



## 实验五、接口与异常

### 实验目的：

- 1) 理解接口、多态的基本概念。
- 2) 体会接口与实现分离设计的优点。
- 3) 通过实验内容深刻理解“接口表示一种能力”和“接口表示一种约定”。
- 4) 理解 throws、try、catch、finally 等语句的语法格式和使用。
- 5) 理解自定义异常类的定义和使用方法。
- 6) 掌握 Java 的异常处理机制、方法与应用。

### 实验要求：

能够根据接口的要求实现具体的方法；能够在同样的接口下要求下体现不同的内部实现；能够判断对象所属的接口类型，并根据相应类型进行正确的操作；能够处理接口与抽象类的关系；能够用接口解决实际问题。

### 实验任务：

【任务一】：假设默认长度单位为米，编写一个计算为各种形状着色花费的程序。

#### 要求：

1. 编写 BorderColorable 接口，要求：

- 具有方法 void paintBorder()，在实现类中该方法需要输出“已经将什么形状的边着色”。
- 方法 boolean isBorderPainted()，返回形状的边是否已经着色。

2. 编写 SolidColorable 接口，要求：

- 具有方法 void paintShape()，在实现类中该方法需要输出“已经将什么形状进行着色”。
- 方法 boolean isShapePainted()，返回形状是否已经着色。

3. Shape2D 是一个抽象类，是所有形状的父亲类：

- Shape2D 具有抽象方法 double getPerimeter()和 double getArea()分别表示获取形状的周长和面积。
- 其中圆(Circle)，三角形(Triangle)和长方形(Rectangle)均继承自 Shape2D 类，而 Square 继承自 Rectangle 类。
- Circle 实现了 BorderColorable 接口，Triangle 实现了 SolidColorable 接口，而 Rectangle 同时实现了 BorderColorable 接口和 SolidColorable 接口。

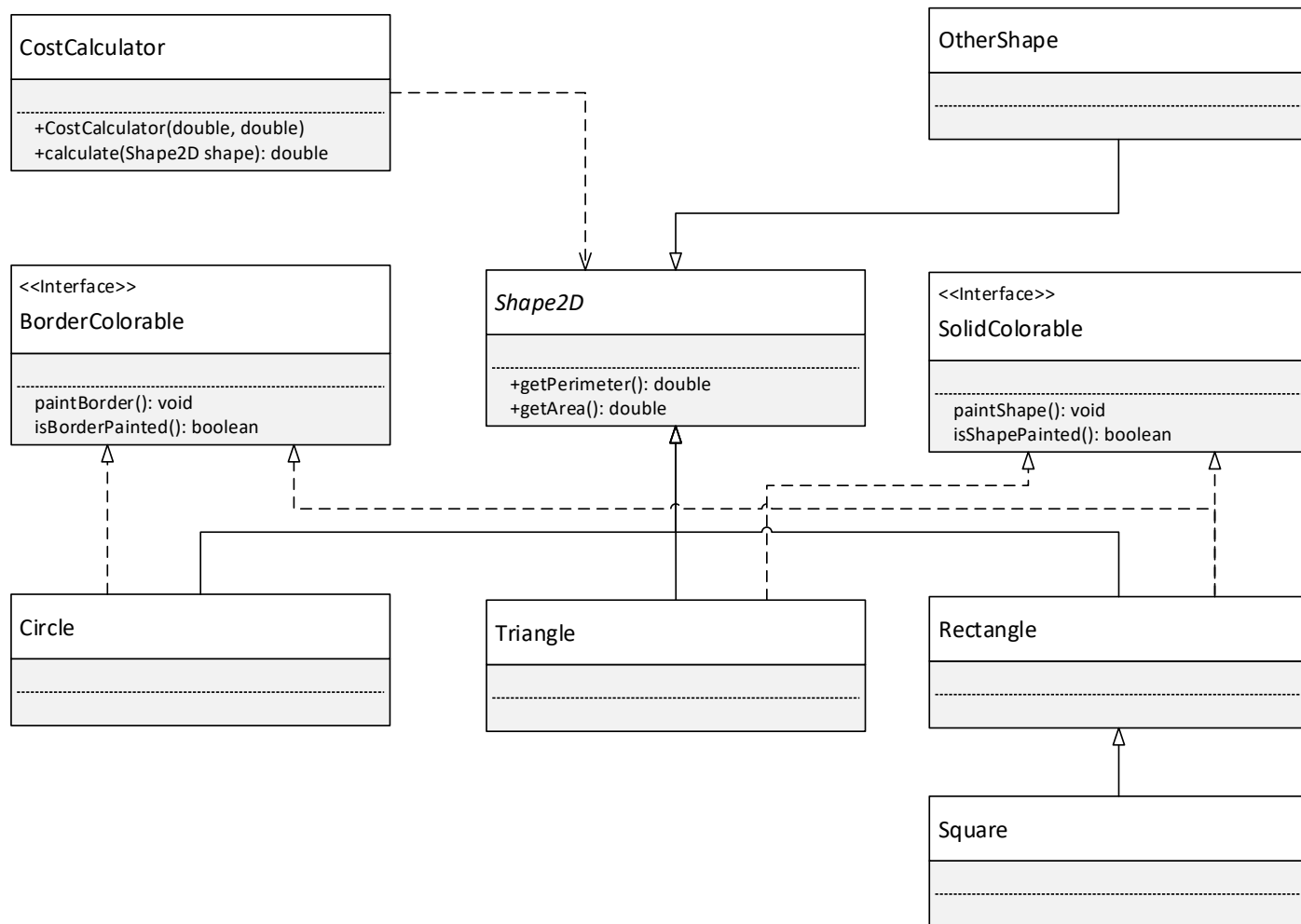
4. CostCalculator 类用来计算为边着色和为形状着色所消耗的花费。

- 构造函数 CostCalculator(double borderCost, double solidCost)，其中 borderCost 表示画 1 米的边需要的单价，solidCost 表示画 1 平方米的边需要消耗的单价。
- double calculate(Shape2D shape)用来计算实际为形状 shape 着色需要的花费。
- 形状着色的花费计算只计算已经着色的部分，且花费总额为边着色和为形状着色的花费的总和。

5. 写一个测试类测试你的程序。

#### 提示：

1. 实现了 BorderColorable 的形状只为边着色，实现了 SolidColorable 的形状为整个形状着色。
2. 总体类图如下图所示。



【任务二】：编写一个程序引起 JVM 的 `OutOfMemoryError`。

要求：

1. 在程序中不断分配内存，并最终引起 JVM 的 `OutOfMemoryError` 错误。
2. 用 `try...catch` 捕捉这个错误，然后查看此时虚拟机总内存和空闲内存是多少。
3. 在异常处理中试图清除已经分配的内容空间，尝试恢复错误。
4. 在错误恢复后，再次查看总内存和空闲内存。

提示：

1. 尝试使用 Java 的 `ArrayList` 类。
2. 可以使用 `Runtime` 类的 `freeMemory()` 方法查看空闲内存。
3. 使用 `Runtime` 类的 `totalMemory()` 方法查看总内存。（`maxMemory()` 方法可以查看最大可占用内存。）
4. 试图恢复这个异常的时候可以清空列表对象，并使用 `System.gc()` 方法，请求虚拟机进行垃圾回收。
5. 请通过搜索引擎了解 Java 虚拟机的内存分配原则。

【任务三】：尝试在上一次实验日记本程序中应用面向接口的编程方式。

要求：

1. 在适当的地方使用接口，例如日记的存储、读取等，设计接口，提供读取、查找、添加日记的能力。只考虑未来需要什么功能，暂时先不实现接口。未来接口可能由不同的类来实现完成不同的功能，比如从网络读取日记、从数据库读取日记或者从普通文件读取日记。
2. 在程序中使用异常处理，程序在运行过程中不能因为异常而中断运行。适当使用异常作为程序流程处理机制，并



且在异常出现时，程序能够提供足够的出错细节信息帮助判断调试。