抽象数据类型概述

抽象数据类型（Abstract Data Type, ADT）是软件构造过程中的一个重要实例，与传统的数据类型不同的是，抽象强调作用于数据上的操作，程序员和用户无需关心数据是如何存储的，只需要设计和使用该数据类型即可。值得注意的是，**抽象数据类型是由操作定义的，与其内部的实现机制无关。**

抽象数据类型中的操作

1. 可变类型&不可变类型

可变类型对象是指**提供可改变其内部数据值的操作的方法的对象。**而不可变类型的对象则是通**过构造新的对象的方式"改变"其内部的数据的值。**

2. 抽象数据类型的方法

抽象数据类型的方法可以分为一下四种。第一种，构造器（Creator），是指**为了创造某个类型的新对象的方法，可能实现为构造函数或静态函数（通常称为工厂类方法）**；第二种，生产器（Producer），是指能从该类型的**旧对象中创建新对象的方法**；第三种，观察器（Observer），是指能够**获取对象类型并返回不同类型的对象的方法**；第四种，变值器（Mutator），是指能够**改变对象属性的方法。**具体如下图所示：

下面给出几个功能显而易见的函数及它们所对应的类别：

Integer.valueOf()	Creator
BigInteger.mod()	Producer
List.addAll()	Mutator
String.toUpperCase()	Producer
Set.contains()	Observer
Collections.unmodifiableList()	Producer
BufferedReader.readLine()	Mutator

ADT的设计原则

1. 设计简洁，追求操作上的一致性；

2. 支持客户对于数据的操作的所有要求，且要尽可能降低用户使用这些方法的难度；

3. 数据类型要么是抽象的，要么是具体的，不可以将抽象和具体混为一谈；

ADT的测试

1. Creator、Producer以及Mutator：调用Observer方法来观察这些方法的结果是否满足规约条件；

2. Observer：调用Creator、Producer以及Mutator方法来产生或该拜年对象，查看其返回结果是否满足改动；

————————————————