abstract(抽象)

abstract：修饰方法

抽象方法：

格式：public abstract 返回值类型 方法名([可选参数列表]);

抽象方法不可以有方法体空方法体也不行；

abstract:修饰类

抽象类：

格式：abstract class 类名{}

注意：1.；如果类中含有抽象方法，则要求这个类必须是抽象类；但是类中可以不包含抽象方法

1. 抽象类的子类：
2. 抽象类：如果不想重写抽象父类的方法可以将该类也声明为抽象类
3. 具体的类：要求必须重写抽象类中的所有的抽象方法
4. 如果类被声明为抽象类，不允许通过new来实例化
5. 类可以包含那些内容？

成员属性：变量可以存在 常量可以存在

成员方法：抽象方法可以 非抽象方法可以

构造方法：可以有，也支持构造方法重载；提供给子类使用

1. 抽象类中为什么既提供了抽象方法也提供非抽象方法？

抽象方法：强制要求具体子类必须重写抽象方法

非抽象方法：可以直接使用父类的非抽象方法--提高代码复用性

1. 抽象类是否可以体现多态？

抽象方法可以重载；抽象方法的意义就是子类重写该方法；也支持向上造型，把子类对象赋值给抽象类的引用，编译看左边，运行看右边。

1. 如果抽象类中一个抽象方法都没有，这个抽象类有什么作用？

不允许创建类的对象，只能通过子类继承该类

1. private+abstract可不可以一起使用修饰方法？

不可以，因为private定义的方法只能本类访问，抽象方法必须要重写

1. final+abstract可不可以一起使用修饰方法？

不可以，因为final定义的方法不能被重写，抽象方法必须要重写

1. static+abstract可不可以一起使用？

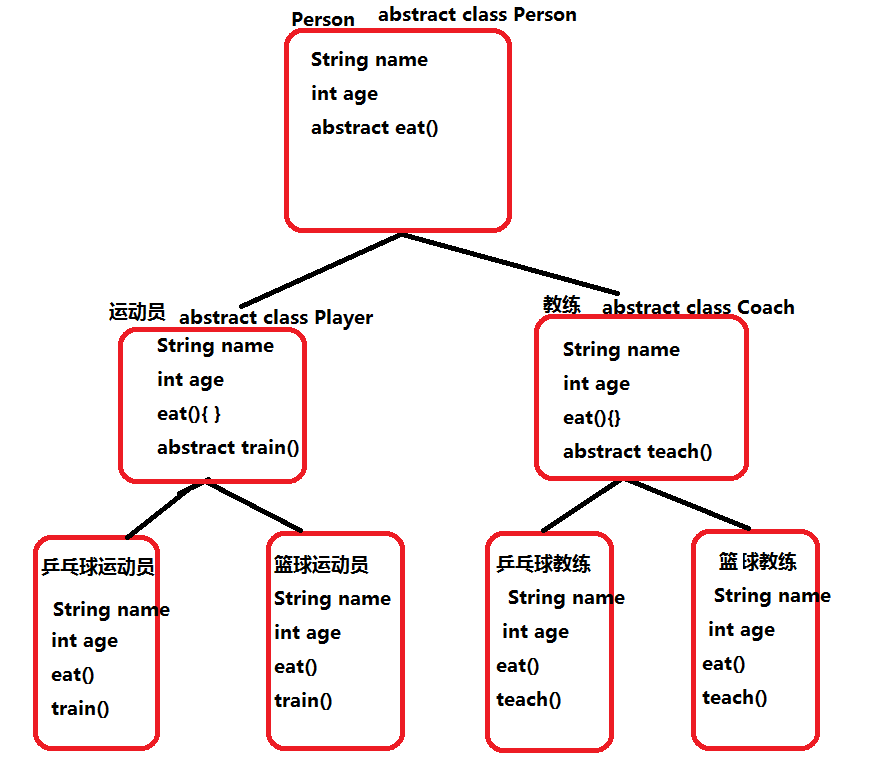
静态方法属于类，不可以被重写，抽象方法必须被重写；静态方法属于类能被类名点调用然后抽象方法没有方法体所以没有意义。

1. this、super关键字可不可以在抽象类中存在？

可以，this指代的是当前对象，用this调用抽象方法时要看实例化的对象，然后找实例化对象中重写的父类抽象方法。

案例：运动员和教练

示例图：



从具体到抽象，从抽象到具体

