public class Employee
    public int getHours()
    public int getVacationDays()
    public String toString()

public class HealthcareWorker extends Employee
    public String getHospital()
    public String toString()
    public int getHours()

public class Doctor extends HealthcareWorker
    public void takePulse()
    public void takePulse(String patient)
    public String toString()

public class Surgeon extends Doctor
    public void performSurgery()
    public String toString()

public class PhysicalTherapist extends HealthcareWorker
    public String toString()

public class Astronaut extends Employee
    public void takeoff()
    public String toString()
    public int getHours()

public class Lawyer extends Employee
    public void argue()
    public String toString()
    public int getHours()
### Class vs Abstract Class vs Interface

<table>
<thead>
<tr>
<th></th>
<th>Class</th>
<th>Abstract Class</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can be instantiated?</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Keyword for inheritance</td>
<td>extends</td>
<td>extends</td>
<td>implements</td>
</tr>
<tr>
<td>Methods predefined?</td>
<td>Yes</td>
<td>Some</td>
<td>No*</td>
</tr>
<tr>
<td>Number able to inherit</td>
<td>0 or 1</td>
<td>0 or 1</td>
<td>As Many as you want!</td>
</tr>
</tbody>
</table>

- **Class**: Use when it “makes sense” as a standalone object. Use inheritance when you have a special case or variant of superclass.

- **Interface**: Use when you know what actions are needed but not how to do them. Implementation is deferred to a class. A “contract” of methods guaranteed to exist.

- **Abstract Class**: Use when superclass is “incomplete” or otherwise does not “make sense” as a standalone. For some methods, we know we need them, but don’t know how to implement them. Other methods can be implemented. Subclass is responsible for completing implementations.