## Optinal的理解

12.视频学习笔记

Optinal的是一个独立的概念，他出现的目的就是解决NPE的问题。在Java中NPE的问题很容易经常出现，如果是一个空对象调用他的属性了，如果是这个对象没有就会出现空指针的问题了。

在程序中经常会出现下面的代码

If (null != person) {

Address address = person.getAddress();

If(null != address) {

…..

}

}

String str = null; 其实null在代码中就是什么都没有指引，是谁提出来的，可以查一下，然后很后悔提出这个NPE的情况，现在很多语言都对NPE的这种情况做规避。

这里我们正式开始学习Optinal，翻开代码，从类注释开始学习。

1.8版本出现。Optional是一个容器对象，这个容器对象可能包含也可能不包含一个非空值。如果这个值存在的话，那么isPresent()方法就是true，get()方法就会返回这个值。

也提供了一些其他的方法，这些方法也提供这个值。如orElse()。

不要在Optinal中使用 == 等一些比较操作，因为他是基于值的一个类。这个里面有个链接。解释了value-based class。链接如下所示：

docs.oracle.com/javase/8/docs/api/java/lang/doc-files/ValueBased.html

总结来说：Optinal里面包一个值，这个值就是T。没有构造方法，使用of等一些方法来构造值。

empty()构造出容器里面为空的一个对象。

of() 方法里面会new Optional() 方法，一定要确保T的参数是要有的。

ofNullable() 方法二者折中的一个方法。

isPresent() 判断这个对象是否存在。

get() 直接就是取出这个值。

下面进行练习一下Optional的一些代码。

Optional op = Optional.of("hello");

if(op.isPresent()) {

System.out.println(op.get());

}

首先写出上面的例子，必须isPresent和get方法一起使用，否则get可能会出错误。

但是，以前NPE判断的时候写代码跟现在是基本一样的。以前是判断不为空，取值。而现在是判断是否存在，然后取值，并且代码还有点长。下面就说推荐的写法。

根据IDEA的提示推荐函数式的编程风格。如下面所示。

op.ifPresent(item -> System.out.println(item));

底层会先检查如果没有值就什么都不做，如果有值才做里面的函数。

接下来可以看如果没有值获取别的值的方法。

op.orElse("world")

op.orElseGet(() -> "nihao")

比如说你从远端获取要给对象，就需要of ()方法。如果很确定不为空就使用ofNullable()方法。

下面看Optiona的第二个例子。

如果一个方法返回一个集合的时候，不返回null，而是返回一个空的集合。这个例子设计一对多的关系，如公司和员工的关系。新建一个Employee和Company类。

Employee ->> private String name;

Company ->> private String name; private List<Employee> employeeList;

案例代码开始：

Empley em1 = new Employee();

em1.setName(“zhangsan”);

Empley em2 = new Employee();

em2.setName(“zhangsan”);

Company com = new Company();

com.setName(“com1”);

com.setEmployeeList(Lists.newArrayList(em1, Em2));

传统方式做法：

List<Emploee> list = company.geEmployee();

if(null != list) {

return list;

} else {

return new ArrayList();

}

Optinal的做法：

Optional<Company> op = Optional.foNullable(company);

System.out.println(op

.map(theCom -> theCom.getEmployList())

.orElse(Collections.emptyList()));

上面的这种做法是典型的函数式编程写法，map里面返回一个集合，如果为空就返回一个空的集合。