() {
            System.out.println("You accessed the
   circle");
     }
}
```

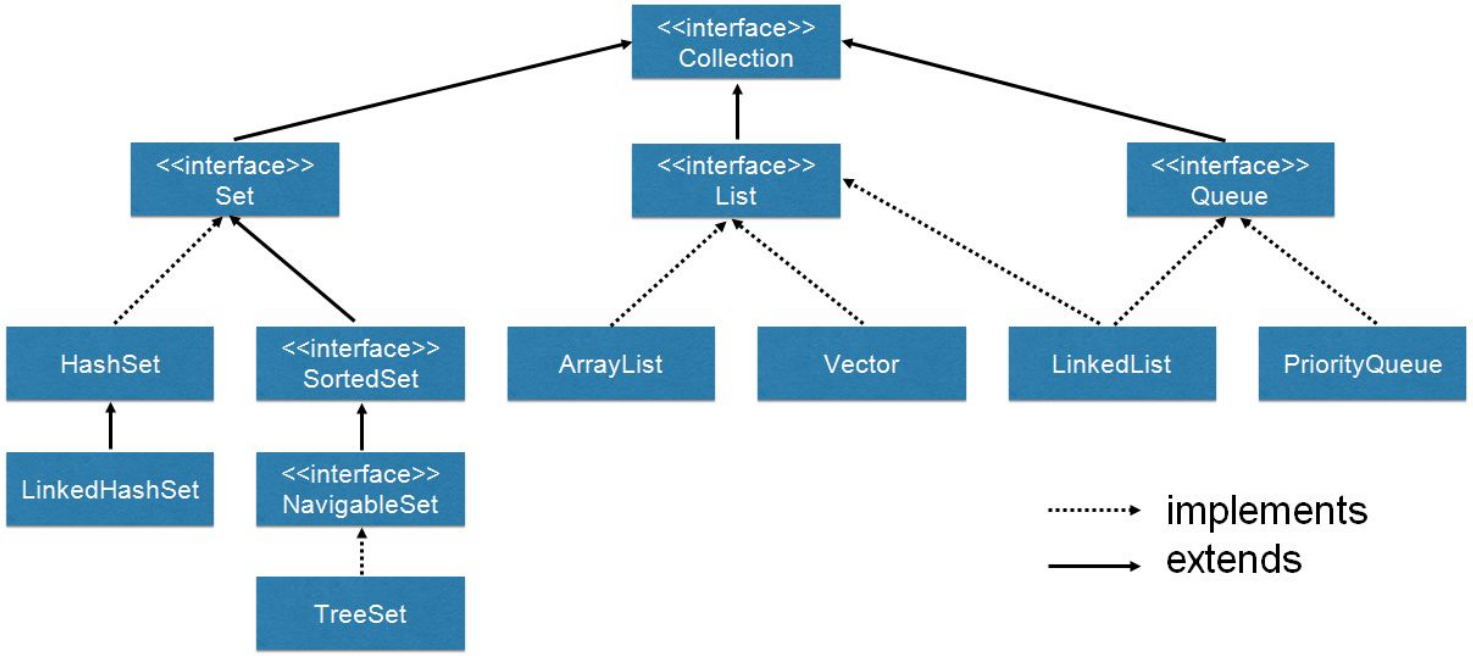
Recap Question

What is the advantage of ArrayLists compared to normal arrays?

Are there any disadvantages?

When would you use an ArrayList?

Our Goal: Collections API



Inheritance

- Reuse code to simplify tasks
- Represent behaviour of real world objects
- Defines a “is-a” relationship
  - e.g. a Car is-a Vehicle, a Human is-a Mammal
- Parent is called superclass, Child is called subclass
- Uses keyword `extends` (or `implements` when using interfaces)
- Java does not support multiple superclasses (but multiple super-interfaces)

Inheritance: Example

```
class Vehicle {
    public void drive() {
        System.out.println(“Vehicle is driving”);
    }
}

class Car extends Vehicle {
}
```

Method call 1: new Vehicle().drive();

Method call 2: new Car().drive();

Inheritance: Example

```
class Vehicle {
    public void drive() {
        System.out.println("Vehicle is driving");
    }
}

class Car extends Vehicle {
    public void drive() {
        System.out.println("Car is driving");
    }
}
```

Method call 1: new Vehicle().drive();

Method call 2: new Car().drive();

Inheritance

- Abstract from details
  - e.g. a caller does not need to know about the internals
  - a caller does not even need to know the “real” type
- Modifiers define what is inherited
  - private - no visibility in subclass
  - default - visible only if subclass is in the same package
  - protected, public - visible in subclass
- java.lang.Object is the top class
  - also implicit superclass if not specified otherwise

Inheritance

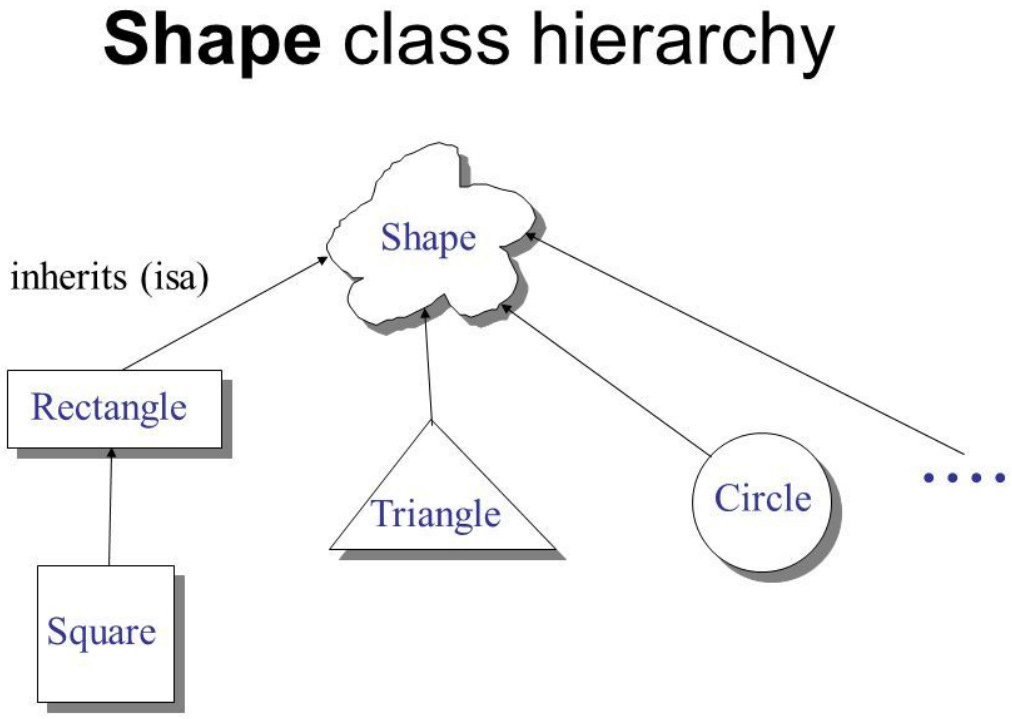
- New keyword `super`
  - Used to reference methods and constructors of the superclass
- Constructor chaining with this() and super()
  - must be the first line in a constructor
- Keyword `this` may become more intuitive
  - this.method() vs super.method()

Inheritance: Example

```
class Vehicle {
    public void drive() {
        System.out.println(“Vehicle is driving”);
    }
}
class Car extends Vehicle {
    public void drive() {
        super.drive();
        System.out.println(“Car is driving”);
    }
}
```

Method call: new Car().drive();

Our class hierarchie



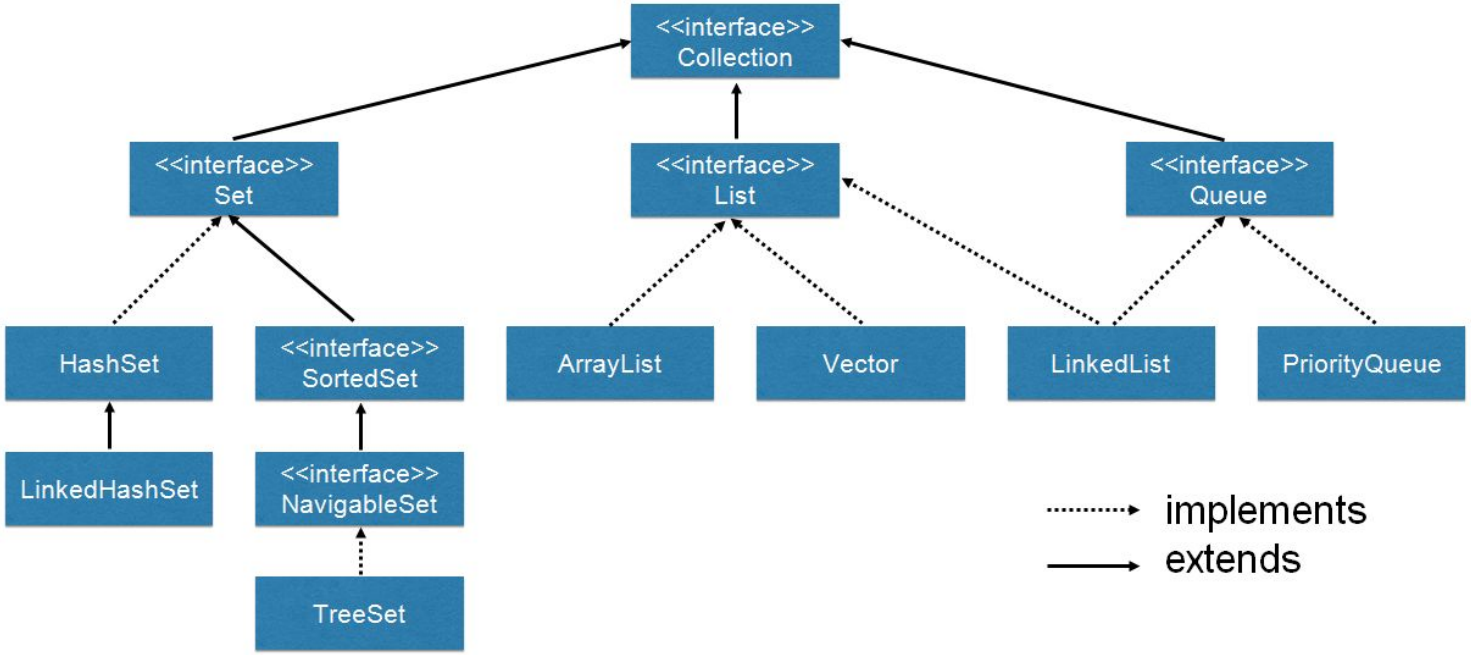
Create class Shape

- Live Example

Inheritance: Keyword abstract

- Can be used for classes and methods
  - public abstract class Abc {}
  - public abstract void method() {}
- abstract methods require an abstract class
- Only non-abstract classes can be instantiated
- Implementing/overriding abstract methods makes them non-abstract
- abstract classes without abstract methods are possible

Revisiting Collections API



Connect 4

Are there any questions about the homework?