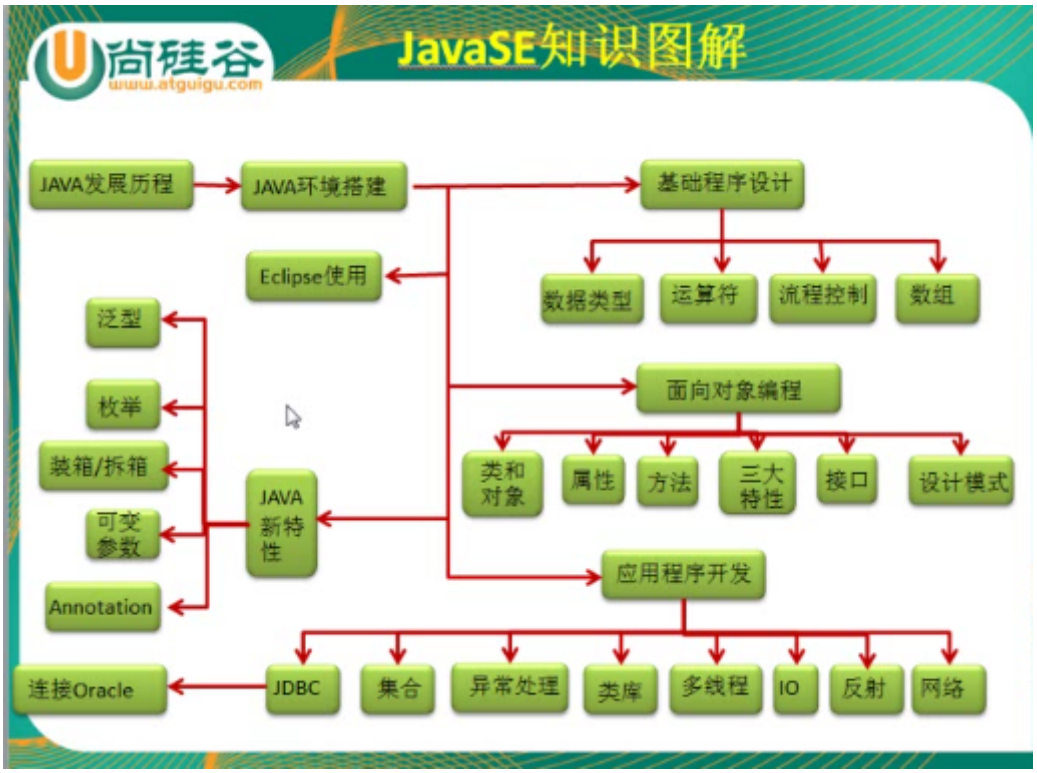
一，再集合中使用泛型（掌握）

二，自定义泛型、泛型接口、泛型方法（理解----->使用）

三，泛型与继承的关系

四，通配符



一，泛型在集合中的使用

1，在没有使用泛型的时候，任何object及其子类的对象都可以添加进来

2，强转数据类型时，可能会报classCastException的异常

使用泛型的必要性：

①解决元素存储的安全性问题

②解决获取元素时，需要类型转换的问题

DAO:database access object（数据库访问对象）

二，自定义泛型类的使用

1，当实例化泛型类对象时，指明泛型的类型，指明以后，对应的类中所有的类使用泛型的位置，都

变为实例化中指定的泛型类型

2，如果我们自定义了泛型类，但是在实例化时没有使用，那么默认类型时object类型的

```java
//继承泛型类或泛型接口时，可以指明泛型的类型
class SubOrder extends Order<Integer>{

}
```

```java
//声明泛型方法
public <E> E getE(E e){
    return e;
}
```

```java
//实现数组到集合的复制
public <E> List<E> fromArrayToList(E[] e,List<E> list){
    for(E e1 : e){
        list.add(e1);
    }
    return list;
}
```

三，泛型与继承的关系

若类A是类B的子类，那么List<A>就不是List<B>的子接口

四，通配符

```java
* 通配符？
* List<A>、List<B>、。。。。都是List<?>的子类
*
* ? extends A :可以存放A及其子类
* ? super A:可以存放A及其父类
```

```java
List<? extends Number> list3 = null;
List<Integer> list4 = null;
list3 = list4;
list3 = list1;
List<? super Number> list5 = null;
list5 = list1;
```

对于泛型类（包含集合类）：

①静态方法中不能使用类的泛型

②如果泛型类是一个接口或者抽象类，则不可以创建泛型类的对象

③不能在catch中使用泛型

④从泛型类派生子类，泛型类型需要具体化

⑤可以读取声明为通配符的集合类的对象

⑥不允许向声明为通配符的集合类中写入对象

泛型与继承的关系：