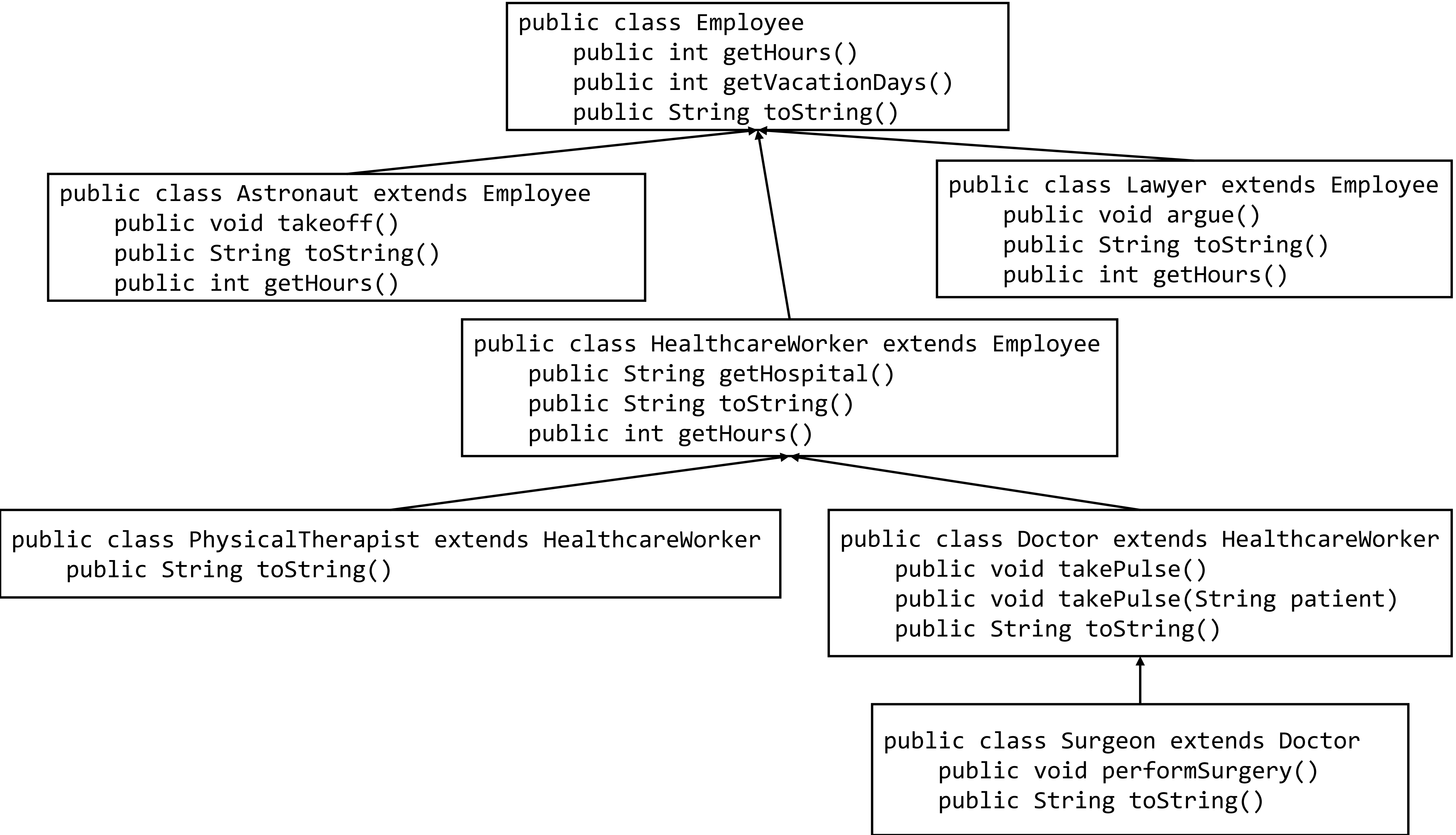

```
public class Employee
    public int getHours()
    public int getVacationDays()
    public String toString()
```

```
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()
```

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

```
public class PhysicalTherapist extends HealthcareWorker
    public String toString()
```

```
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()
```

Class vs Abstract Class vs Interface

	Class	Abstract Class	Interface
Can be instantiated?	Yes	No	No
Keyword for inheritance	extends	extends	implements
Methods predefined?	Yes	Some	No*
Number able to inherit	0 or 1	0 or 1	As Many as you want!

- **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