SY-023

实 验 报 告

课程名称： Python程序设计 \_

系部名称： 计算机科学与技术 \_

专业班级： 计科22-3班 \_\_

学生姓名： 邹晨骏 \_\_

学 号： 20222163 \_

指导教师： 曲天伟 \_\_ \_

**黑龙江工程学院教务处制**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实验项目 | 实验三 Python组合数据类型综合实验 | | | | | 实验日期 | 2024.9.9 |
| 实验地点 | A607 | 同组人数 | 1 | 分工 |  | 实验台号 |  |
| 实验类型 | □验证性 □ 综合性 □ 设计性 □ 其 他 | | | | | | |
| 一、实验目的  练习Python组合数据类型在解决实际数据存储与管理问题时的应用，会分析具体的程序代码，及代码实现中根据具体应用选择什么组合数据类型进行数据存储。并会根据分析结果实现一个新的相似功能的程序数据存储和程序功能的实现。 | | | | | | | |
| 二、实验仪器设备实验工具软件  笔记本电脑，python3.8.2解释器，pycharm2021编辑器 | | | | | | | |
| 三、实验要求  （1）请阅读下面代码，这是一个手机通讯论的代码，阅读后分析这段程序中使用了哪些组合数据类型，分别用来存储什么。  （2）再将这个程序改写成一个学生成绩系统。成绩基本信息包括学生学号，学生姓名，课程名称，成绩。（课程名称为python，成绩满分为100分）  要求实现的功能有：  1.添加学生成绩信息。  2.查看学生成绩信息。  3.删除学生成绩信息。  4.修改学生成绩信息。  5.查找学生成绩信息。  6.退出。 | | | | | | | |
| 四、实验步骤  student\_info = [] # 用于存储学生成绩的列表  while True:  print("=" \* 20)  print('欢迎进入学生成绩管理系统')  print("1.添加学生成绩")  print("2.查看学生成绩")  print("3.删除学生成绩")  print("4.修改学生成绩")  print("5.查找学生成绩")  print("6.退出系统")  print("=" \* 20)    # 定义用于存储学生数据的字典  student\_data = {}    # 选择操作功能  action = input('请输入功能编号:')  if action == '1': # 添加学生成绩  name = input('请输入学生姓名：')  course = input('请输入课程名称：')  score = input('请输入学生成绩：')  # 检查输入是否为空  if not name.strip() or not course.strip() or not score.strip():  print('请完整输入所有信息！')  else: # 保存信息到字典并添加到列表  student\_data['姓名'] = name  student\_data['课程'] = course  student\_data['成绩'] = score  student\_info.append(student\_data)  print('学生成绩已成功保存！')  elif action == '2': # 查看学生成绩  if not student\_info:  print('当前没有学生成绩记录')  else:  for student in student\_info:  for key, value in student.items():  print(f'{key}: {value}')  elif action == '3': # 删除学生成绩  if student\_info:  del\_name = input('请输入要删除的学生姓名：')  for student in student\_info:  if del\_name == student.get('姓名'):  student\_info.remove(student)  print(f'学生 {del\_name} 的信息已删除')  break  else:  print(f'没有找到姓名为 {del\_name} 的学生')  else:  print('当前没有学生成绩记录')  elif action == '4': # 修改学生成绩  if student\_info:  modify\_name = input('请输入要修改的学生姓名：')  for student in student\_info:  if modify\_name == student.get('姓名'):  # 获取学生信息的索引  idx = student\_info.index(student)  current\_student = student\_info[idx]    # 显示当前信息  for key, value in current\_student.items():  print(f'{key}: {value}')  # 输入新的信息，可以留空以保留原信息  new\_name = input('请输入新的姓名（留空表示不修改）：') or current\_student['姓名']  new\_course = input('请输入新的课程名称（留空表示不修改）：') or current\_student['课程']  new\_score = input('请输入新的成绩（留空表示不修改）：') or current\_student['成绩']  # 更新信息  current\_student.update({'姓名': new\_name, '课程': new\_course, '成绩': new\_score})  print(f'学生 {modify\_name} 的信息已更新')  break  else:  print(f'没有找到姓名为 {modify\_name} 的学生')  else:  print('当前没有学生成绩记录')  elif action == '5': # 查找学生成绩  if student\_info:  search\_name = input('请输入要查找的学生姓名：')  for student in student\_info:  if search\_name == student.get('姓名'):  for key, value in student.items():  print(f'{key}: {value}')  break  else:  print(f'没有找到姓名为 {search\_name} 的学生')  else:  print('当前没有学生成绩记录')  elif action == '6': # 退出系统  print('已退出系统，再见！')  break  else:  print('无效的功能编号，请重新输入') | | | | | | | |
| 五．实验结果  功能1    功能2    功能3    功能4    功能5    功能6 | | | | | | | |
| 六、实验中遇到的问题及解决方法  查找时会便利整个字典，在if时精确到str=item.name即可  返回之后没有目录很不方便，把表单放到while里，每次都可以出现  对每个输入做限制使其更加合理 | | | | | | | |