# 对Tuple，NamedTuple和类的读写测试

## 两种类的定义：

|  |
| --- |
| *class* TestClass():  *def \_\_init\_\_*(self, *LATEST*:str,*BP1*:str,*SP1*:str):  self.LATEST = *LATEST* self.BP1 = *BP1* self.SP1 = *SP1*  *class* TestClass2:  \_\_slots\_\_ = ["LATEST", "BP1",'SP1'] |

写1000000次

|  |  |  |  |
| --- | --- | --- | --- |
| Tuple | TestClass2 | TestClass | 字典 |
| 3.5082058 | 5.0410135 | 8.3083346 | 4.426293600000001 |

读1000000次

|  |  |  |  |
| --- | --- | --- | --- |
| Tuple | TestClass2 | TestClass | 字典 |
| 1.1327162000000008 | 1.1523705 | 1.1107318 | 1.0181097000000001 |

由于NamedTuple的读写速度太慢，因此排除掉了它

# 程序代码

|  |
| --- |
| *import* time *from* collections *import* namedtuple *import* random   *class* TestClass:  *def \_\_init\_\_*(self, *LATEST*:str,*BP1*:str,*SP1*:str):  self.LATEST = *LATEST* self.BP1 = *BP1* self.SP1 = *SP1 class* TestClass2:  \_\_slots\_\_ = ["LATEST", "BP1",'SP1']  *def* main():  test\_num = 10000000   t1 = list()  t2 = list()  t3 = list()  t4 = list()  latest = random.sample('zyxwvutsrqponmlkjihgfedcba', 5)  sp1 = random.sample('zyxwvutsrqponmlkjihgfedcba', 5)  bp1 = random.sample('zyxwvutsrqponmlkjihgfedcba', 5)  # 写数据  start\_time = time.perf\_counter()  *for* i *in* range(test\_num):  t1.append((latest,sp1,bp1))  end\_time = time.perf\_counter()  print("元组的写时间:", end\_time - start\_time)   N = namedtuple('Tuple', 'LATEST BP1 SP1')  start\_time = time.perf\_counter()  *for* i *in* range(test\_num):  t = TestClass2()  t.LATEST = latest  t.SP1 = sp1  t.BP1 =bp1  t2.append(t)  end\_time = time.perf\_counter()  print("TestClass2的写时间:", end\_time - start\_time)   start\_time = time.perf\_counter()  *for* i *in* range(test\_num):  t3.append(TestClass(latest,bp1,sp1))  end\_time = time.perf\_counter()  print("TestClass的写时间:", end\_time - start\_time)   start\_time = time.perf\_counter()  *for* i *in* range(test\_num):  t4.append({"LATEST":latest,"SP1":sp1, "BP1":bp1})  end\_time = time.perf\_counter()  print("字典的写时间:", end\_time - start\_time)   #读数据  start\_time = time.perf\_counter()  *for* i *in* range(len(t1)):  a = t1[i][0]  b = t1[i][1]  c = t1[i][2]  end\_time = time.perf\_counter()  print("元组的读时间",end\_time - start\_time)   start\_time = time.perf\_counter()  *for* i *in* range(len(t2)):  a = t2[i].LATEST  b = t2[i].BP1  c = t2[i].SP1  end\_time = time.perf\_counter()  print("TestClass2的读时间", end\_time - start\_time)   start\_time = time.perf\_counter()  *for* i *in* range(len(t3)):  a = t3[i].LATEST  b = t3[i].BP1  c = t3[i].SP1  end\_time = time.perf\_counter()  print("TestClass的读时间", end\_time - start\_time)   start\_time = time.perf\_counter()  *for* i *in* range(len(t4)):  a = t4[i]['LATEST']  b = t4[i]['BP1']  c = t4[i]['SP1']  end\_time = time.perf\_counter()  print("字典的读时间",end\_time-start\_time)  *if* \_\_name\_\_ == '\_\_main\_\_':  main() |