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】：定义一个 ReList 接口，并使用数组和链表两种方法来实现接口。

要求：

1. MyList 接口包含六个方法：

- void add(Object obj)：往列表尾部添加对象
- void addFirst(Object obj)：向列表头部添加对象
- Object get(int index)：从列表中获取索引为 index 的对象
- void clear()：清空所有的对象
- boolean isEmpty()：判断列表中是否有对象
- int size()：获取列表中对象的个数
- int capacity()：所分配的空间大小

2. MyArrayList 类实现 MyList 接口，内部以数组的方式实现，要求：

- 构造函数 ReArrayList(int incSize)：参数 incSize 为数组初始化大小和空间的增量。若用户调用 incSize 非法，则设为**默认值 5**。
- 当调用 add()和 addFirst()方法往试图往 MyArrayList 中增加对象时，如果内部数组已满，则增加数组大小，增量为 incSize。
- 调用 clear()方法可以清空所有通过 add()方法加入的对象。
- 调用 get()方法时，如果传入的参数非法，则返回 null 对象，并使用 System.err 提示用户参数错误。

3. MyLinkedList 类实现 MyList 接口，要求内部使用数组方式实现，链表要求自己实现，不允许直接使用 Java 自带的链表类。

【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. 合理利用上面的 MyList 接口的实现。

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。

【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。