

# Lab11

TA:

- 14302010005@fudan.edu.cn 姜卓立
- 14302010040@fudan.edu.cn 武多才
- 14302010042@fudan.edu.cn 何培剑

## 1 摘要

---

1. 学习并熟练掌握抽象类、接口
2. 使用上述知识点重构LAB10的代码

## 2 抽象类与接口

---

### 2.1 抽象类

```
public abstract class TestAbstract {  
    public abstract void method1();  
  
    public void method2() {  
        System.out.println("superclass method2");  
    }  
}
```

抽象方法的修饰符不可为private。

```
public class Test extends TestAbstract {  
    @Override  
    public void method1() {  
        System.out.println("subclass method1");  
    }  
}
```

子类需要重写抽象父类中的所有抽象方法，否则子类依然为抽象类。

### 2.2 接口

```
public interface TestInterface {  
    void method1();  
  
    void method2();  
}
```

接口中的方法不能有方法体，默认且唯一的修饰符为public。

```
public class Test implements TestInterface {  
    @Override  
    public void method1() {  
        System.out.println("subclass method1");  
    }  
  
    @Override  
    public void method2() {  
        System.out.println("subclass method2");  
    }  
}
```

子类需要重写接口中的所有方法，否则子类为抽象类。

## 3 作业：使用抽象类和接口重构LAB10

---

### 3.1 示例

所有的物体都用一个char类型的值来显示。因此可以定义一个接口来表示有图标的物体。

```
public interface HasIcon {  
    char getIcon();  
}
```

地图由墙和道路构成，其区别在于道路可以通过，而墙不可以通过。先把两者归为一类。

```
public abstract class Place implements HasIcon {  
    protected int row, col;  
  
    public abstract boolean isWalkable();  
}
```

再实现具体类。

```

public class Wall extends Place {
    @Override
    public boolean isWalkable() {
        return false;
    }

    @Override
    public char getIcon() {
        return '■';
    }
}

```

```

public class Space extends Place {
    @Override
    public boolean isWalkable() {
        return true;
    }

    @Override
    public char getIcon() {
        return '\u00A0';
    }
}

```

在PJ中，大部分物体都是会触发事件的，例如怪物、宝物等。因此可以定义一个接口来表示会触发事件的物体。

```

public interface HasEvent {
    void handleEvent();
}

```

先把所有的物体归为一类，每个物体都有自己的位置。用一个抽象类来实现。

```

public abstract class Entity implements HasIcon, HasEvent {
    protected Place place;
}

```

接着把其中会移动的物体再归为一类，仍然用一个抽象类来实现。

```

public abstract class Mob extends Entity {
    public abstract void move();
}

```

最后会动的物体，例如玩家、怪物等，就可以继承Mob类。

```
public class Player extends Mob {  
    /**  
     *  
     * override methods...  
     *  
     */  
}
```

而不会动的物体，例如宝物等，则直接继承Entity类。

```
public class Treasure extends Entity {  
    /**  
     *  
     * override methods...  
     *  
     */  
}
```

## 3.2 要求

本次LAB中，不要求同学们按照TA的思路来重构，上面的代码仅作为一种思路。鼓励同学们根据自己的逻辑来重构，尽量使用抽象类与接口。

## 4 提交

---

1. 提交地址：**ftp://10.132.141.33/classes/17/171 程序设计A(戴开宇)/WORK\_UPLOAD/lab11/**
2. 提交物提交步骤：请同学们把源码打包成**lab11\_你的学号.zip**
3. 提交要求与前两个LAB相同，除了源码之外，还需提交一个jar包和bat文件。
4. Deadline: 2017年12月10日23:59:59

## 5 声明

---

任何形式的作业都欢迎同学们相互讨论，但抄袭是严格禁止的。一旦发现抄袭行为，抄袭者和被抄袭者都以0分处理