Interfaces

```java
public interface Employee {
    (public) double getSalary();
    (public) boolean isManager();
}
```

don not have yields

Sort of abstract method
An interface guarantees methods exist.

public class Flunky implements Employee
If a class implements an interface it must implement every method in that interface.
Interfaces separate polymorphism from inheritance providing a facade to an object.
no reason that full-time/part-time is more fundamental than hour/1/2/4.

Usliness for Jesus' right inheritance.
Why not multiple-inheritance?

- Confusion of methods
  - `circle`
  - `draw`
  - `SunShelter`
  - `draw`

- Confusion of fields
  - Inherits fields
Problems are not insurmountable.

\[
\text{C++}
\]

\[
\text{class A : B, C}
\]

different

\[
\text{class A : C, B}
\]
In Java

Comparable

Serializable

\rightarrow storing data

(persisting)
Composition is another structuring technique.

Diagram:
- Car
- Engine
- ...
From composition
we get the idea of delegation.

Set of data

- average
- standard deviation

Compute order statistics
Other way is interfaces
Cloneable c = new X();
Comparable k = new X();
Employee e = new X();
X myX = new X();

Cloneable c = myX;

Composable K = myX;

Employee e = myX;

K.clone();  c.clone();
class Shape {
    void draw() {
        // Draw code
    }
}

class RoundShooter extends Shape, Gunslinger {
    void draw() {
        // Draw code
    }
}

class Gunslinger {
    void draw() {
        // Draw code
    }
}
interface Shape {
    Color getColor();
    void draw();
}

class RoundShooter implements Shape, Gunslinger

interface Shape {
    Color getColor();
    void draw();
}

class Circle implements Shape {
    void draw() {
        // Implementation
    }
}
```java
class Triangle implements Shape {  
  double draw() { 
    return 3; 
  }  
}  
```
Adapter Pattern

interface MouseListener {
    mouseDown
    mouseUp
    mouseMove
    mouseDrag
}
class mouseAdapter implements MouseListener {

  void mouseDown() {
    ...
    class MouseAdapter extends javax.swing.event.MouseInputAdapter {
      ...
    }
  }
}