《Python语言与编程》课程实验指导书

**实验二 Python面向对象编程**

1. 实验目的

学生通过使用Python进行面向对象编程，理解并掌握Python面向对象编程的基本方法，同时巩固加深对已学Python的知识的理解与掌握。

1. 实验内容

（详细内容见附件一）

1. 实验环境

Windows操作系统（或Mac操作系统），Python 3.6

1. 实验要求
2. 独立完成实验内容要求。
3. 使用Python的IDLE 进行编程，shell下演示结果（截屏保存）。
4. 需要上交源程序文件（以题目为编号）。
5. 实验学时

2学时

**附件一**

**1实现一个堆栈Stack类，堆栈(Stack)是一种具有后进先出(last-in-first-out，LIFO)特性的数据结构。要求实现以下功能并通过程序验证：**

1. **pushstack() 向堆栈中压入一个数据项**
2. **popstack() 从堆栈中移出一个数据项**
3. **isempty() 如果堆栈是空的，返回布尔值True,否则返回False**
4. **peekstack() 取出堆栈顶部的数据项，但并不移除它**
5. **printAll() 打印当前栈内所有元素，如果是空栈给出提示**

**2 实现一个队列Queue类，数据从前端被移除，从后端被加入。要求实现以下功能并通过程序验证：**

1. **enqueue() 在队列的尾部加入一个新的数据**
2. **dequeue() 在队列的头部取出一个数据，返回它并且把它从队列中删除**
3. **putfile() 将队列中数据写入文件中**
4. **getfile() 将文件中数据加入队列中**
5. **printAll() 显示当前队列内所有元素，如果是空队列给出提示。**

**3 实现一个集合Set类，集合是不允许有相同元素存在的数据结构，要求实现以下功能:**

1. **Set( aList ): 利用初始化数据（aList是一个数据列表）创建一个集合**
2. **addElement( x ): 将x加入到集合中**
3. **deleteElement( x): 将x从集合中删除。如果没有x，则集合保持不变**
4. **isMember(x): 判断x是否属于集合，返回True或False**
5. **intersection( Set2): 返回当前集合与集合Set2的交集**
6. **union(Set2): 返回当前集合与集合Set2的并集**
7. **substract(Set2):返回当前集合与集合Set2的差集（属于当前集合但不属于Set2的元素）**
8. **printAll() 显示当前集合内所有元素，如果是空集给出提示**

**4 现在想将常用的排序算法封装成一个算法工具类SortLib，请尝试实现。要求：**

1. **至少包含1种排序算法（如：冒泡排序）。**
2. **随机生成一个含20个整数的列表，来测试实现的工具类SortLib。**

附件二：

**北京邮电大学软件学院**

**2016－2017学年第二学期实验报告**

**课程名称： Python语言与编程**

**项目名称： Python面向对象编程**

**项目完成人：**

**姓名：\_\_\_**苏可欣**\_\_\_\_\_学号：\_\_2016211954\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**指导教师：\_\_\_\_管皓\_\_\_\_\_\_\_\_\_\_\_\_**

**日 期： 2018 年 5 月 26 日**

1. **实验目的**

（说明通过本实验希望达到的目的）

学生通过使用Python进行面向对象编程，理解并掌握Python面向对象编程的基本方法，同时巩固加深对已学Python的知识的理解与掌握。

1. **实验内容**

（说明本实验的内容）

（详细内容见附件一）

1. **实验环境**

（说明本实验需要的环境）

Windows操作系统（或Mac操作系统），Python 3.6

1. **实验结果**

（说明实验完成情况）

1实现一个堆栈Stack类，堆栈(Stack)是一种具有后进先出(last-in-first-out，LIFO)特性的数据结构。要求实现以下功能并通过程序验证：

1. pushstack() 向堆栈中压入一个数据项
2. popstack() 从堆栈中移出一个数据项
3. isempty() 如果堆栈是空的，返回布尔值True,否则返回False
4. peekstack() 取出堆栈顶部的数据项，但并不移除它
5. printAll() 打印当前栈内所有元素，如果是空栈给出提示

用list实现栈结构

用\_\_防止外部获得栈内部内容

用list的append实现在栈顶添加元素

用list的pop实现在栈顶删除函数

用len > 0判断是否为空

用list[-1]来获取顶部元素

2 实现一个队列Queue类，数据从前端被移除，从后端被加入。要求实现以下功能并通过程序验证：

1. enqueue() 在队列的尾部加入一个新的数据
2. dequeue() 在队列的头部取出一个数据，返回它并且把它从队列中删除
3. putfile() 将队列中数据写入文件中
4. getfile() 将文件中数据加入队列中
5. printAll() 显示当前队列内所有元素，如果是空队列给出提示。

使用list实现队列结构

使用\_\_保护队列内部元素不被获取

使用list.append实现在队尾添加元素

使用list.pop(0)实现删除队头的元素

文件操作用with open

3 实现一个集合Set类，集合是不允许有相同元素存在的数据结构，要求实现以下功能:

1. Set( aList ): 利用初始化数据（aList是一个数据列表）创建一个集合
2. addElement( x ): 将x加入到集合中
3. deleteElement( x): 将x从集合中删除。如果没有x，则集合保持不变
4. isMember(x): 判断x是否属于集合，返回True或False
5. intersection( Set2): 返回当前集合与集合Set2的交集
6. union(Set2): 返回当前集合与集合Set2的并集
7. substract(Set2):返回当前集合与集合Set2的差集（属于当前集合但不属于Set2的元素）
8. printAll() 显示当前集合内所有元素，如果是空集给出提示

使用python的set数据结构实现set类

构造函数可传入list对象，可能包含重复元素，使用set(list)将list转换成不重复无序的

set对象

使用set.add(),set.remove()实现在集合中添加删除元素

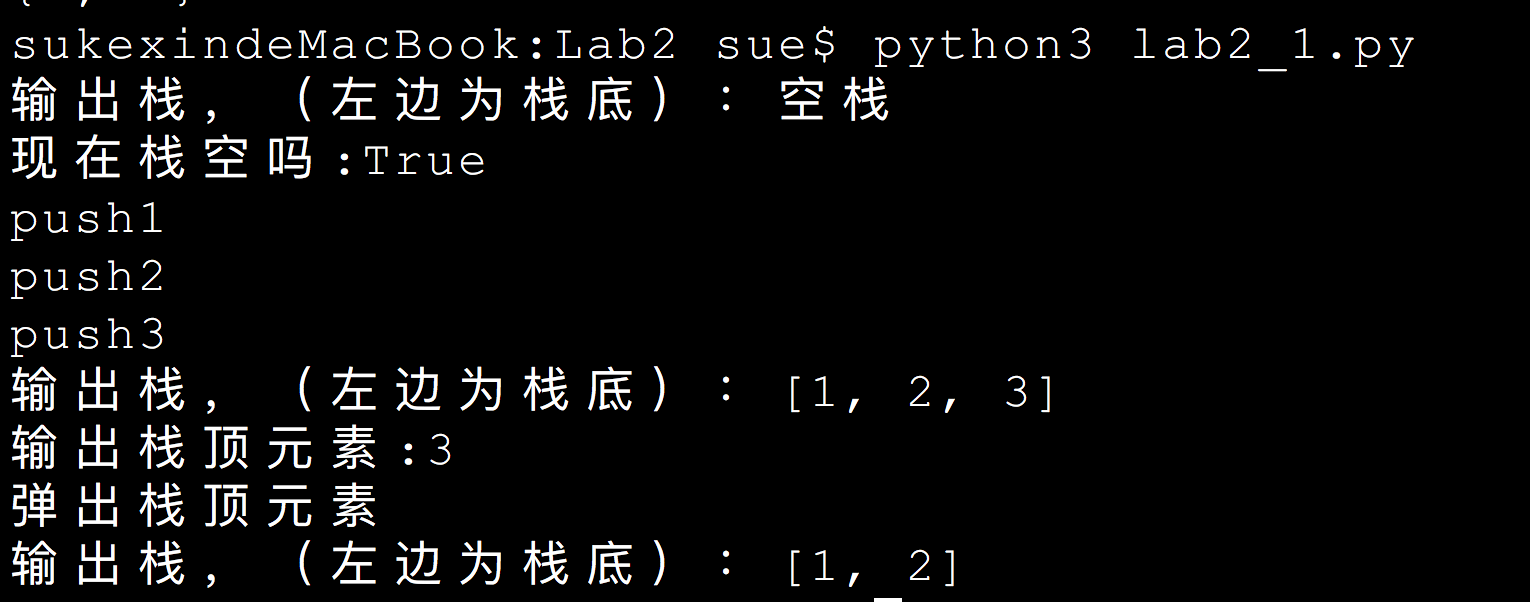
使用if x in set:实现判断是否是集合中的元素

使用set.intersection(set),union(set),difference(set)实现交并差

4 现在想将常用的排序算法封装成一个算法工具类SortLib，请尝试实现。要求：

1. 至少包含1种排序算法（如：冒泡排序）。
2. 随机生成一个含20个整数的列表，来测试实现的工具类SortLib。

实现了冒泡排序，插入排序，快速排序，选择排序

****

