Java泛型通配符 ? 快速手册

前言

故事的开始在于完成 `MIT6.830 lab3 exercise3` 的时候遇到了如下的代码：

```
Class<?> c = Class.forName("simpledb.execution.HashEquiJoin");
```

开始不明所以起来了，发现触及了自己的知识盲区，了解之后发现还十分有用，刚好可以让 `Exercise 2` 的屎山代码得到优化，故记录

`Exercise 2` 的情景如下：

现在有两个类： `IntHistogram` 类 和 `StringHistogram` 类， 留下本讲需要的关键结构后长这样：

```
// Class IntHistogram
public class IntHistogram {
    public double estimateSelectivity(Predicate.Op op, int v){...}
    ...
}
```

```
// Class StringHistogram
public class StringHistogram {
    public double estimateSelectivity(Predicate.Op op, String s){...}
    ...
}
```

现在有一个 `Field` 类型的数组： `ArrarList<Field> arr` , 而 `Field` 有两个子类 `IntField` 和 `StringField`

你需要做的是：

根据 `Field` 子类的类型选择不同的 `Histogram` ， 然后将它们存在一个容器中，之后会不断的调用 `estimateSelectivity` 方法.

但是由于 `IntHistogram` 类 和 `StringHistogram` 类都没有继承父类并且方法的参数也不一样， 所以最开始新建了一个父类搞来搞去，最后成了屎山.....

? 无界通配符

最直接的理解就是 `?` 可以用来表示所有的类型，我们之前在定义 `ArrarList` 的时候， 需要在 `<>` 中指定这个数组中要存入的类型，就比如 `ArrarList<String> arr`

但是有些时候我们不知道要存入的类型，或者说我们想在一个数组中存储不同的类

当然你可以直接 `ArrarList<Object> arr` ， 但是还有另外一种更加优雅的方式 `ArrarList<?> arr`

但是这样操作也是有代价的，最主要的就是 **不能在这个数组中插入元素了**

```
// Error
public static void main(String[] args) {
    ArrayList<?> arr = new ArrayList<>();
    arr.add("error"); arr.add(-1)
}
```

所以无界通配符一般用于作为传入的函数中

```
public static void printf(ArrayList<?> arr) {
    for(Object o : arr)
        System.out.print(o + " ");
    System.out.println();
}
public static void main(String[] args) {
    ArrayList<String> arr = new ArrayList<>(
        List.of(new String[]{"aaa", "bbb", "ccc"}));
    ArrayList<Integer> arr2 = new ArrayList<>(
        List.of(new Integer[]{1, 2, 3}));
    printf(arr); printf(arr2);
}
/*
aaa bbb ccc
1 2 3
*/
```

基本的用法就这...主要是灵活运用

`<? extends E>` 上界通配符

在无界通配符中，`ArrayList<?> arr` 会存入所有继承自 `Object` 类的子类

但是如果我们在后面使用了 `extends E` ， 那么就只能存入 `E` **以及** `E` **的子类了**

不过还是要注意： **不能在这个数组中插入元素了**

`<? super E>` 上界通配符

与上界通配符类似， 但是这次是只能存入 `E` **以及** `E` **的父类了**

注意， 这个时候是**可以插入数据的**

当我们将 `E` 变成 `Object` 的时候就可以完成许多神奇操作

回到我们上面提到的问题，在学完通配符后有以下两种方式：

法一：

```
public static ArrayList<? super Object> arr = new ArrayList<>();
public double estimateSelectivity(Predicate.Op op, Object v, int i) {
    if(arr.get(i).getClass().equals(IntHistogram.class))
        return ((IntHistogram) arr.get(i)).estimateSelectivity(Predicate.Op op,
(int) v);
    else
        return ((StringHistogram) arr.get(i)).estimateSelectivity(Predicate.Op
op, (String) v);
}
```

法二：

```
public static ArrayList<? super Object> arr = new ArrayList<>();
public double estimateSelectivity(Predicate.Op op, String v, int i) {
    return ((StringHistogram) arr.get(i)).estimateSelectivity(Predicate.Op op,
v);
}
public double estimateSelectivity(Predicate.Op op, int v, int i) {
    return ((IntHistogram) arr.get(i)).estimateSelectivity(Predicate.Op op, v);
}
```

尾声

其实不知道你有没有发现其实不用通配符 `?` 也可以实现， 即改成 `ArrayList<Object> arr`.....

但感觉这样写更帅哈哈哈

不过八当初的空缺弥补了，很开心.....