

# 简易学生管理系统

## 1. 代码段：

```
'''
简易学生管理系统
'''

import os
import datetime

class Birthday:
    '''出生日期类'''

    def __init__(self, year=0, month=0, day=0):
        self._year = year # 年
        self._month = month # 月
        self._day = day # 日

    def __str__(self):
        return ('%d-%d-%d') % (self._year, self._month, self._day)

    @property
    def year(self):
        return self._year

class InfoAddr:
    '''户籍信息类'''

    def __init__(self, name='', no='', birthday=Birthday(),
addr=''):
        self._name = name # 姓名
        self._no = no # 身份证号
        self._birthday = birthday # 出生年月日
        self._addr = addr # 地址

    def __str__(self):
        return ('姓名: %s; 身份证号: %s; 出生年月日: %s; 地址: %s') \
            % (self._name, self._no, self._birthday, self._addr)

    @property
    def name(self):
        return self._name

    @property
    def no(self):
        return self._no
```

```

@property
def birthday(self):
    return self._birthday

@property
def addr(self):
    return self._addr

class InfoStudent:
    '''学籍信息类'''

    def __init__(self, student='', no='', college='', profession='',
s_class=''):
        self._student = student # 学号
        self._no = no # 身份证号
        self._college = college # 学院
        self._profession = profession # 专业
        self._s_class = s_class # 班级

    def __str__(self):
        return ('学号: %s; 身份证号: %s; 学院: %s; 专业: %s; 班级: %s') \
            % (self._student, self._no, self._college,
self._profession, self._s_class)

@property
def student(self):
    return self._student

@property
def no(self):
    return self._no

@property
def college(self):
    return self._college

@property
def profession(self):
    return self._profession

@property
def s_class(self):
    return self._s_class

# def list_input():
#     '''
#     输入异常捕获
#     '''
#     while True:

```

```

#         try:
#             info_list = list(map(str, input('>> ').split(' ')))
#             return info_list
#         except:
#             print('输入格式有误, 重新输入')

def int_input():
    """
    输入整数异常捕获
    """
    while True:
        try:
            select_index = int(input('请输入选项: '))
            return select_index
        except:
            print('\n- !!! 请输入整数序号\n')

def add_info_addr(info_addrs):
    """
    添加户籍信息
    """
    print('\n添加户籍信息（格式: 姓名 身份证号 出生年月日（年 月 日） 住址）-- 使用空格隔开')
    info_list = list(map(str, input('>> ').split(' ')))
    birthday = Birthday(int(info_list[2]), int(
        info_list[3]), int(info_list[4]))
    info = InfoAddr(info_list[0], info_list[1], birthday,
info_list[5])
    info_addrs.append(info)
    print('\n- 添加成功: %s' % (info))

def add_info_student(info_students):
    """
    添加学籍信息
    """
    print('\n添加学籍信息（格式: 学号 身份证号 学院 专业 班级）-- 使用空格隔
开')
    info_list = list(map(str, input('>> ').split(' ')))
    info = InfoStudent(info_list[0], info_list[1],
        info_list[2], info_list[3], info_list[4])
    info_students.append(info)
    print('\n- 添加成功: %s' % (info))

def print_info_addrs(info_addrs):
    """
    打印户籍信息
    """
    tpl1t = '{:^10}\t{:^10}\t{:^15}\t{:^10}\t{:^10}'

```

```

print('\n户籍信息 >>>>>\n')
print(tplt.format('序号', '姓名', '身份证号', '出生年月', '住址'))
for index, info in enumerate(info_addrs):
    print(tplt.format(index + 1, info.name,
                      info.no, str(info.birthday), info.addr))

def print_info_students(info_students):
    """
    打印学籍信息
    """
    tplt = '{:^10}\t{:^10}\t{:^15}\t{:^15}\t{:^15}\t{:^15}'
    print('\n学籍信息 >>>>>\n')
    print(tplt.format('序号', '学号', '身份证号', '学院', '专业', '班级'))
    for index, info in enumerate(info_students):
        print(tplt.format(index + 1, info.student, info.no,
                          info.college, info.profession,
                          info.s_class))

def get_info_base(no, info_addrs):
    """
    根据学生身份证号获取户籍信息
    """
    for index, info in enumerate(info_addrs):
        if info.no == no:
            return index
    return -1

def print_info_bases(info_addrs, info_students):
    """
    打印基础信息
    """
    tplt = '{:^10}\t{:^10}\t{:^20}\t{:^20}\t{:^15}\t{:^15}'
    print('\n基础信息 >>>>>\n')
    print(tplt.format('序号', '学号', '姓名', '年龄', '学院', '班级'))
    for index, info_student in enumerate(info_students):
        addr_index = get_info_base(info_student.no, info_addrs)
        if addr_index is not -1:
            age = datetime.datetime.now().year - \
                info_addrs[addr_index].birthday.year
            print(tplt.format(index + 1, info_student.student,
                              info_addrs[addr_index].name,
                              age, info_student.profession,
                              info_student.s_class))
        else:
            print('- 户籍信息查询错误')
            # if info_addrs[addr_index].birthday.year == 0:
            #     age = 0
            # else:

```

```

        # age = datetime.datetime.now().year -
        info_addrs[addr_index].birthday.year
        # print(tplt.format(index + 1, info_student.student,
        info_addrs[addr_index].name,
        # age, info_student.profession,
        info_student.s_class))

def print_menu():
    '''
    打印菜单
    '''
    print('-'*50)
    print('简易学生管理系统'.center(25, chr(12288)))
    print('0.添加户籍信息')
    print('1.添加学籍信息')
    print('2.打印户籍信息')
    print('3.打印学籍信息')
    print('4.打印基础信息')
    print('5.退出')
    print('-'*50)

def select_menu(info_addrs, info_students):
    '''
    选择菜单
    '''
    # index = int(input('请输入选项: '))
    index = int_input()
    if index is 0:
        add_info_addr(info_addrs) # 添加户籍信息
        add_info_student(info_students) # 添加学籍信息
    elif index is 1:
        add_info_student(info_students) # 添加学籍信息
        add_info_addr(info_addrs) # 添加户籍信息
    elif index is 2:
        print_info_addrs(info_addrs) # 打印户籍信息
    elif index is 3:
        print_info_students(info_students) # 打印学籍信息
    elif index is 4:
        print_info_bases(info_addrs, info_students) # 打印基础信息
    elif index is 5:
        print('\n- GOODBYE\n')
        os._exit(0)
    else:
        print('\n- !!! 请输入正确的序号\n')
        select_menu(info_addrs, info_students)

def main():
    info_addrs = [] # 户籍信息
    info_students = [] # 学籍信息

```

```

while True:
    print_menu()
    select_menu(info_addr, info_students)

if __name__ == '__main__':
    main()

```

## 2. 运行结果

### 2.1 添加户籍信息 & 学籍信息

```

-----
          简易学生管理系统
0.添加户籍信息
1.添加学籍信息
2.打印户籍信息
3.打印学籍信息
4.打印基础信息
5.退出
-----
请输入选项：0

添加户籍信息（格式：姓名 身份证号 出生年月日（年 月 日） 住址）-- 使用空格隔开
>> 张三 620423199803020000 2010 9 9 北京市海淀区

- 添加成功：姓名：张三；身份证号：620423199803020000；出生年月日：2010-9-9；地址：北京市海淀区

添加学籍信息（格式：学号 身份证号 学院 专业 班级）-- 使用空格隔开
>> 1151002068 620423199803020000 信息工程学院 计算机科学与技术 2班

- 添加成功：学号：1151002068；身份证号：620423199803020000；学院：信息工程学院；专业：计算机科学与技术；班级：2班

```

### 2.2 打印户籍信息

```

-----
          简易学生管理系统
0.添加户籍信息
1.添加学籍信息
2.打印户籍信息
3.打印学籍信息
4.打印基础信息
5.退出
-----
请输入选项：2

户籍信息 >>>>>

      序号      姓名      身份证号      出生年月      住址
      1          张三      620423199803020000      2010-9-9      北京市海淀区

```

### 2.3 打印学籍信息

```

-----
          简易学生管理系统
0.添加户籍信息
1.添加学籍信息
2.打印户籍信息
3.打印学籍信息
4.打印基础信息
5.退出
-----
请输入选项：3

学籍信息 >>>>>

      序号      学号      身份证号      学院      专业      班级
      1          1151002068      620423199803020000      信息工程学院      计算机科学与技术      2班

```

### 2.4 打印基础信息

```
-----
                        简易学生管理系统
0.添加户籍信息
1.添加学籍信息
2.打印户籍信息
3.打印学籍信息
4.打印基础信息
5.退出
-----
请输入选项： 4

基础信息 >>>>>

      序号          学号          姓名          年龄          学院          班级
      1            1151002068      张三            9            计算机科学与技术      2班
```