

任务分析

我们需要实现一个宿舍管理系统，支持插入、查询、展示全部信息

学生的姓名、性别、宿舍房间号、联系电话都有可能重复，但是学号是唯一的

所以学号可以作为主键，再根据程序要求里的提示，我最终选择了字典套字典的存储结构

stu 是一个全局字典，键为学生学号，值是另一个字典（包含学生的详细信息）

内部的字典结构为：{'name':name,'gender':gender,'room':room,'phone':phone}

为了保证程序持续运行，写一个死循环，每次循环让用户选择3中操作的一个

但是用户还要有办法退出程序，所以再加一个退出系统的选项

为了增加用户体验，操作后需要有暂停，而不是直接返回菜单

循环体结尾加入 input("按回车键继续.....") 即可

代码详解

1. 主函数

```
while True:
    menu()
    choice=input("请选择操作(1-4): ").strip()
    if choice=='1':
        query()
    elif choice=='2':
        insert()
    elif choice=='3':
        show_all()
    elif choice=='4':
        print("感谢使用宿舍管理系统，再见！")
        break
    else:
        print("错误：请输入1-4之间的数字！")
    input("按回车键继续.....")
```

写一个死循环，从而让程序持续不断地运行，每次循环让用户选择3中操作的一个

但是用户还要有办法退出程序，所以再加一个退出系统的选项4

每次让用户先输入一个操作编号，再调用对应的函数

strip的作用是：移除用户输入字符串开头和结尾的空白字符，用户可能不小心在输入前后加了空格

2. 输出菜单

```
def menu():
    print("-----宿舍管理系统-----")
    print("1. 按学号查找某一位学生的具体信息")
    print("2. 录入新的学生信息")
    print("3. 显示现有的所有学生信息")
    print("4. 退出系统")
    print("-----")
```

每次输出固定的内容，提示用户输入哪个数字对应哪个选项

第一行和最后一行是分割线，起到整齐和美观的效果

3. 插入操作

```
def insert():
    while True:
        id=input("请输入学号: ").strip()
        if not id:
            print("错误: 学号不能为空!")
            continue
        if not id.isdigit() or len(id) != 11:
            print("错误: 学号必须为11位数字!")
            continue
        if id in stu:
            print("错误: 该学号已存在!")
            continue
        break

    while True:
        name=input("请输入姓名: ").strip()
        if not name:
            print("错误: 姓名不能为空!")
            continue
        break

    while True:
        gender=input("请输入性别(男/女): ").strip()
        if gender not in ['男', '女']:
            print("错误: 性别只能是'男'或'女'!")
            continue
        break

    while True:
        room=input("请输入宿舍房间号: ").strip()
        if not room:
            print("错误: 宿舍房间号不能为空!")
            continue
        if not room.isdigit() or len(room) != 3:
            print("错误: 房间号必须为3位数字!")
            continue
        break

    while True:
        phone=input("请输入联系电话: ").strip()
        if not phone:
            print("错误: 联系电话不能为空!")
            continue
        if not phone.isdigit() or len(phone) != 11:
            print("错误: 电话必须为11位数字!")
            continue
        break

    confirm=input("确认添加?(y/n): ")
    if confirm.lower()=='y':
```

```

stu[id]={ 'name':name, 'gender':gender, 'room':room, 'phone':phone}
print("已录入新的学生信息！")
else:
    print("未录入新的学生信息！")

```

代码较长，我们先看输入学号的错误处理逻辑：

```

while True:
    id=input("请输入学号：").strip()
    if not id:
        print("错误：学号不能为空！")
        continue
    if not id.isdigit() or len(id) != 11:
        print("错误：学号必须为11位数字！")
        continue
    if id in stu:
        print("错误：该学号已存在！")
        continue
    break

```

上述代码中，使用input读入，strip 去除开头和结尾的空白字符

为什么外层使用while循环呢？

因为如果用户输入了不符合要求的学号，需要不断地重新输入

如果输入内容为空，报错“学号不能为空”，并continue，即重新执行循环体

如果输入的不是11位数字，报错“学号必须为11位数字”，并continue，即重新执行循环体

如果输入的学号已存在，报错“该学号已存在”，并continue，即重新执行循环体

其中，判断是否全为数字可以用isdigit函数

判断学号是否已经在stu中作为第一关键字出现过，可以用 id in stu

如果上述三个判断结果都为假，那么输入内容是符合要求的，break，此时id中存的就是符号要求的学号

姓名、性别、宿舍号、电话号的判断逻辑同理

最后来看如何添加到字典里：

```

confirm=input("确认添加?(y/n): ")
if confirm.lower()=='y':
    stu[id]={ 'name':name, 'gender':gender, 'room':room, 'phone':phone}
    print("已录入新的学生信息！")
else:
    print("未录入新的学生信息！")

```

用户可能在输入信息后不想添加这个学生信息了，因此要提示用户输入一个y/n，知道是否确认添加

用户可能不小心输入了大写的Y，confirm.lower()可以将Y和y统一转化成y

如果确认添加，利用{'name':name,'gender':gender,'room':room,'phone':phone}构造一个新的字典

并将这个新的字典作为第二关键字，与第一关键字（学号）组合，存入stu中

4. 查询操作

```
def query():
    id=input("请输入要查找的学生学号: ").strip()

    if not id:
        print("错误: 学号不能为空!")
        return

    if id in stu:
        info=stu[id]
        print("找到学生信息:")
        print(f"    学号: {id}")
        print(f"    姓名: {info['name']}")
        print(f"    性别: {info['gender']}")
        print(f"    宿舍房间号: {info['room']}")
        print(f"    联系电话: {info['phone']}")
    else:
        print(f"未找到学号为 {id} 的学生信息")
```

让用户输入一个字符串代表学号，并去除首位空白字符

如果输入内容为空，报错“学号不能为空”，并return

如果学号在stu中，新建一个临时变量用来存放**与id同属一个元素的第二关键字**（学生的详细信息）

先输出学号，再通过info['表项']的方式，依次输出学生的详细信息

如果学号不在stu中，报错“未找到学号为 {id} 的学生信息”，其中 {id} 会被替换为id实际的值

5. 展示全部信息

```
def show_all():
    if not stu:
        print("当前没有学生信息")
        return

    print("-----所有学生信息-----")
    for id,info in stu.items():
        print(f"学号: {id}")
        print(f"姓名: {info['name']}")
        print(f"性别: {info['gender']}")
        print(f"宿舍房间号: {info['room']}")
        print(f"联系电话: {info['phone']}")
    print("-----")
```

如果stu为空，意味着没有任何学生信息，输出“当前没有学生信息”，并return

否则，利用 id,info in stu.items() 的方式遍历字典中每个元素，第一关键字存入 id，第二关键字存入 info

先输出学号，再通过info['表项']的方式，依次输出学生的详细信息

第一行和最后一行是分割线，起到整齐和美观的效果