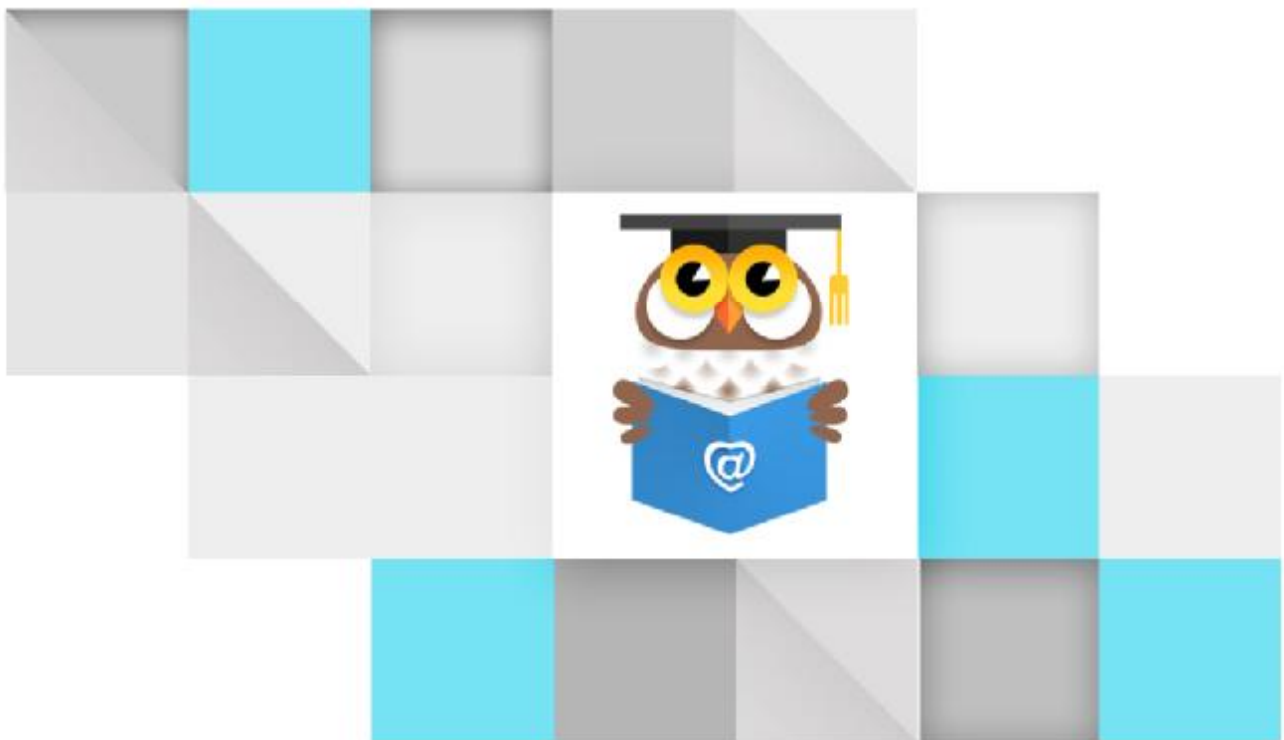


消灭泡泡糖 (Java)

实训指导手册

实训场景 007 – 体验接口隔离性



Campus Solution Group

目 录

一、任务编号：PRJ-BU2-JAVA-007	1
1、实训技能	1
2、涉及知识点	1
3、实现效果	1
4、场景说明	2
5、快速开始	4
6、任务 1 – 处理泡泡糖点击事件	5
7、任务 2 – 实现点击-消除效果（不考虑移动）	7
8、场景总结	10

一、任务编号：PRJ-BU2-JAVA-007

1、实训技能

- I Java 面向对象编程技能

2、涉及知识点

- I 事件接口
- I 内部类与作用域
- I 内部类的特殊语法规则
- I 局部内部类

3、实现效果



图 3-1

4、场景说明

1、业务说明：

1-1. 本场景主要用于体验接口的隔离性。一个类对另外一个类的依赖性应当是建立在最小的接口上的。

1-2. 通过鼠标点击事件完成泡泡糖的消除功能。

1-3. 隔离说明：基于接口的隔离性，本场景只需要关注鼠标的点击事件。

2、实现思路：

2-1. 创建一个处理点击事件的类并实现相应的事件接口。

2-2. 重写事件处理方法的逻辑，完成泡泡糖的消除功能。

3、核心组件介绍：

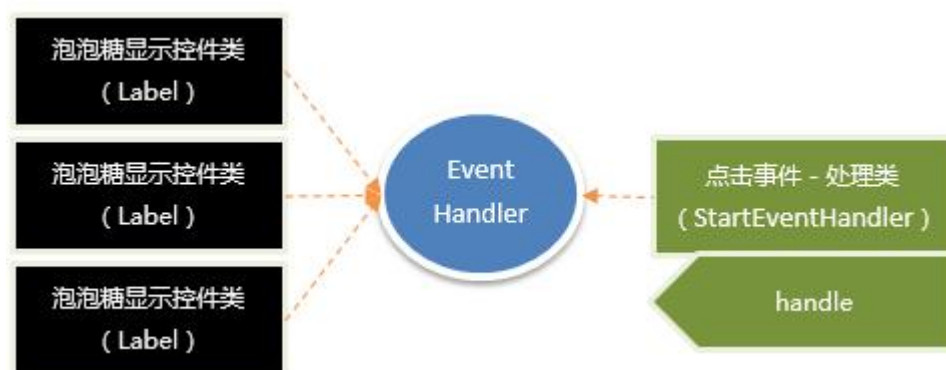


图 4-1

3-1. MainForm - 游戏界面类：

负责游戏数据显示、响应用户在界面上的各类操作。

3-2. StartEventHandler – 泡泡糖点击事件处理类：

当某个泡泡糖显示控件 - Label被点击后，点击事件需要交由该类处理。

因此，Label初始化时，需要将【点击事件处理】对象传递给每个Label对象。

3-3. StartEventHandler – handle：

【点击事件处理】类中负责处理点击事件的功能函数。

当某个泡泡糖显示控件 - Label被点击后，该方法需要完成以下操作：

3-3.1. 从图 4-1可知，该【点击事件处理】类会被多个Label共用，因此handle方法首先需要判断哪个Label被点击了。

3-3.2. 调用StarService接口的业务实现类，并访问toBeClearedStars函数获取【待消除泡泡糖】集合。

3-3.3. 在界面上清除【待消除泡泡糖】对象 - Star对应的显示控件 - Label。

3-4. EventHandler – JavaFX提供的点击事件处理接口：

3-4.1. JavaFX为Java界面开发提供了大量实用的显示控件，例如《消灭泡泡糖》游戏中广泛使用的Label控件、按钮控件、下拉框控件、菜单控件、单选与复选控件等。

3-4.2. 游戏界面通常需要接收大量的用户操作行为，例如：点击操作、鼠标滑动、键盘按键等，JavaFX将这些操作理解为**事件**。

3-4.3. 由于JavaFX无法知晓每个操作行为该如何配套业务处理逻辑，因此它提供了一组用户操作行为的**处理标准**（事件处理接口），并明确告知开发者通过实现这些标准，来应对用户的各类操作行为。

3-4.4. EventHandler就是JavaFX提供的处理【点击】事件的标准接口，本场景将主要通过编写事件处理类**StartEventHandler**来实现该接口，并妥善处理用户对于泡泡糖显示控件 - Label的点击操作行为。

3-4.5. 这种设计方法，就是利用了接口的隔离性，它将【显示控件】与【控件事件处理】两大组件进行了分割，确保它们可以分别由不同的研发人员按需实现。

4、了解更多：

请参考《消灭泡泡糖 - 需求说明文档》。

5、前置条件：

5-1. 前置场景：PRJ-BU2-JAVA-006 – 体验接口解耦特性。

5-2. 必备知识与技能：

5-2.1. Java面向对象编程技能（类、方法调用、接口、重写方法、内部类）。

5、快速开始

1、开发环境：

1-1. Oracle JDK8.x 以上版本

1-2. Eclipse Luna (4.4.x) 以上版本

1-3. 工程包：PRJ_BU2_JAVA_007

2、进入开发环境：

详见SPOC平台上《PRJ-BU2-JAVA-007 前置任务：进入开发环境》



图 5-1

6、任务 1 – 处理泡泡糖点击事件

1、任务描述：

- 1-1. 创建Label控件【点击事件处理类】，并实现【点击事件】处理接口EventHandler。
- 1-2. 由于该类服务于游戏界面，因此将【点击事件处理类】创建于MainForm窗体类内部。
- 1-3. 依次把【点击事件处理】对象，绑定到界面中的每个泡泡糖显示控件 - Label上。

2、推荐步骤：

- 2-1. 定位到类：cn.campsg.practical.bubble.MainForm。
- 2-2. 新建一个内部类StartEventHandler，并实现事件处理的接口EventHandler。
- 2-3. 重写接口handle方法，该方法用来处理并响应用户对于Label控件的点击操作。
- 2-4. 处理Label控件的点击操作：

+ **业务说明**：由于所有Label控件都将共用同一个【事件处理类】StartEventHandler，因此handle方法首先需要知晓用户点击了哪个Label控件对象。

- 2-4.1. 通过handle方法的参数：鼠标事件类 - MouseEvent，获取被点击的泡泡糖显示控件 - Label对象。

+ 提示

- 1) 本处使用了JavaFX的知识，请按以下代码实现任务需求：

Label starFrame = (Label) event.getTarget(); //获取被点击的泡泡糖控件

- 2) 由于显示泡泡糖的控件是Label，因此我们需要将鼠标事件返回的对象强制类型转换成Label对象。

- 2-4.2. 将被点击的泡泡糖视图控件转换成泡泡糖对象 - Star。

+ 提示

1) 本处使用了JavaFX的知识，请按以下代码实现任务需求：

```
Star base = StarFormUtils.convert(starFrame); // 将视图转换为泡泡糖对象
```

2) 以上代码通过StarFormUtils类的convert方法，把界面Label对象转换为Star对象。

3) 场景PRJ-BU2-JAVA-006中，我们已经实现类【泡泡糖对象】Star转换成【泡泡糖显示控件】Label的业务，这里执行反向转换的逻辑与场景006的逻辑一致，因此不再重复实现，有兴趣的用户可以去了解一下convert的实现代码。

2-4.3. 向控制台输出该泡泡糖对象的属性信息

2-5. 定位到该类的initGameStars方法中，找到相应注释处。

2-6. 创建一个【点击事件处理类】对象。

2-7. 将【点击事件处理类】对象设置给每个泡泡糖视图控件 – Label对象。

+ 提示

1) 将【点击事件处理类】对象设置给泡泡糖视图控件 – Label，可以通过调用Label对象的setOnMouseClicked方法实现。

2) 由于2-5, 2-6被编写在for循环体内，因此，我们可以方便的给每个泡泡糖视图控件 – Label对象设置点击事件。

3、验证与测试：

3-1. 定位到程序入口类：cn.campsg.practical.bubble.MainClass。

3-2. 运行该项目，使用鼠标点击界面中任意一个泡泡糖，观察控制台是否会输出该泡泡糖的（坐标-类型）信息：

```
(0,1-PURPLE)
(0,3-YELLOW)
(0,2-RED)
(2,1-GREEN)
(2,0-BLUE)
```

图 6-1

7、任务 2 – 实现点击-消除效果（不考虑移动）

1、任务描述：

1-1. 点击界面上任意泡泡糖，界面实现消除泡泡糖操作（不考虑移动）。

1-2. 执行流程如下：

1-2.1. StartEventHandler获取被点击的泡泡糖显示控件（任务1已完成）。

1-2.2. 将显示控件-Label转换为泡泡糖对象Star（任务1已完成）。

1-2.3. 调用服务类StarServiceImpl的tobeClearedStars函数，获取所有【待消除泡泡糖】集合。

1-2.4. 界面执行清除操作。

2、推荐步骤：

2-1. 定位到内部类：cn.campsg.practical.bubble.MainForm.StartEventHandler。

2-2. 创建一个成员变量，变量类型为StarService。

2-3. 创建一个带参构造器，用于给StarService成员变量赋值。

+ 业务说明

1) 从任务描述已知，点击事件需要通过调用服务类StarServiceImpl的tobeClearedStars函数，获取所有【待消除泡泡糖】集合。因此，我们需要将StarServiceImpl对象传递给【点击事件处理】类。

2) 传递StarServiceImpl对象给StartEventHandler的最佳途径，是在实例化【点击事件处理】类时，通过构造函数传递。

2-4. 在StartEventHandler类中定位到handle方法中

2-4.1. 删掉任务1中用于测试的控制台输出语句。

2-4.2. 调用业务对象的tobeClearedStars方法，获取待消除的泡泡糖列表。

2-4.3. 判断toBeClearedStars方法返回的【待消除的泡泡糖列表】：

1) 如果该集合为null或者集合长度等于0，那么退出点击事件的处理。

2-4.4. 使用for循环遍历toBeClearedStars返回的【待消除的泡泡糖列表】。

2-4.5. 在循环体中进行清除操作：

1) 通过循环变量从【待消除列表】中依次获取每个【待消除的泡泡糖】对象。

2) 在游戏界面中找到【待消除的泡泡糖】- Star对应的【泡泡糖显示控件】- Label。

+ 提示

1) 本处使用了JavaFX的知识，请按以下代码实现任务需求：

```
Label frame = StarFormUtils.findFrame(star, mStarForm);
```

2) 以上代码通过star对象的相关属性信息，从界面容器中找到相应的泡泡糖控件对象。

3) 执行清除动画，并清除界面上【待消除的泡泡糖】控件 - Label。

+ 提示

1) 本处使用了JavaFX的动画知识，请按以下代码实现任务需求：

```
StarAnimation.clearStarLable(mStarForm, frame);
```

2) 以上代码将会执行一组JavaFX提供的动画代码，以渐入渐出的动画方式清除界面上【待消除的泡泡糖】。

3) 对于JavaFX动画感兴趣的用户可以查看clearStarLable函数。

4) 同步清除【完整的泡泡糖列表】的泡泡糖对象。

+ 提示

1) 清除集合中的泡泡糖，可调用StarList集合中：setNull方法。

2) setNull需要知晓被置空的泡泡糖对象的行值与列值，请直接将2-4.5步骤获取的泡泡糖的行值与列值传递给setNull函数。

+ 业务说明

- 1) 【完整的泡泡糖列表】用于对应游戏界面的泡泡糖矩阵。
- 2) 当游戏界面上的泡泡糖被消除后，【完整的泡泡糖列表】中的对象也需被同步置空（设置null），以此表述该泡泡糖不再可用。
- 3) 业务类其他功能函数在判断【完整的泡泡糖列表】时，如果遇到了null值，便可知该位置的泡泡糖已经被清除。
- 4) 【完整的泡泡糖列表】与游戏界面的泡泡糖矩阵对应关系如图7-1：

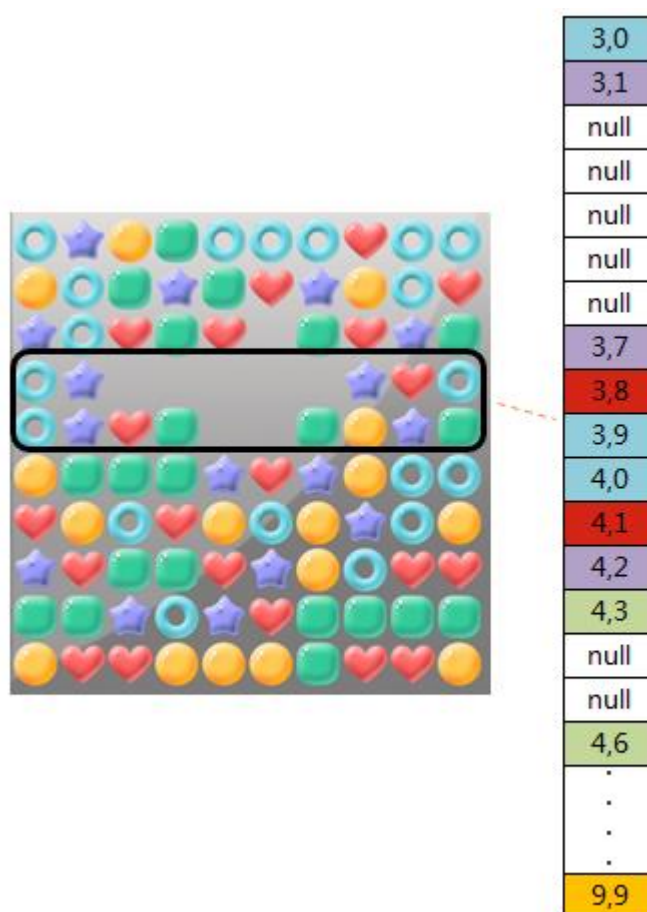


图 7-1

2-5. 定位方法：MainForm中的initGameStars方法中。

2-6. 修改任务1创建的【点击事件处理】对象：

2-6.1. 将StarServiceImpl对象作为参数，通过构造器传递给【点击事件处理】对象。

+ **提示**：StarServiceImpl对象已通过getService()获得。

3、验证与测试：

3-1. 定位到程序入口类：cn.campsg.practical.bubble.MainClass。

3-2. 运行该项目，点击任意泡泡糖，如泡泡糖四周有同色对象，这些对象将会被消除（不考虑移动）。



图 7-2

8、场景总结

无