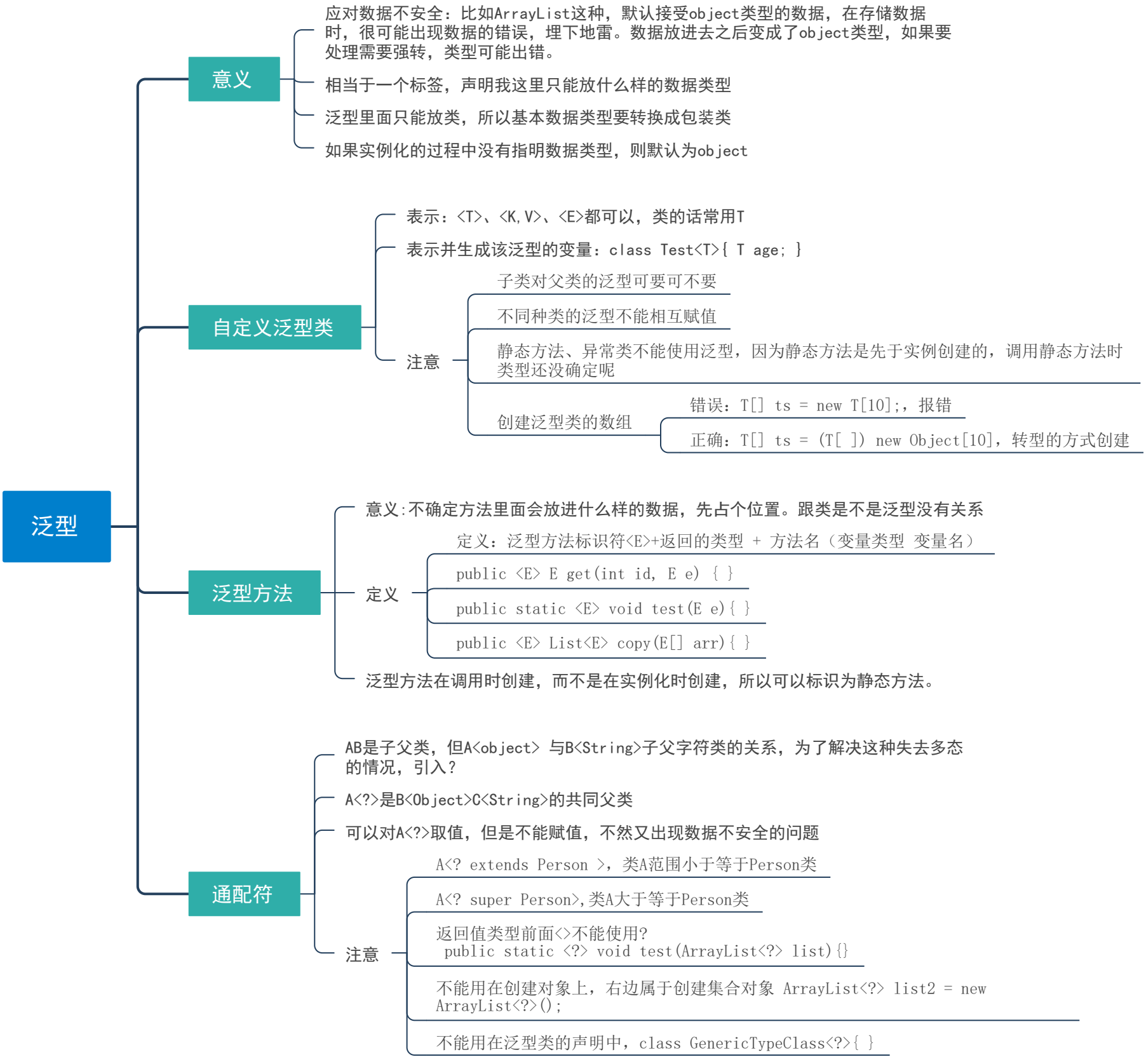
泛型
意义
应对数据不安全：比如ArrayList这种，默认接受object类型的数据，在存储数据时，很可能出现数据的错误，埋下地雷。数据放进去之后变成了object类型，如果要处理需要强转，类型可能出错。
相当于一个标签，声明我这里只能放什么样的数据类型
泛型里面只能放类，所以基本数据类型要转换成包装类
如果实例化的过程中没有指明数据类型，则默认为object
自定义泛型类
表示：<T>、<K,V>、<E>都可以，类的话常用T
表示并生成该泛型的变量：class Test<T>{ T age; }
注意
子类对父类的泛型可要可不要
不同种类的泛型不能相互赋值
静态方法、异常类不能使用泛型，因为静态方法是先于实例创建的，调用静态方法时类型还没确定呢
创建泛型类的数组
错误：T[] ts = new T[10];，报错
正确：T[] ts = (T[ ]) new Object[10]，转型的方式创建
泛型方法
意义:不确定方法里面会放进什么样的数据，先占个位置。跟类是不是泛型没有关系
定义
定义：泛型方法标识符<E>+返回的类型 + 方法名（变量类型 变量名）
public <E> E get(int id, E e) { }
public static <E> void test(E e){ }
public <E> List<E> copy(E[] arr){ }
泛型方法在调用时创建，而不是在实例化时创建，所以可以标识为静态方法。
通配符
AB是子父类，但A<object> 与B<String>子父字符类的关系，为了解决这种失去多态的情况，引入?
A<?>是B<Object>C<String>的共同父类
可以对A<?>取值，但是不能赋值，不然又出现数据不安全的问题
注意
A<? extends Person >，类A范围小于等于Person类
A<? super Person>,类A大于等于Person类
返回值类型前面<>不能使用?
public static <?> void test(ArrayList<?> list){}
不能用在创建对象上，右边属于创建集合对象 ArrayList<?> list2 = new ArrayList<?>();
不能用在泛型类的声明中，class GenericTypeClass<?>{ }