Type Checking

static checking：编译时检查

- **Syntax errors** 语法错误, like extra punctuation or spurious words. Even dynamically-typed languages like Python do this kind of static checking.
- **Wrong names** 类名/函数名错误, like `Math.sine(2)` . (The right name is `sin`)
- **Wrong number of arguments** 参数数目错误, like `Math.sin(30, 20)` .
- **Wrong argument types** 参数类型错误, like `Math.sin("30")` .
- **Wrong return types** 返回值类型错误, like `return "30";` from a function that's declared to return an `int` .

dynamic checking：运行时检查

- **Illegal argument values** 非法的参数值. For example, the integer expression `x/y` is only erroneous when `y` is actually `zero`; otherwise it works. So in this expression, `divide-by-zero` is not a static error, but a dynamic error.
- **Unrepresentable return values** 非法的返回值, i.e., when the specific return value can't be represented in the type.
- **Out-of-range indexes** 越界, e.g., using a negative or too-large index on a string.
- **Calling a method on a `null` object reference.** 空指针

总结：

1. 静态检查：关于"类型"的检查，不考虑值
2. 动态检查：关于"值"的检查

Data Type

immutable

不变对象：一旦被创建，始终指向同一个值/引用

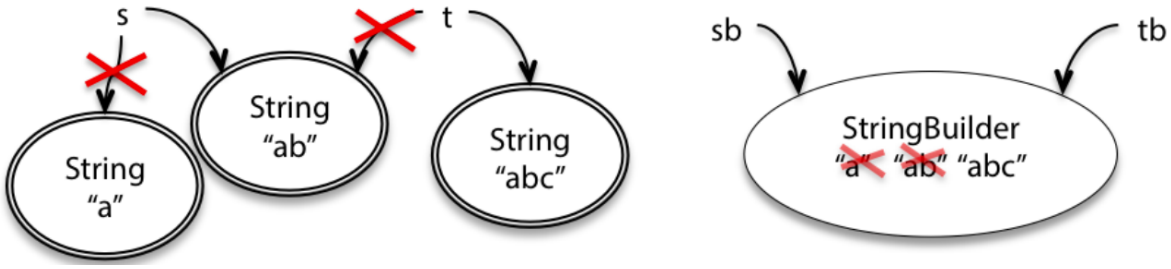
mutable

可变对象：拥有方法可以修改自己的值/引用

例如：

```java
String str = "a";
str = str + "b";
String otherStr = str;
otherStr += "c";
StringBuilder sb = new StringBuilder("a");
sb.append("b");
StringBuilder otherSb = sb;
otherSb.append("c");
```

snapshot diagram

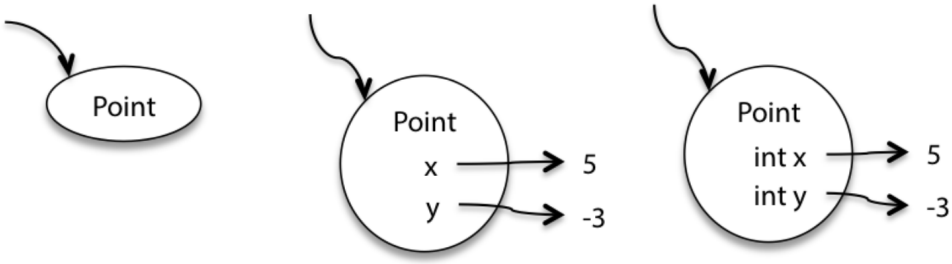


尽量少使用可变数据类型

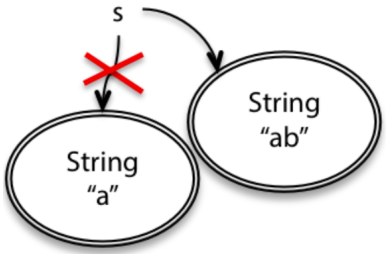
Snapshot Diagram

基本类型：

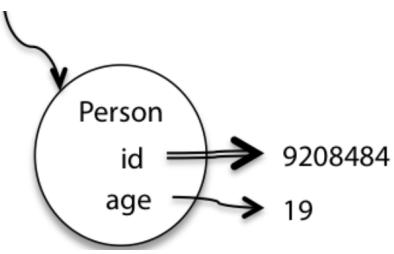
对象类型：



不可变数据对象用双椭圆：



不可变的引用用双箭头：



迭代器错误

原因：iterator是一个mutable的类，它里面有mutator

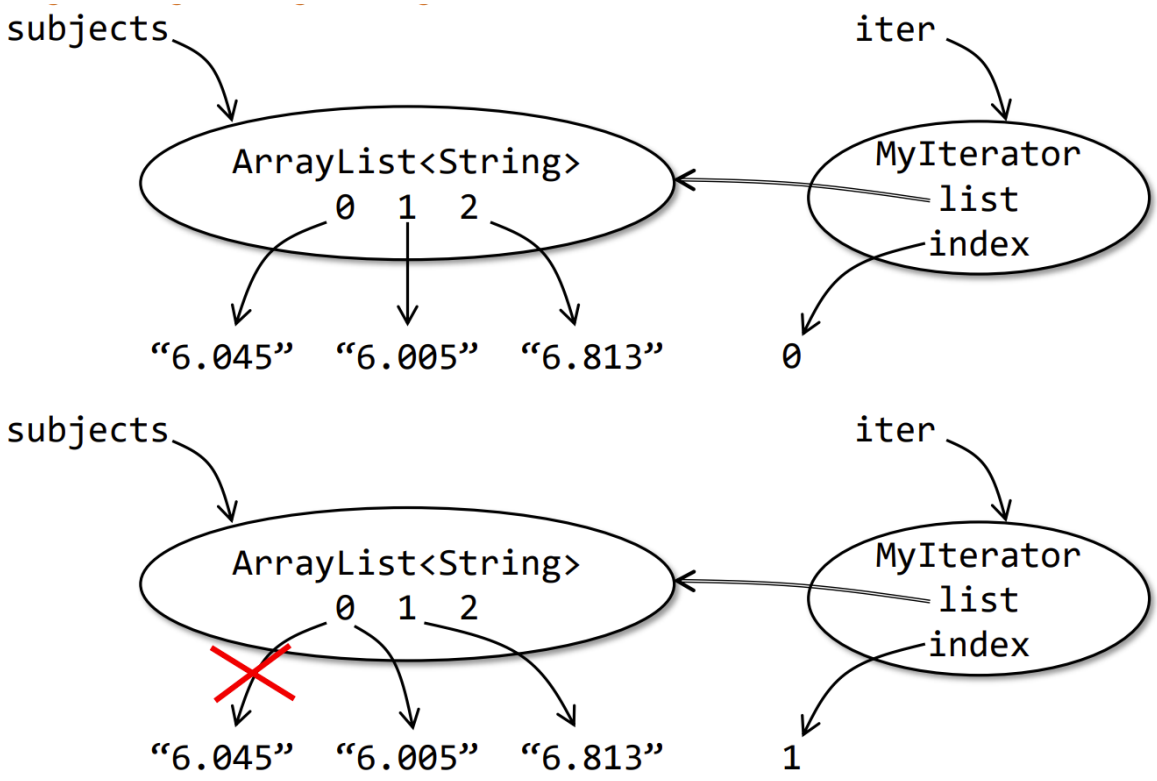
- **The implementation:**

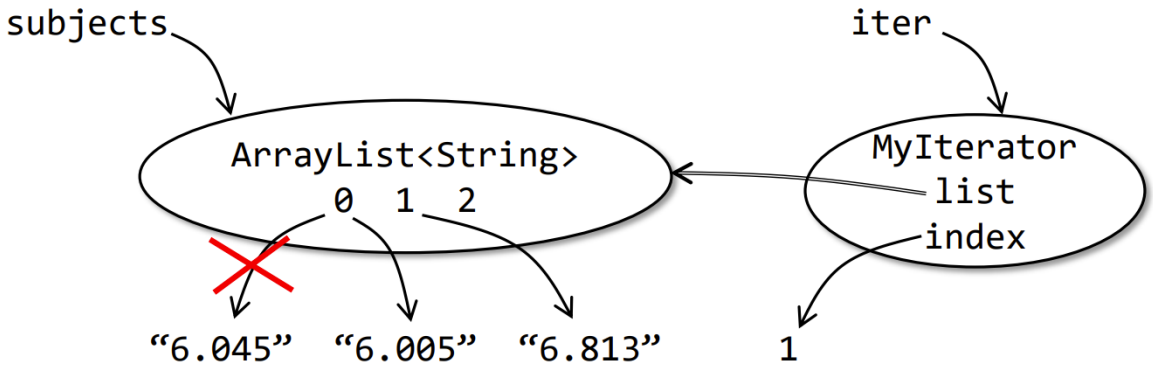
```
public static void dropCourse6(ArrayList<String> subjects) {
    MyIterator iter = new MyIterator(subjects);
    while (iter.hasNext()) {
        String subject = iter.next();
        if (subject.startsWith("6.")) {
            subjects.remove(subject);
        }
    }
}
```

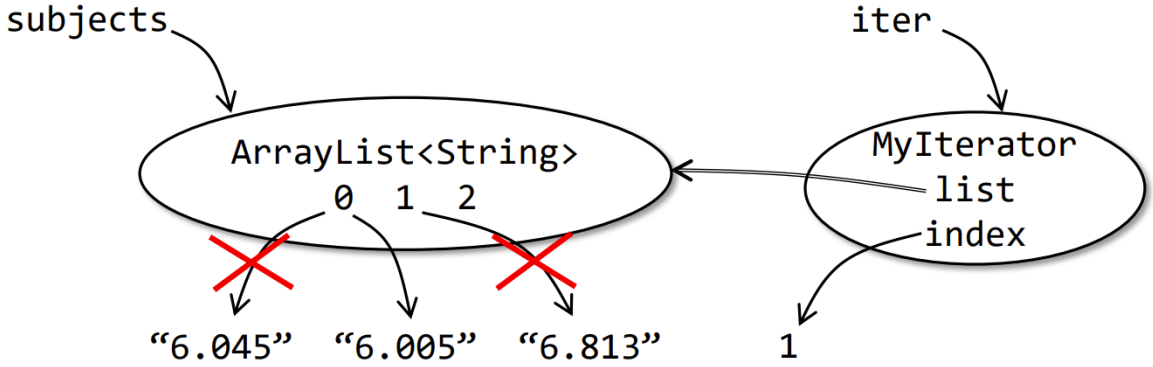
- **Run it**

```
// dropCourse6(["6.045", "6.005", "6.813"])
//   expected [], actual ["6.005"]
```

Why?
Draw the snapshot
and analysis…







- **How about this code?**

```
for (String subject : subjects) {
    if (subject.startsWith("6.")) {
        subjects.remove(subject);
    }
}
```

```
Exception in thread "main" java.util.ConcurrentModificationException
        at java.util.ArrayList$Itr.checkForComodification(Unknown Source)
        at java.util.ArrayList$Itr.next(Unknown Source)
        at Immutable.main(Immutable.java:23)
```

- **Try this:**

```
Iterator iter = subjects.iterator();
while (iter.hasNext()) {
    String subject = iter.next();
    if (subject.startsWith("6.")) {
        iter.remove();
    }
}
```

The iterator adjusts its index appropriately.

ArrayList就是一个动态数组