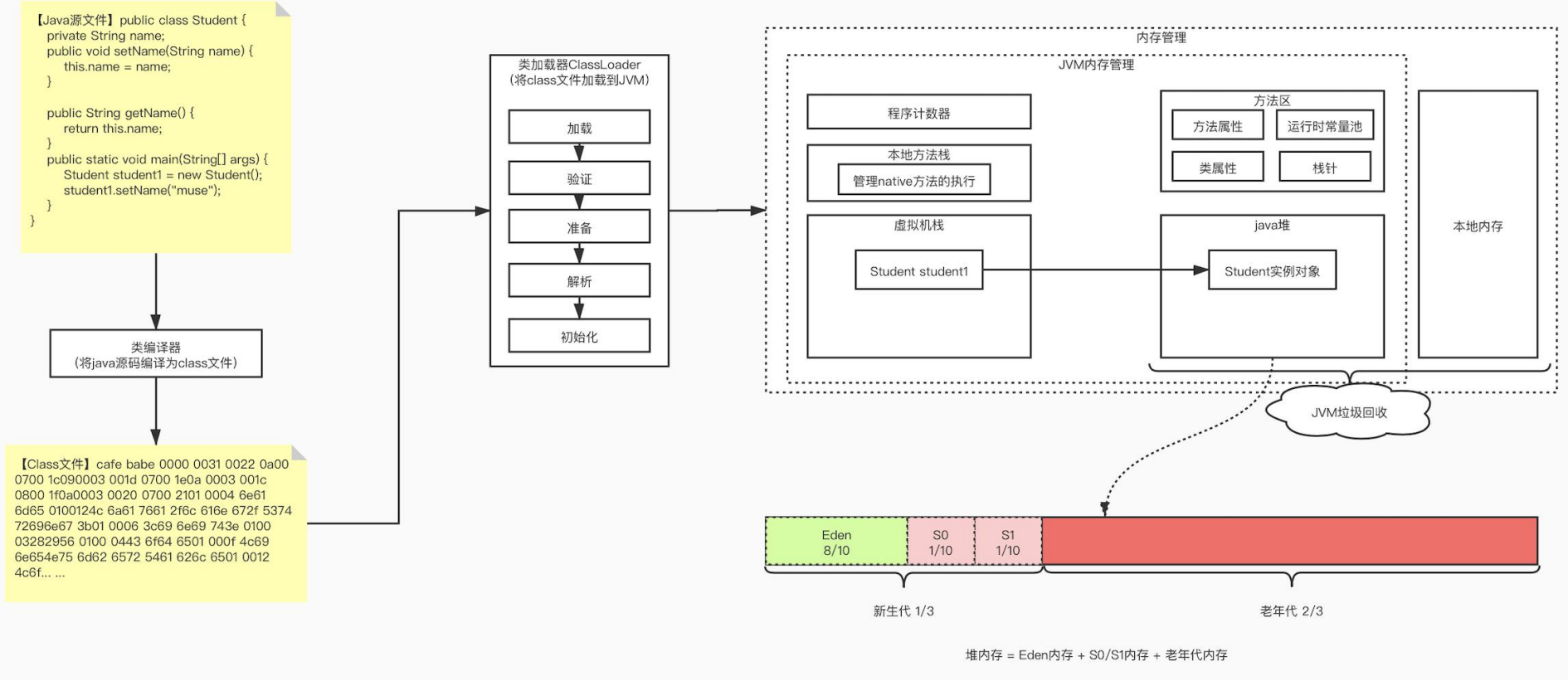
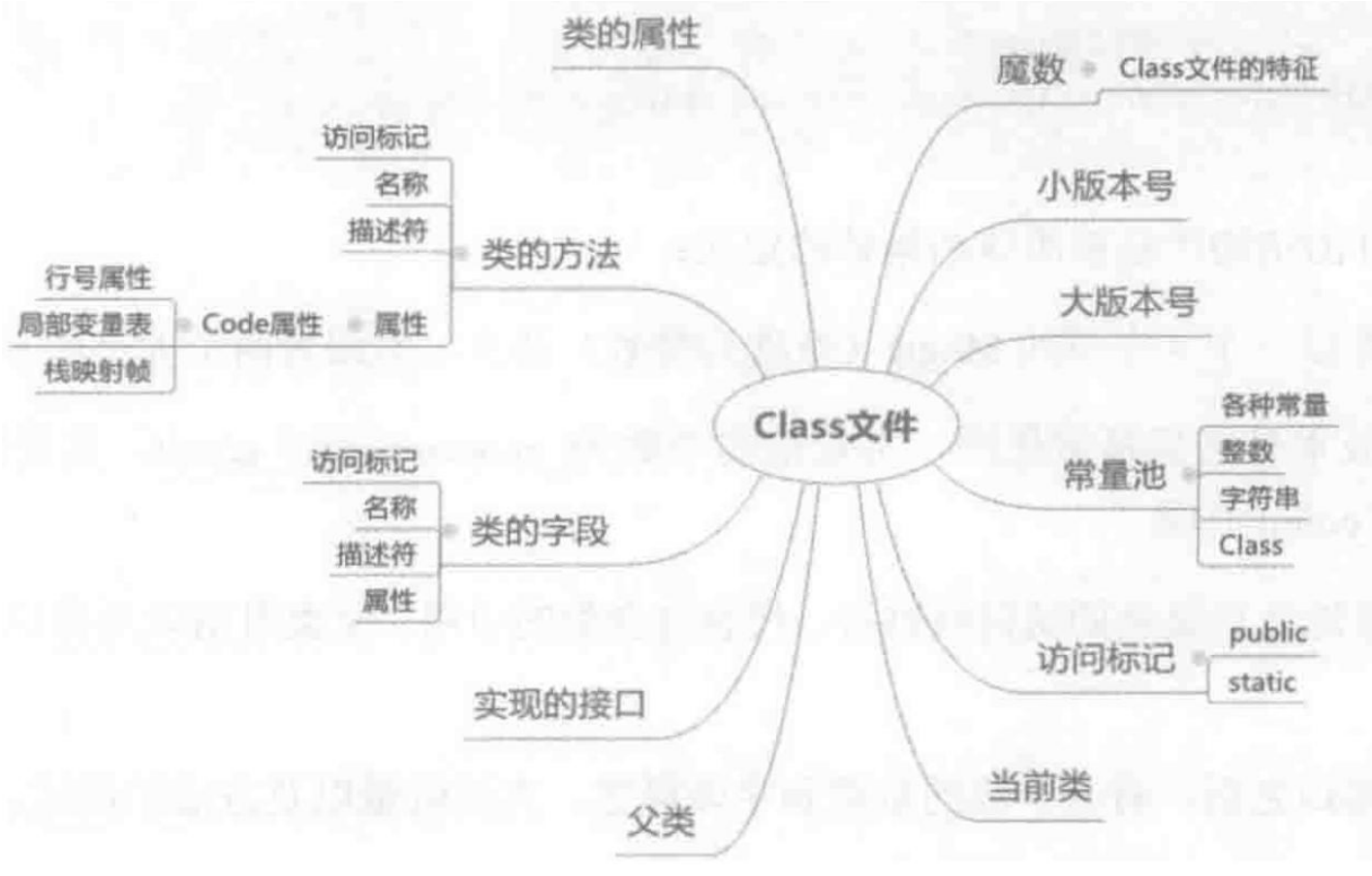
反射、泛型、容器

反射

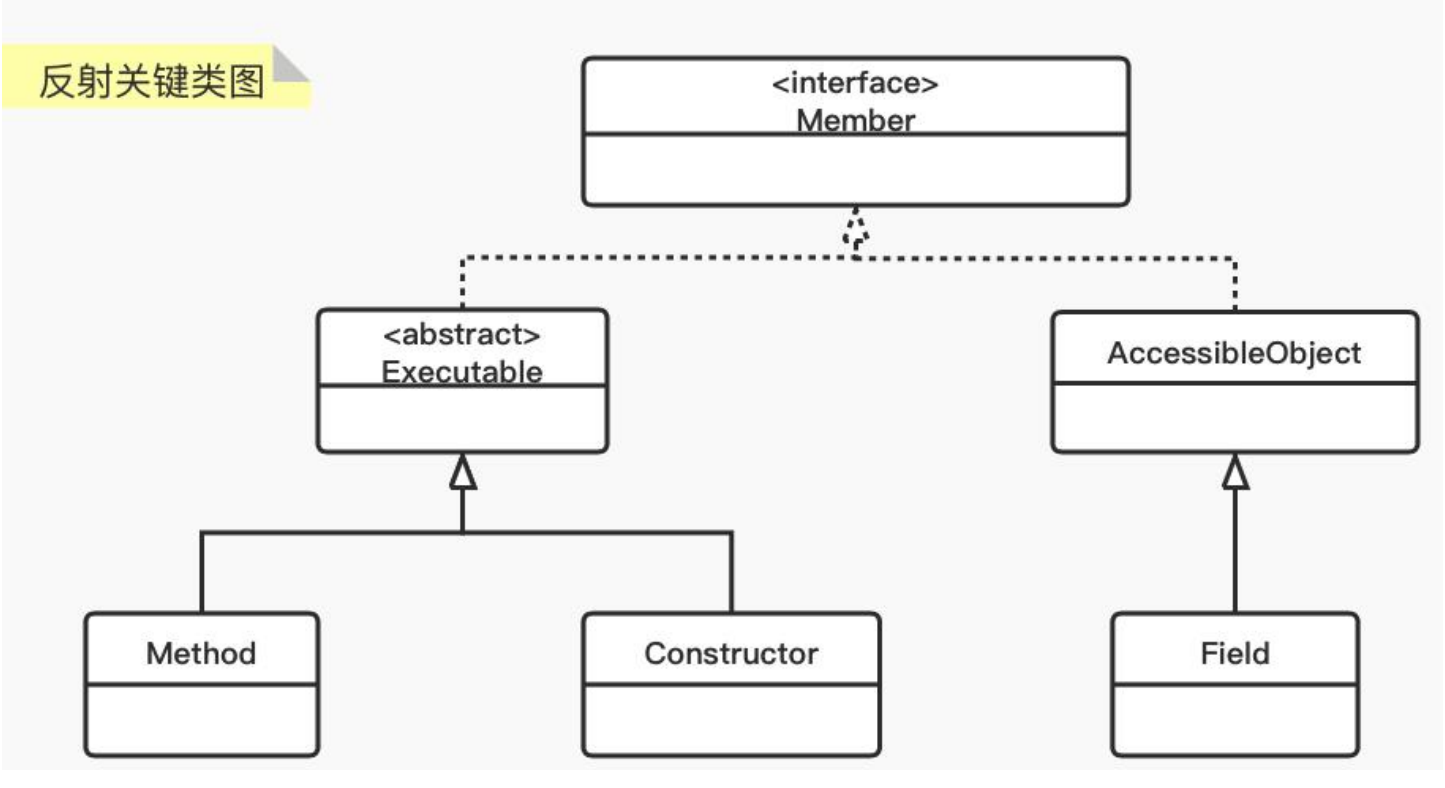
JVM类加载流程和内存结构



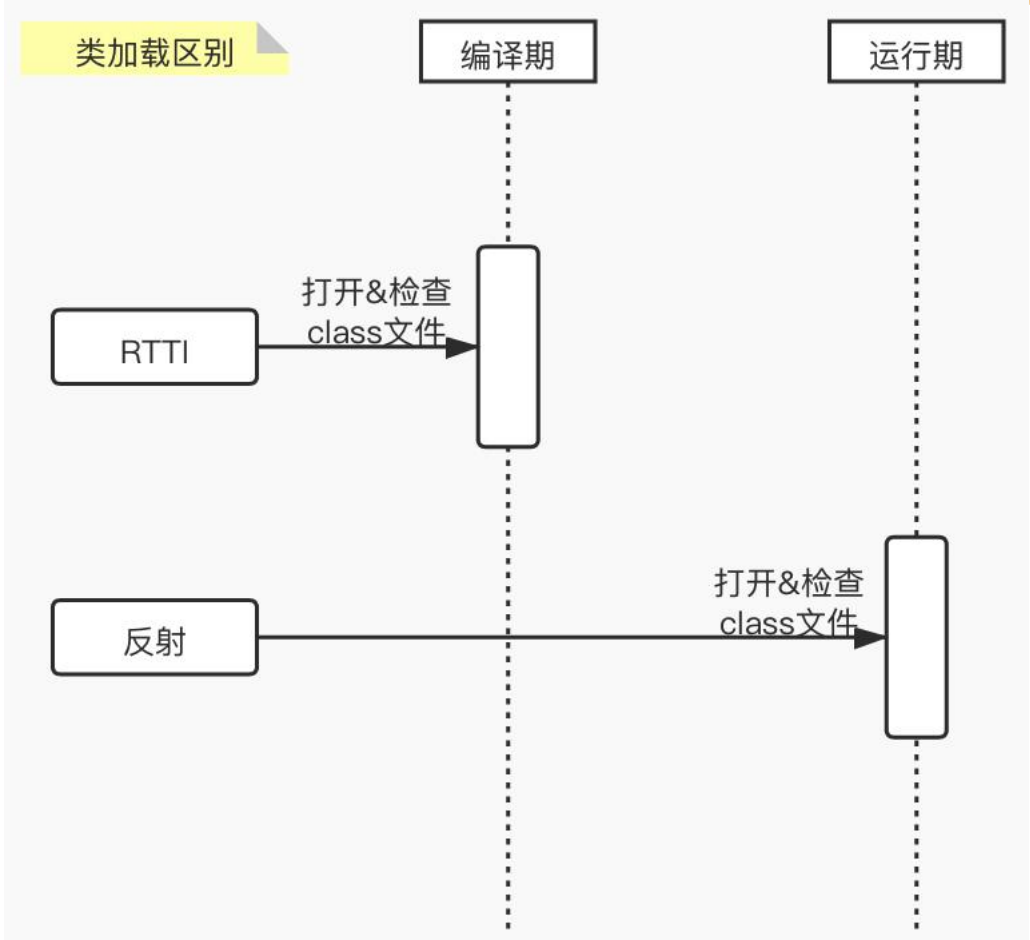
Class文件包含的内容



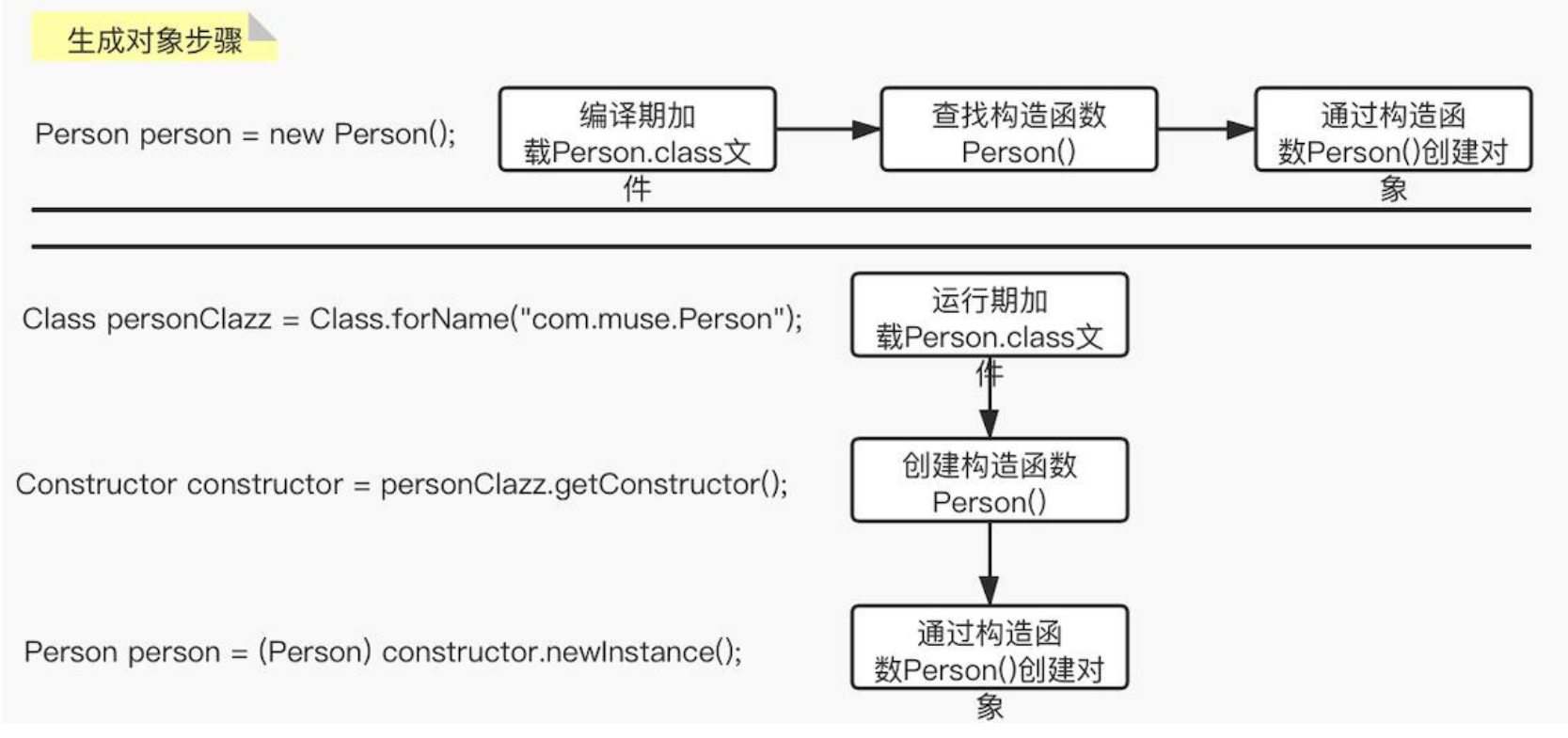
反射关键类图


反射关键类图
<interface>
Member
<abstract>
Executable
AccessibleObject
Method
Constructor
Field

类加载区别



生成对象的步骤


生成对象步骤
Person person = new Person();
编译期加载Person.class文件
查找构造函数Person()
通过构造函数Person()创建对象
Class personClazz = Class.forName("com.muse.Person");
运行期加载Person.class文件
Constructor constructor = personClazz.getConstructor();
创建构造函数Person()
Person person = (Person) constructor.newInstance();
通过构造函数Person()创建对象

我们采用反射机制来实现一个工具BeanUtils，
可以将一个对象属性相同的值赋值给另一个对象。

泛型

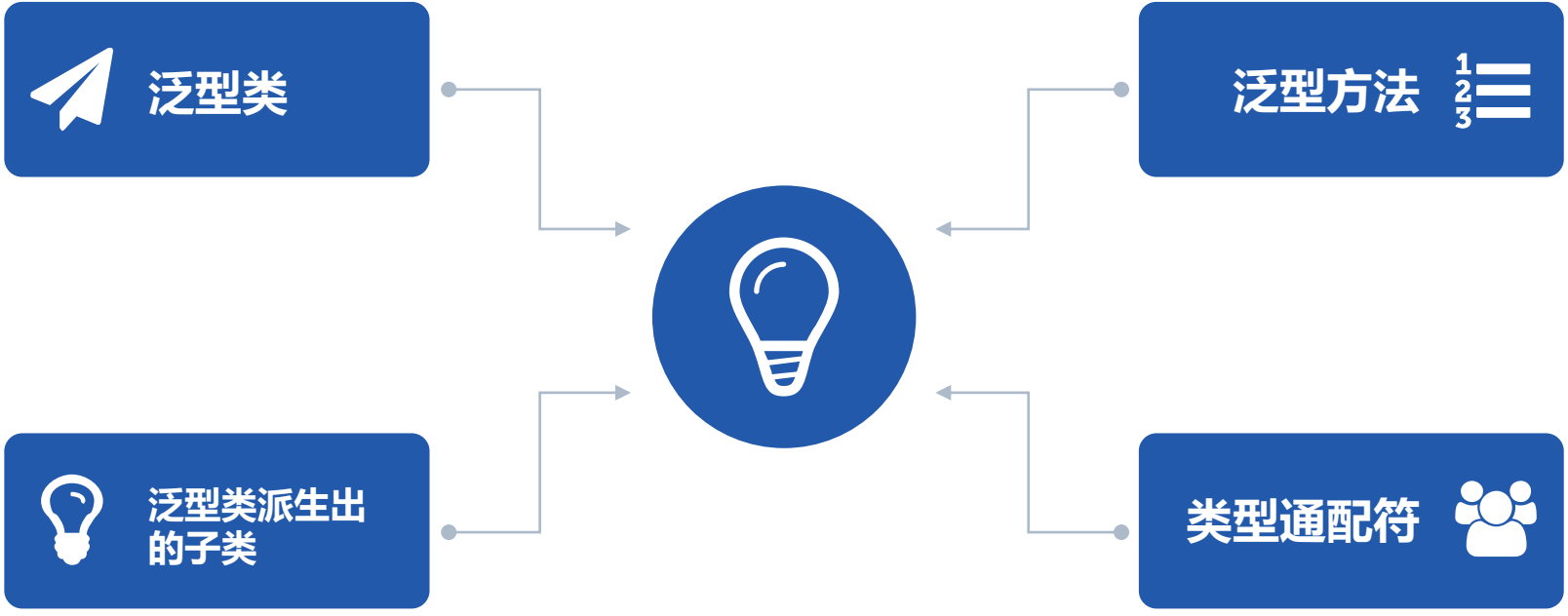
什么是泛型

泛型本质是指类型参数化。

允许在定义类、接口、方法时使用类型形参，当使用时指定具体类型。

所有使用该泛型参数的地方都被统一化，保证类型一致。如果未指定具体类型，默认是Object类型。集合体系中的所有类都增加了泛型，泛型也主要用在集合。

泛型的基础



泛型的上限和下限

泛型的上限：

- **格式：** 类型名称 <? extends 类 > 对象名称
- **意义：** 只能接收该类型及其子类

泛型的下限：

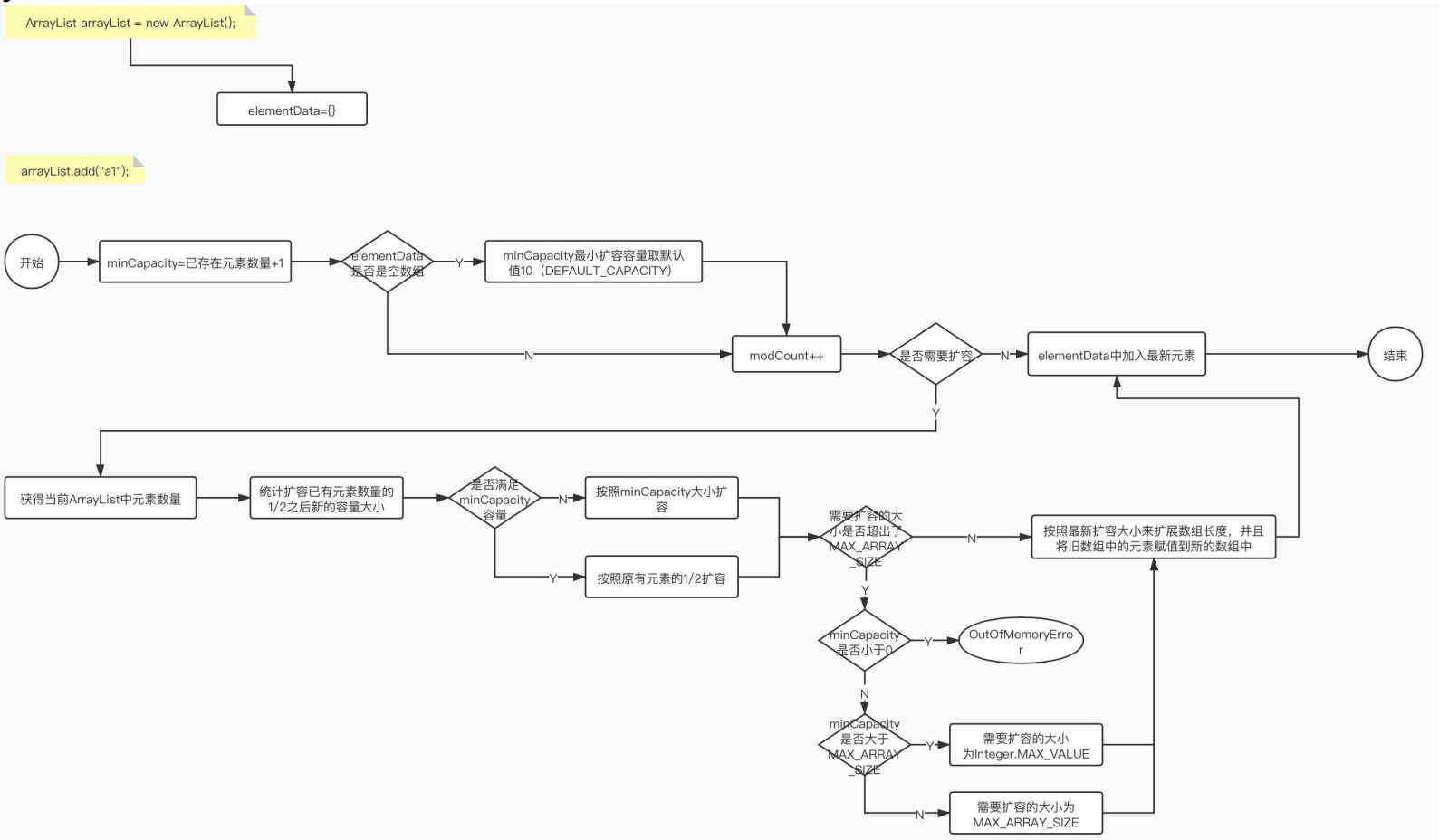
- **格式：** 类型名称 <? super 类 > 对象名称
- **意义：** 只能接收该类型及其父类型

泛型是提供给javac编译器使用的，它用于限定集合的输入类型，让编译器在源代码级别上，即挡住向集合中插入非法数据。但编译器编译完带有泛形的java程序后，生成的class文件中将不再带有泛型信息，以此使程序运行效率不受到影响，这个过程称之为“擦除”。

由于类型被擦除了，为了维持多态性，所以编译器就自动生成了桥接方法。

集合容器

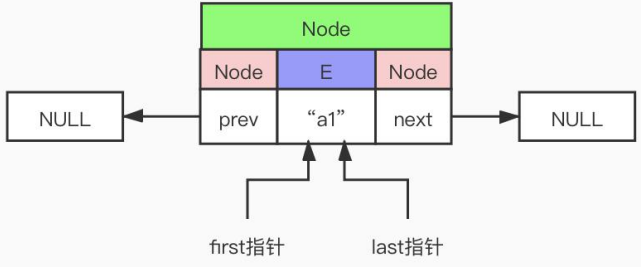
ArrayList源码解析



LinkedList源码解析

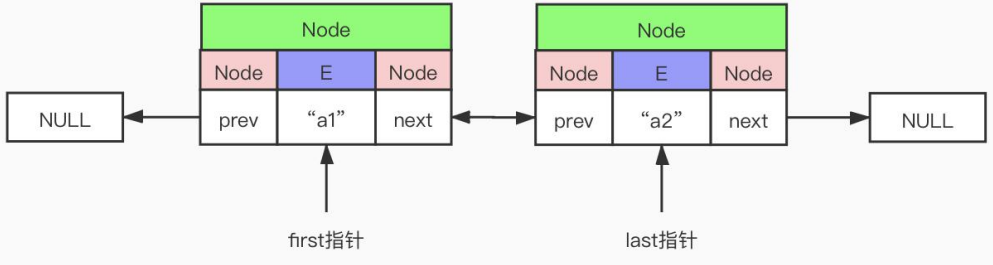
```
LinkedList linkedList = new LinkedList();
```

```
linkedList.add("a1");
```

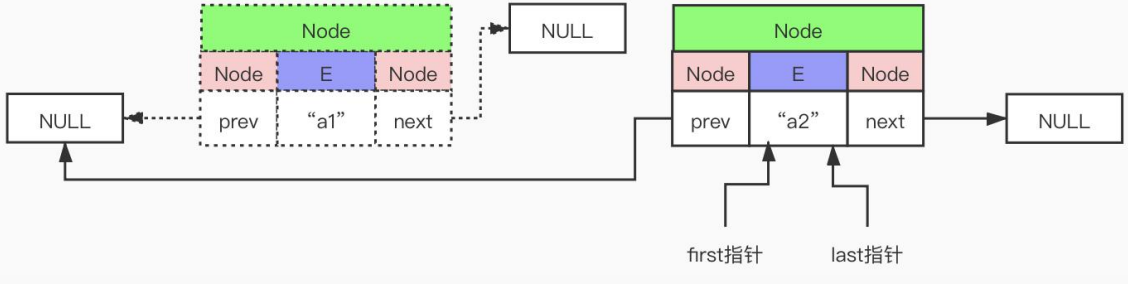


first指针

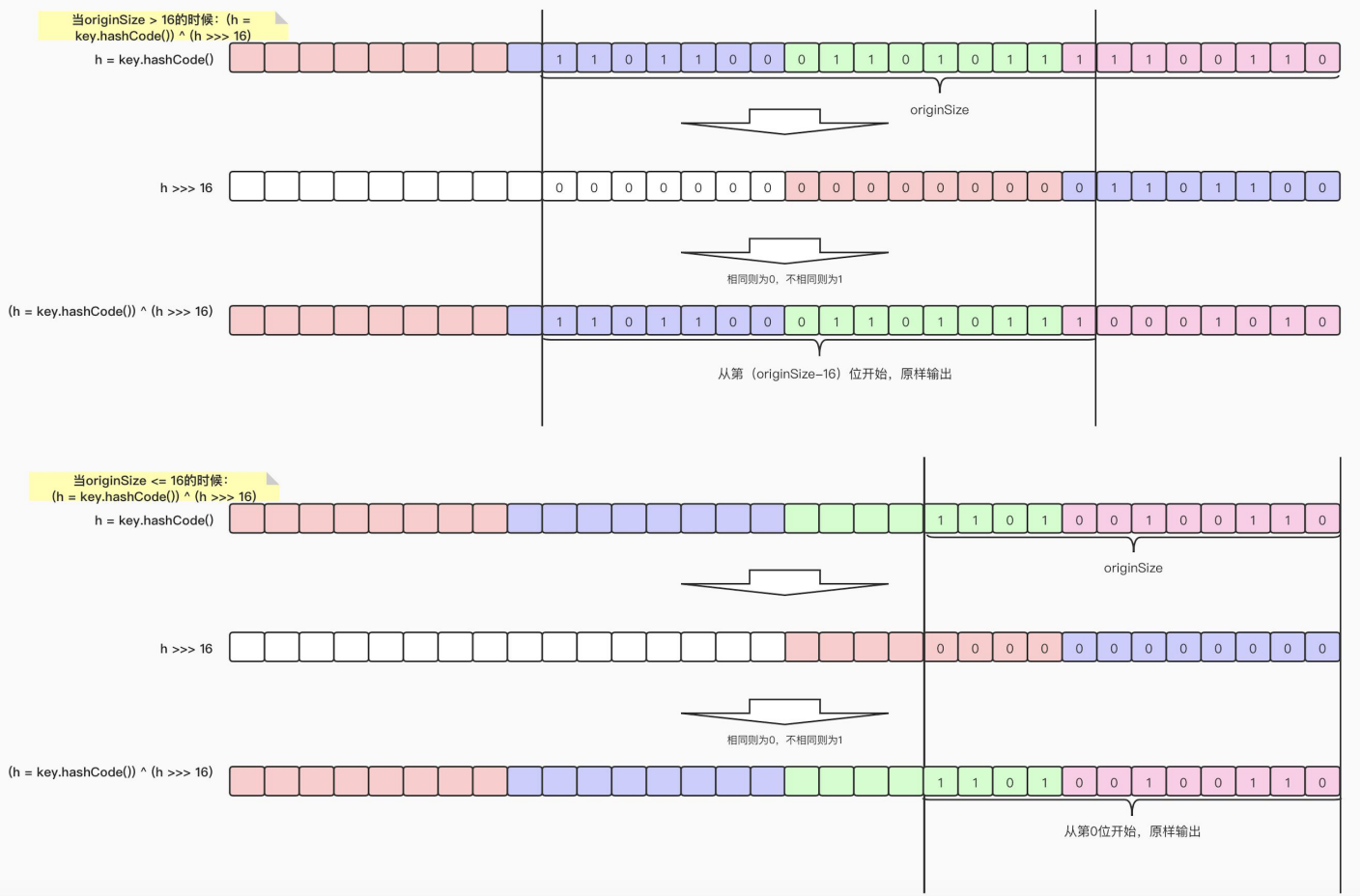
```
linkedList.add("a2");
```



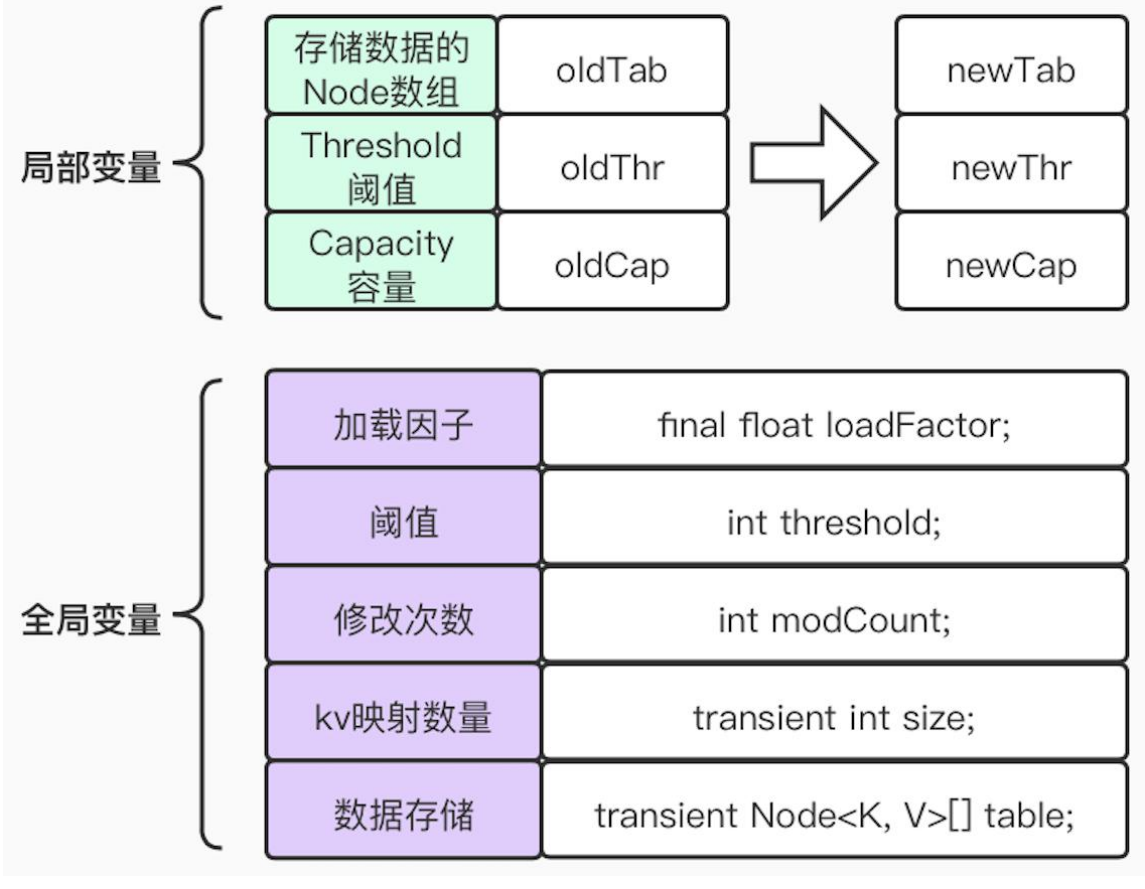

```
linkedList.remove(1);
```



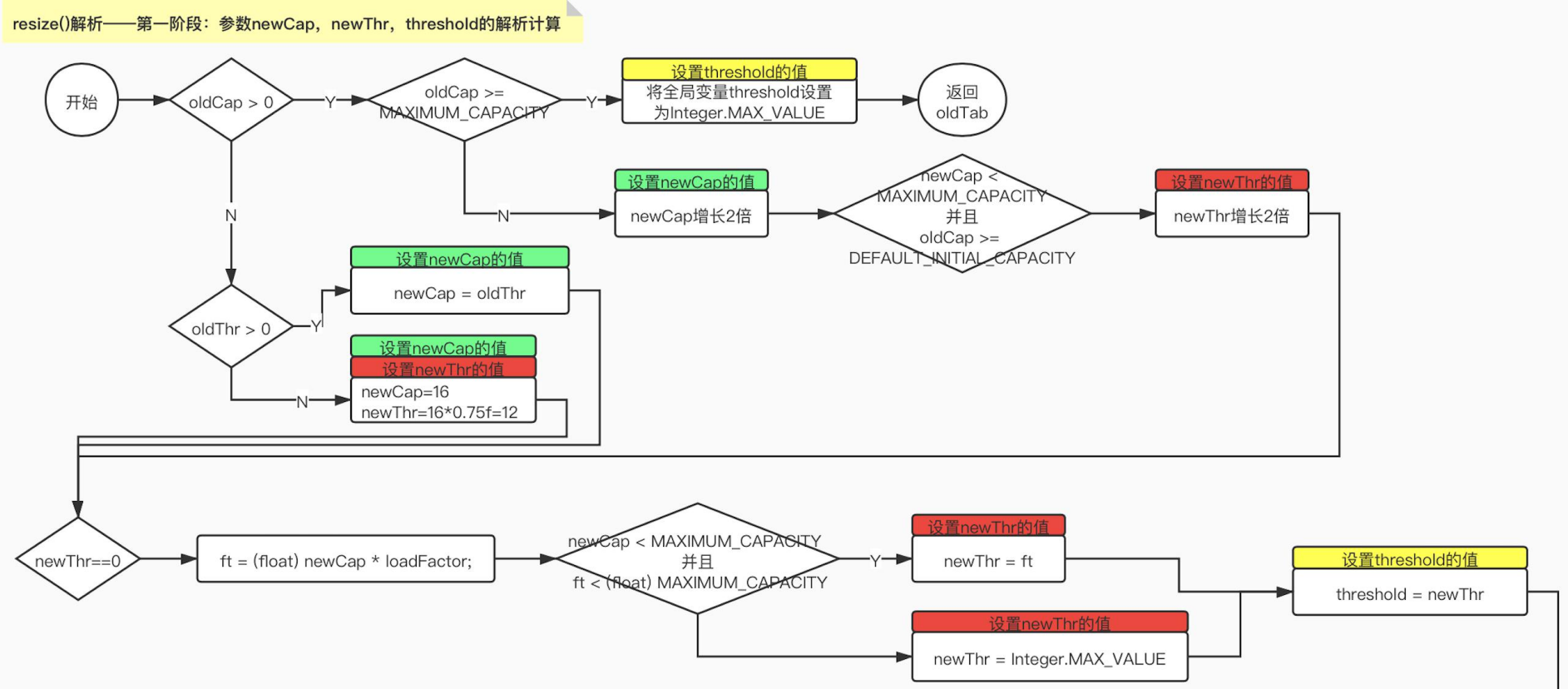
HashMap源码解析——HashMap.hash(Object key)



HashMap源码解析——关键变量



HashMap源码解析——resize



HashMap源码解析——resize