**四川大学实验课程期末报告**

**（2023——2024学年第2学期）**

课程号： 课序号： 课程名称： 任课教师： 成绩：

适用专业年级： 学生人数： 学号：2022141461047 姓名：郑忆

|  |
| --- |
| 请完成以下3题，给出编程思路、程序代码以及运行结果。   1. 在health.txt文件中包含健康指标如下，编写health.py模块，能够读取文件内容，并根据人工输入的体重和身高进行计算BMI，测算健康指数。当输入身高为0时，能够进行异常处理。   health.txt  <18.5: "过轻"  <24: "正常"  <27: "过重"  <32: "肥胖"  >32: "非常肥胖"  **编程思路：**   1. 读取health.txt文件并解析内容，得到各健康等级的BMI范围。 2. 编写一个函数，接收用户输入的体重和身高，并计算BMI。 3. 根据计算的BMI值判断用户的健康指数，并处理异常情况（如身高为0时）。   **程序代码：**  # health.py  def read\_health\_data(file\_path):  health\_data = {}  with open(file\_path, 'r', encoding='utf-8') as file:  for line in file:  line = line.strip()  if line:  parts = line.split(':')  bmi\_range = parts[0].strip('<> ')  description = parts[1].strip().strip('"')  if '<' in parts[0]:  limit = float(parts[0].strip('<'))  health\_data[('lt', limit)] = description  elif '>' in parts[0]:  limit = float(parts[0].strip('>'))  health\_data[('gt', limit)] = description  return health\_data  def calculate\_bmi(weight, height):  if height == 0:  raise ValueError("Height cannot be zero.")  return weight / (height \*\* 2)  def determine\_health\_index(bmi, health\_data):  for (op, limit), description in health\_data.items():  if op == 'lt' and bmi < limit:  return description  elif op == 'gt' and bmi > limit:  return description  return "未知"  if \_\_name\_\_ == "\_\_main\_\_":  health\_data = read\_health\_data('health.txt')  try:  weight = float(input("请输入体重(kg): "))  height = float(input("请输入身高(m): "))  bmi = calculate\_bmi(weight, height)  health\_index = determine\_health\_index(bmi, health\_data)  print(f"BMI: {bmi:.2f}, 健康指数: {health\_index}")  except ValueError as e:  print(f"输入错误: {e}")  **运行结果：**           1. 随机生成二维数据，并以散点图的形式进行展示，模拟随机游走的过程。   **编程思路：**   1. 生成一组随机二维数据，模拟随机游走过程。 2. 使用matplotlib绘制散点图展示随机游走的轨迹。   **程序代码：**  import matplotlib.pyplot as plt  import numpy as np  def random\_walk(n\_points=1000):  x = np.cumsum(np.random.randn(n\_points))  y = np.cumsum(np.random.randn(n\_points))  return x, y  if \_\_name\_\_ == "\_\_main\_\_":  x, y = random\_walk()  plt.figure(figsize=(10, 6))  plt.scatter(x, y, c=range(len(x)), cmap='viridis', s=10)  plt.colorbar(label='Step')  plt.title('Random Walk Simulation')  plt.xlabel('X')  plt.ylabel('Y')  plt.show()  **运行结果：** 程序会生成一个随机游走的散点图，颜色表示步数。     1. 设计并实现一个简单的图书管理系统，该系统具有以下功能： 2. 添加书籍信息（书名、作者、ISBN、数量）。 3. 查看所有书籍信息。 4. 搜索书籍（根据书名或作者）。 5. 借书和还书功能（根据ISBN号）。   具体要求如下：   1. 使用模块化编程思想，将不同功能分成多个模块。 2. 主程序调用各个模块实现上述功能。 3. 每个模块应具有清晰的接口和封装。   程序设计说明：   1. Book类：定义书籍的基本属性和方法。 2. Library模块：管理图书的添加、查看和搜索。 3. BorrowReturn模块：实现借书和还书功能。 4. Main模块：提供用户交互界面，调用其他模块完成具体功能。   **编程思路：**   1. 定义Book类，包含书籍的基本属性和方法。 2. 编写Library模块，实现书籍的添加、查看和搜索功能。 3. 编写BorrowReturn模块，实现借书和还书功能。 4. 编写Main模块，提供用户交互界面，调用其他模块实现具体功能。   **程序代码**：  # book.py  class Book:  def \_\_init\_\_(self, title, author, isbn, quantity):  self.title = title  self.author = author  self.isbn = isbn  self.quantity = quantity  def \_\_str\_\_(self):  return f"{self.title} by {self.author}, ISBN: {self.isbn}, Quantity: {self.quantity}"  # library.py  from book import Book  class Library:  def \_\_init\_\_(self):  self.books = []  def add\_book(self, title, author, isbn, quantity):  new\_book = Book(title, author, isbn, quantity)  self.books.append(new\_book)  def view\_books(self):  return self.books  def search\_books(self, keyword):  return [book for book in self.books if keyword.lower() in book.title.lower() or keyword.lower() in book.author.lower()]  # borrow\_return.py  from book import Book  class BorrowReturn:  def \_\_init\_\_(self, library):  self.library = library  self.borrowed\_books = {}  def borrow\_book(self, isbn):  for book in self.library.books:  if book.isbn == isbn and book.quantity > 0:  book.quantity -= 1  self.borrowed\_books[isbn] = self.borrowed\_books.get(isbn, 0) + 1  return f"Borrowed {book.title}"  return "Book not available"  def return\_book(self, isbn):  if isbn in self.borrowed\_books and self.borrowed\_books[isbn] > 0:  self.borrowed\_books[isbn] -= 1  for book in self.library.books:  if book.isbn == isbn:  book.quantity += 1  return f"Returned {book.title}"  return "Book not borrowed"  from library import Library  from borrow\_return import BorrowReturn  def main():  library = Library()  br\_system = BorrowReturn(library)  while True:  print("1. 添加书籍")  print("2. 查看所有书籍")  print("3. 搜索书籍")  print("4. 借书")  print("5. 还书")  print("6. 退出")  choice = input("请选择操作: ")  if choice == '1':  title = input("书名: ")  author = input("作者: ")  isbn = input("ISBN: ")  quantity = int(input("数量: "))  library.add\_book(title, author, isbn, quantity)  elif choice == '2':  for book in library.view\_books():  print(book)  elif choice == '3':  keyword = input("请输入书名或作者: ")  results = library.search\_books(keyword)  for book in results:  print(book)  elif choice == '4':  isbn = input("请输入ISBN: ")  print(br\_system.borrow\_book(isbn))  elif choice == '5':  isbn = input("请输入ISBN: ")  print(br\_system.return\_book(isbn))  elif choice == '6':  break  else:  print("无效选择，请重新输入")  if \_\_name\_\_ == "\_\_main\_\_":  main()  **运行结果：** 程序提供用户界面，用户可以通过输入选择不同操作，如添加书籍、查看书籍、搜索书籍、借书和还书等。 |