**法一：**

import unittest  
class Student(object):  
 def \_\_init\_\_(self,number,name,grade):  
 self.number=number  
 self.name=name  
 self.grade=grade  
 def get\_number(self):  
 if self.number not in list:  
 return 'A'  
 return 'B'  
 def get\_name(self):  
 for i in range(0,len(self.name)):  
 t=self.name[i]  
 if not(((t>='a')and(t<='z')) or ((t>='A')and(t<='Z'))):  
 return 'B'  
 return 'A'  
 def get\_grade(self):  
 if (self.grade<0 or self.grade>100):  
 return 'B'  
 return 'A'  
class TestStudent(unittest.TestCase):  
 global list  
 list=[]  
 def test\_pd(self):  
 for i in range(1,11):  
 print("请输入第{}位同学的相关信息:".format(1))  
 number = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s= Student(number, name, grade)  
 self.assertEqual(s.get\_name(), 'A')  
 self.assertEqual(s.get\_grade(), 'A')  
 self.assertEqual(s.get\_number(), 'A')  
 list.append(number)  
  
 # print("请输入第{}位同学的相关信息:".format(1))  
 # number1 = eval(input(" 该位同学的学号:"))  
 # name = input(" 该位同学的姓名(拼音):")  
 # grade = eval(input(" 该位同学的成绩:"))  
 # s1= Student(number1, name, grade)  
 #  
 # print("请输入第{}位同学的相关信息:".format(2))  
 # number2 = eval(input(" 该位同学的学号:"))  
 # name = input(" 该位同学的姓名(拼音):")  
 # grade = eval(input(" 该位同学的成绩:"))  
 # s2= Student(number2, name, grade)  
 #  
 # print("请输入第{}位同学的相关信息:".format(3))  
 # number3 = eval(input(" 该位同学的学号:"))  
 # name = input(" 该位同学的姓名(拼音):")  
 # grade = eval(input(" 该位同学的成绩:"))  
 # s3= Student(number3, name, grade)  
 #  
 # print("请输入第{}位同学的相关信息:".format(4))  
 # number4 = eval(input(" 该位同学的学号:"))  
 # name = input(" 该位同学的姓名(拼音):")  
 # grade = eval(input(" 该位同学的成绩:"))  
 # s4= Student(number4, name, grade)  
 #  
 # print("请输入第{}位同学的相关信息:".format(5))  
 # number5 = eval(input(" 该位同学的学号:"))  
 # name = input(" 该位同学的姓名(拼音):")  
 # grade = eval(input(" 该位同学的成绩:"))  
 # s5= Student(number5, name, grade)  
 #  
 # print("请输入第{}位同学的相关信息:".format(6))  
 # number6 = eval(input(" 该位同学的学号:"))  
 # name = input(" 该位同学的姓名(拼音):")  
 # grade = eval(input(" 该位同学的成绩:"))  
 # s6= Student(number6, name, grade)  
 #  
 # print("请输入第{}位同学的相关信息:".format(7))  
 # number7 = eval(input(" 该位同学的学号:"))  
 # name = input(" 该位同学的姓名(拼音):")  
 # grade = eval(input(" 该位同学的成绩:"))  
 # s7= Student(number7, name, grade)  
 #  
 # print("请输入第{}位同学的相关信息:".format(8))  
 # number8 = eval(input(" 该位同学的学号:"))  
 # name = input(" 该位同学的姓名(拼音):")  
 # grade = eval(input(" 该位同学的成绩:"))  
 # s8= Student(number8, name, grade)  
 #  
 # print("请输入第{}位同学的相关信息:".format(9))  
 # number9 = eval(input(" 该位同学的学号:"))  
 # name = input(" 该位同学的姓名(拼音):")  
 # grade = eval(input(" 该位同学的成绩:"))  
 # s9= Student(number9, name, grade)  
 #  
 # print("请输入第{}位同学的相关信息:".format(10))  
 # number10 = eval(input(" 该位同学的学号:"))  
 # name = input(" 该位同学的姓名(拼音):")  
 # grade = eval(input(" 该位同学的成绩:"))  
 # s10= Student(number10, name, grade)  
 #  
 #  
 # self.assertEqual(s1.get\_name(), 'A')  
 # self.assertEqual(s1.get\_grade(),'A')  
 # self.assertEqual(s1.get\_number(),'A')  
 # list.append(number1)  
 #  
 # self.assertEqual(s2.get\_name(), 'A')  
 # self.assertEqual(s2.get\_grade(), 'A')  
 # self.assertEqual(s2.get\_number(), 'A')  
 # list.append(number2)  
 #  
 # self.assertEqual(s3.get\_name(), 'A')  
 # self.assertEqual(s3.get\_grade(), 'A')  
 # self.assertEqual(s3.get\_number(), 'A')  
 # list.append(number3)  
 #  
 # self.assertEqual(s4.get\_name(), 'A')  
 # self.assertEqual(s4.get\_grade(), 'A')  
 # self.assertEqual(s4.get\_number(), 'A')  
 # list.append(number4)  
 #  
 # self.assertEqual(s5.get\_name(), 'A')  
 # self.assertEqual(s5.get\_grade(), 'A')  
 # self.assertEqual(s5.get\_number(), 'A')  
 # list.append(number5)  
 #  
 # self.assertEqual(s6.get\_name(), 'A')  
 # self.assertEqual(s6.get\_grade(), 'A')  
 # self.assertEqual(s6.get\_number(), 'A')  
 # list.append(number6)  
 #  
 # self.assertEqual(s7.get\_name(), 'A')  
 # self.assertEqual(s7.get\_grade(), 'A')  
 # self.assertEqual(s7.get\_number(), 'A')  
 # list.append(number7)  
 #  
 # self.assertEqual(s8.get\_name(), 'A')  
 # self.assertEqual(s8.get\_grade(), 'A')  
 # self.assertEqual(s8.get\_number(), 'A')  
 # list.append(number8)  
 #  
 # self.assertEqual(s9.get\_name(), 'A')  
 # self.assertEqual(s9.get\_grade(), 'A')  
 # self.assertEqual(s9.get\_number(), 'A')  
 # list.append(number9)  
 #  
 # self.assertEqual(s10.get\_name(), 'A')  
 # self.assertEqual(s10.get\_grade(), 'A')  
 # self.assertEqual(s10.get\_number(), 'A')  
 # list.append(number10)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

**法二：**

import unittest  
class Student(object):  
 def \_\_init\_\_(self,number,name,grade):  
 self.number=number  
 self.name=name  
 self.grade=grade  
 def get\_number(self):  
 if self.number not in list:  
 return 'A'  
 return 'B'  
 def get\_name(self):  
 for i in range(0,len(self.name)):  
 t=self.name[i]  
 if not(((t>='a')and(t<='z')) or ((t>='A')and(t<='Z'))):  
 return 'B'  
 return 'A'  
 def get\_grade(self):  
 if (self.grade<0 or self.grade>100):  
 return 'B'  
 return 'A'  
class TestStudent(unittest.TestCase):  
 global list  
 list=[]  
 def test\_pd0(self):  
 print("请输入第{}位同学的相关信息:".format(1))  
 number1 = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s1 = Student(number1, name, grade)  
 self.assertEqual(s1.get\_name(), 'A')  
 self.assertEqual(s1.get\_grade(),'A')  
 self.assertEqual(s1.get\_number(),'A')  
 list.append(number1)  
  
 def test\_pd1(self):  
 print("请输入第{}位同学的相关信息:".format(2))  
 number2 = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s2 = Student(number2, name, grade)  
 self.assertEqual(s2.get\_name(), 'A')  
 self.assertEqual(s2.get\_grade(), 'A')  
 self.assertEqual(s2.get\_number(), 'A')  
 list.append(number2)  
  
 def test\_pd2(self):  
 print("请输入第{}位同学的相关信息:".format(3))  
 number3 = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s3 = Student(number3, name, grade)  
 self.assertEqual(s3.get\_name(), 'A')  
 self.assertEqual(s3.get\_grade(), 'A')  
 self.assertEqual(s3.get\_number(), 'A')  
 list.append(number3)  
  
 def test\_pd3(self):  
 print("请输入第{}位同学的相关信息:".format(4))  
 number4 = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s4 = Student(number4, name, grade)  
 self.assertEqual(s4.get\_name(), 'A')  
 self.assertEqual(s4.get\_grade(), 'A')  
 self.assertEqual(s4.get\_number(), 'A')  
 list.append(number4)  
  
 def test\_pd4(self):  
 print("请输入第{}位同学的相关信息:".format(5))  
 number5 = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s5 = Student(number5, name, grade)  
 self.assertEqual(s5.get\_name(), 'A')  
 self.assertEqual(s5.get\_grade(), 'A')  
 self.assertEqual(s5.get\_number(), 'A')  
 list.append(number5)  
  
 def test\_pd5(self):  
 print("请输入第{}位同学的相关信息:".format(6))  
 number6 = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s6 = Student(number6, name, grade)  
 self.assertEqual(s6.get\_name(), 'A')  
 self.assertEqual(s6.get\_grade(), 'A')  
 self.assertEqual(s6.get\_number(), 'A')  
 list.append(number6)  
  
 def test\_pd6(self):  
 print("请输入第{}位同学的相关信息:".format(7))  
 number7 = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s7 = Student(number7, name, grade)  
 self.assertEqual(s7.get\_name(), 'A')  
 self.assertEqual(s7.get\_grade(), 'A')  
 self.assertEqual(s7.get\_number(), 'A')  
 list.append(number7)  
  
 def test\_pd7(self):  
 print("请输入第{}位同学的相关信息:".format(8))  
 number8 = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s8 = Student(number8, name, grade)  
 self.assertEqual(s8.get\_name(), 'A')  
 self.assertEqual(s8.get\_grade(), 'A')  
 self.assertEqual(s8.get\_number(), 'A')  
 list.append(number8)  
  
 def test\_pd8(self):  
 print("请输入第{}位同学的相关信息:".format(9))  
 number9 = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s9 = Student(number9, name, grade)  
 self.assertEqual(s9.get\_name(), 'A')  
 self.assertEqual(s9.get\_grade(), 'A')  
 self.assertEqual(s9.get\_number(), 'A')  
 list.append(number9)  
  
 def test\_pd9(self):  
 print("请输入第{}位同学的相关信息:".format(10))  
 number10 = eval(input(" 该位同学的学号:"))  
 name = input(" 该位同学的姓名(拼音):")  
 grade = eval(input(" 该位同学的成绩:"))  
 s10 = Student(number10, name, grade)  
 self.assertEqual(s10.get\_name(), 'A')  
 self.assertEqual(s10.get\_grade(), 'A')  
 self.assertEqual(s10.get\_number(), 'A')  
 list.append(number10)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()