

一、请根据 Map 的基本原理，设计一个基于 Python 内建的 MutableMapping 的名为 MapBase 的抽象基类（Abstract Base Class），需包含：

- 1) 内建类 Item，其内包含名为 key 与 value 的私有属性
- 2) 实现 Item 的操作符 ==
- 3) 实现 Item 的操作符 !=
- 4) 实现 Item 的操作符 <

二、根据抽象基类 MapBase 实现基于数组的字典，名为 UnsortedTableMap，需包含：

- 1) 构造函数：初始化内建数组
- 2) 函数 getitem(self, k) 实现在词典中通过键值 k 查找，注意：在这一步先不使用 “__”。如词典内不存在键值 k，则报错 KeyError
- 3) 函数 setitem(self, k, value) 实现更新键值为 k 的 (k, v) 对，如词典不存在 k，则创建新的 (k, value) 对将其填入到词典中
- 4) 函数 delitem(self, k) 实现删除键值为 k 的 (k, v) 对，如词典不存在 k，则报错 KeyError
- 5) 函数 __len__() 返回内建数组长度
- 6) 函数 __iter__() 返回一个迭代器
- 7) 将 getitem()、setitem()、delitem() 转化成覆盖特殊操作符 M[k]，M[k] = v，以及 del M[k] 的形式，测试其作用

三、编写自己的 hash function，要求存储 (str, str) 对实现词典

- 1) 通过将键值直接转成十进制整数的方法获得 hash code
- 2) 通过二进制移位的方法将键值转成整数获得 hash code，选取移位数量为 5
- 3) 通过多项式叠加法生成键值的 hash code，选取 $a = 33$
- 4) 通过使用 Division Method 实现压缩函数： $i \bmod N$
- 5) 通过适用 MAD 方法实现压缩函数： $[(ai + b) \bmod p] \bmod N$

四、编写自己的 Map 类，通过扩展抽象基类 MapBase 来实现。

- 1) 建立一个 MapBase 的子类内，建立可以实现 Separate chaining 的数据格式，即：内建数组，数组单元格存储二级容器实现哈希冲突的解决方式
- 2) 建立一个 MapBase 的抽象子类，支持开放寻址（open addressing）的方式。定义一个抽象函数 _find_slot，使该函数可以被子类函数覆盖。定义 _get_item(), _set_item(), _delitem() 函数。定义哨兵结点 _SENTINEL，在 _delitem() 执行时使用。
- 3) 建立一个名为 LinearProbHashMap 的子类，以 2) 内定义的类为父类，实现线性探索 (linear probing) 的算法
- 4) 建立一个名为 QuadraProbHashMap 的子类，以 2) 内定义的类为父类，实现二次探索 (quadratic probing) 的算法

五、写一个针对于数组的二分查找方法

六、创建一个类，实现跳表的 ADT