**1.3 背包，队列和栈**

1.3.1 API

背包是一种不支持从中删除元素的集合数据类型。迭代的顺序不确定且与用例无关。

先进先出队列或简称队列，是一种基于先进先出策略的集合类型。

下压栈是一种基于后进先出策略的集合类型。

编程：计算算术表达式（表达式每次运算都包含括号）

Dijkstra算法，用两个栈，一个用于保存运算符，一个用于保存操作数。

将操作数压入栈；将运算符压入栈；忽略左括号；在遇到右括号时，弹出一个运算符，弹出所需数量的操作数，并将运算结果压入栈。

1.3.2集合类数据类型的实现

编程：下压栈的完整实例（数组实现）

接口：压栈，出栈，数量，是否为空

特性：支持迭代，动态调整数组大小

调整数组大小，push时候，如果没有多余的空间，我们会将数组的长度加倍，pop时，如果栈大小小于数组的四分之一，将数组长度减半。

1.3.3链表

链表是一种递归的数据结构，它或者为空，或者是指向一个结点的引用，该结点含有一个泛型的元素和一个指向另一条链表的引用。

private class Node{Item item; Node next;}

在链表中比较容易实现的操作：

在表头插入结点；从表头删除结点；在表尾插入结点。

不容易实现的操作：

删除指定的结点；在指定结点前插入一个新结点

实现任意插入和删除操作的标准解决方案是使用双向链表。

编程：下压堆栈（链表实现）

编程：先进先出队列（用链表实现）

在结构化存储数据集时，链表是数组的一种重要的替代方式。

编程：背包的实现

数组和链表分别称为顺序存储和链式存储。