东北林业大学 李全欣

java程序设计 课程评价 任务 统计 资料 通知 作业 考试 分组任务(PBL) 讨论 体验新

P6-Optional-编程题 返回

姓名：李全欣 班级：计算机2018级4-7 成绩：100分

一.简答题（共3题,100.0分）

1 以下类

```
public class Course {
    private String name;
    private boolean elective;
    private Teacher teacher;
    // getter/setter
}

public class Teacher {
    private String name;
    // getter/setter
}
```

说明：

基于Optional操作

需求描述中的指定X，均指方法的参数

有返回值的方法，尝试直接编写return语句，基于Optional操作直接返回所需结果

尝试使用简写

注意过滤代码格式

方法1，基于指定课程，打印课程名称

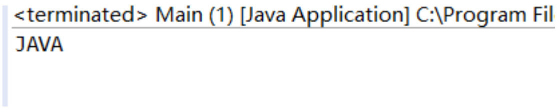
正确答案：

```
/**
    * 基于指定课程，打印课程名称
    * @param course
    */
   private static void print(Course course) {
      Optional.ofNullable(course)
            .ifPresent(c -> System.out.println(c.getName()));
   }
```

我的答案：

```
import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import java.util.Scanner;

class Course {

   private String name;
   private boolean elective;
   private Teacher teacher;
```

```
 public Course(String name, boolean elective, Teacher teacher) {
 this.name = name;
 this.elective = elective;
 this.teacher = teacher;
 }

 public String getName() {
 return name;
 }

 public void setName(String name) {
 this.name = name;
 }

 public boolean isElective() {
 return elective;
 }

 public void setElective(boolean elective) {
 this.elective = elective;
 }

 public Teacher getTeacher() {
 return teacher;
 }

 public void setTeacher(Teacher teacher) {
 this.teacher = teacher;
 }

}

class Teacher {

 private String name;

 public Teacher(String name) {
 this.name = name;
 }

 public String getName() {
 return name;
 }

 public void setName(String name) {
 this.name = name;
 }

}
public class Main {

public static void main(String[] args) {
printCourse(new Course("JAVA", true, null));
}

private static void printCourse(Course course) {
Optional.ofNullable(course)
.map(Course::getName)
.ifPresent(System.out::println);
}
}
```



```
<terminated> Main (1) [Java Application] C:\Program Fil
JAVA
```

2 方法2，基于给定课程，如果课程不是选修课，返回课程的名称

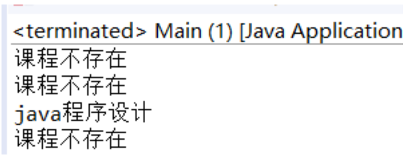
任何不符合条件，返回 课程不存在

正确答案：

```
/**
    * 基于给定课程，如果课程不是选修课，返回课程的名称
    * 任何不符合条件，返回 课程不存在
    * @param course
    * @return
    */
   private static String filter(Course course) {
       return Optional.ofNullable(course)
               .filter(c -> !c.isElective())
               .map(Course::getName)
               .orElse("课程不存在");
   }
```

我的答案：

```
public class Main {

 public static void main(String[] args) {


       Course c1 = new Course("大数据开发技术",true,new Teacher("老师A"));
 Course c2 = new Course(null,true,null);
       Course c3 = new Course("java程序设计",false,new Teacher("老师B"));
       Course c4 = new Course(null,false,new Teacher(null));

       List<Course> Courses = new ArrayList<>();
 Courses.add(c1);
 Courses.add(c2);
 Courses.add(c3);
 Courses.add(c4);

 Courses.forEach(c->{
 System.out.println(notElective(c));
 });
 }

 public static String notElective(Course course) {
 return Optional.ofNullable(course)
 .filter(c->!c.isElective())
 .map(c->c.getName())
 .orElse("课程不存在");
 }
}
```



```
<terminated> Main (1) [Java Application
课程不存在
课程不存在
java程序设计
课程不存在
```

3 方法3，基于给定课程，如果课程是选修课，返回课程的授课教师的姓名

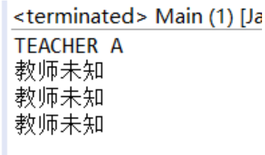
任何不符合条件，返回 教师未知

正确答案：

```
/**
    * 基于给定课程，如果课程是选修课，返回课程的授课教师的姓名
    * 任何不符合条件，返回 教师未知
    * @param course
    * @return
    */
   private static String map(Course course) {
       return Optional.ofNullable(course)
               .filter(Course::isElective)
               .map(Course::getTeacher)
               .map(Teacher::getName)
               .orElse("教师未知");
   }
```

我的答案：

```
public class Main {

 public static void main(String[] args) {

 Course c1 = new Course("大数据开发技术",true,new Teacher("TEACHER A"));
 Course c2 = new Course(null,true,null);
        Course c3 = new Course("java程序设计",false,new Teacher("TEACHER B"));
        Course c4 = new Course(null,false,new Teacher(null));

        List<Course> Courses = new ArrayList<>();
 Courses.add(c1);
 Courses.add(c2);
 Courses.add(c3);
 Courses.add(c4);


 Courses.forEach(c->{
 System.out.println(isElective(c));
 });

 }

 public static String isElective(Course course) {
 return Optional.ofNullable(course)
 .filter(c->c.isElective())
                .map(c->c.getTeacher())
 .map(c->c.getName())
 .orElse("教师未知");

 }
}
```



```
<terminated> Main (1) [Ja
TEACHER A
教师未知
教师未知
教师未知
```