

Dictionary

一、Dictionary的使用

1.赋值方法

与HashTable类似，但使用前要先声明key和value的类型

```
Dictionary<int,string> dic = new Dictionary<int, string>();
//已声明key是int型, value是string型。
//1.Add()添加法      不能添加同名键
dic.Add(1,"张三");
dic.Add(2,"李四");
dic.Add(3,"王五");
dic.Add(0,"北京烤鸭");
dic.Add(5,"南京板鸭");
dic.Add(4,"上饶鸡腿");

//2.直接赋值法      添加同名键时, 新值会覆盖老值
dic[1] = "新来的";
dic[6] = "萍乡炒粉";
```

2.遍历方法

```
//1.与HashTable类似, 遍历Dictionary的key
foreach (var item in dic.Keys)
{
    System.Console.WriteLine(item + " : " + dic[item]);
}

//2.特有方法, 直接遍历Dictionary
/**
 * KeyValuePair<int,string> : 键值对, 尖括号内参数分别是key和value的类型
 * kv : 就是item改个名, 即集合中被遍历的每一项
 * dic : 被遍历的集合
 */
foreach(KeyValuePair<int,string> kv in dic)
{
    System.Console.WriteLine(kv.Key + " : " + kv.Value);
}
```

3.练习

```
//练习3
//统计Welcome to China中每个字符出现的次数 不考虑大小写
```

```
string str = "Welcome to China";
Dictionary<char,int> dic = new Dictionary<char, int>();
//思路, 将字符赋给key, 出现次数赋给value, 出现相同key则value +1
for(int i = 0;i < str.Length;i++){
    if(str[i] == ' '){        //跳过空格
        continue;
    }
    if(dic.ContainsKey(str[i])){    //如果已经存在该key,value +1
        dic[str[i]] ++;
    }else{        //如果不存在, 添加该key, value赋为1
        dic[str[i]] = 1;
    }
}
foreach (KeyValuePair<char,int> kv in dic)
{
    System.Console.WriteLine("字母" + kv.Key + "共出现了: " + kv.Value + "次");
}
//注: 没有判定大小写
```