### week 1

**work in week1.ipynb**

1. 思考：python元组与列表的区别，举例说明使用的情况。

* 列表中的元素可以进行任意修改，而元组中的元素无法修改，除非替换掉元组整体。
* 可以理解为，元组是一个只读版本的列表。
  1. 需要增加、删减或者改变元素时，列表更好。
  2. 元组作为很多内置函数和序列类型方法的返回值存在。
  3. 元组可以在映射（和集合的成员）中当做“键”使用，而列表不行。

1. 学习使用列表，设计和编写程序，存储 1978-2010 上海市的GDP数据（或气温，降水，人口等），计算输出最大最小值，平均值和变异系数。

* 存储的数据在**1978-2010上海GDP.csv**内。

min GDP 2485

max GDP 78989

mean GDP 22519.71875

coefficient of variation in GDP 1.0172593066333713

### week 2

**work in week2.ipynb**

1. 顺序表实现GIS图层列表的管理，用Python语言设计一个简单LayerManager类，实现对图层对象layer的增加、删除、修改和排序等操作。

* 使用Python语言完成。
* Get(self,index)
* 返回index位置处的值，若index超出范围则print报错，并且返回空值。
* Insert(self,Input,index=-1)
* 在index位置处插入值，若没有输入index则默认在列表末尾插入。若index超出范围则print报错。
* Delete(self,index)
* 删除index位置处的值，若index超出范围则print报错。
* Modify(self,Input,index)
* 修改index位置处的值为Input，若index超出范围则print报错返回报错。
* Sort(self)
* 将列表排序。
* Loop(self)
* 循环输出列表中的所有值，如果列表为空则print报错。

1. 设计一个Point类，用顺序表实现一个Points列表，实现对Points中的元素进行管理的基本功能，包括：指定位置增加点，删除指定点，修改指定点的值，遍历和打印输出等操作。

使用Python语言完成。

Insert(self,Input,index=-1)

在index位置处插入值，若没有输入index则默认在列表末尾插入。若index超出范围则print报错。

Delete(self,index)

删除index位置处的值，若index超出范围则print报错。

Modify(self,Input,index)

修改index位置处的值为Input，若index超出范围则print报错返回报错。

Sort(self)

将列表排序。

Loop(self)

循环输出列表中的所有值，如果列表为空则print报错。

### week 3

**链式表的结构在LQuene.cs内**

**循环表的结构和点类型的定义在PolygonPoints.cs内**

1.用链接实现多边形顶点编辑的算法。

将列表排序。

public void Modify(Point data, int index)

修改index位置的内容为新的Point，如果索引超出范围则Console.WriteLine报错。

public Point Find(Point data)

寻找值为data的点并且返回点，如果点集合为空，或者该点不在点集合内，则Console.WriteLine报错。

public void Modify(Point data, int index)

修改index位置的内容为新的Point，如果索引超出范围则Console.WriteLine报错。

public void Loop()

循环输出列表中的所有值，如果列表为空则Console.WriteLine报错。

public Point Get(int index)

得到第index个元素的点坐标，如果列表为空，或索引超出范围则Console.WriteLine报错。

public bool IsEmpty()

判断循环表里是否有元素，返回boolean值。

public void Insert(Point data, int index)

插入data在第index个位置上，如果超出范围会Console.WriteLine报错。

public void Delete(int index)

删除第index处的节点，如果超出范围会Console.WriteLine报错。

2、计算多边形（折线）的周长和面积。

public double GetArea()

判断多边形是否成立（多边形包括三个及以上的点），若存在则计算多边形面积并且返回数值，若不存在则会Console.WriteLine报错并且返回0。

public double GetPer()

判断多边形是否成立（多边形包括三个及以上的点），若存在则计算多边形周长并且返回数值，若不存在则会Console.WriteLine报错并且返回0。