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 | 实验五 文件操作编程练习 | | | | | 实验日期 | 2024.10.12 |
| 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  掌握文件的打开，读、写、关闭操作，目录的操作及文件拷贝操作。能将录入的数据信息存储在外部文本文件中，并能把外部文本文件中的内容读入到程序中进行处理。 | | | | | | | |
| 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）设计一个Circle（圆）类，该类中包括属性radius（半径），还包括\_\_init\_\_()、get\_perimeter()（求周长）和get\_area()（求面积）共三个方法。设计完成后，创建Circle类的对象求圆的周长和面积。  （2）设计一个Course（课程）类，该类中包括number（编号）、name（名称）、teacher（任课教师）、location（上课地点）共4个属性，其中location是私有属性；还包括\_\_init\_\_()、show\_info()（显示课程信息）共两个方法。设计完成后，创建Course类的对象显示课程的信息。  2. 猜拳游戏一般包含三种手势：石头、剪刀、布，判定规则为石头胜剪刀，剪刀胜布，布胜石头。要求编写代码，实现基于面向对象思想的人机猜拳游戏。运行结果如下。    分析如下：此问题中有三个类，一是人工游戏玩家Player，一个是计算机游戏玩家AIPlayer，还有一个游戏结果判断类Game。Player类里面应有一个字典变量属性用来存储游戏的三种手势0剪刀、1石头、2布，还应该有一个出手势的方法gesture，这个方法的实现是通过人工输入0，1，2表示分别出了对应的手势。AIPlayer应该继承于Player类，同时应该有一个计算机出手势的方法Aigesture，计算机出手势，不都是随机的，只有第一次是随机出手势，随后会根据人工出手势的概率判断计算机出什么手势获胜概率最大。所以AIPlayer类还应该有一个存储人工已经出过手势的数据列表属性play\_data。Game类用来实现游戏结果的判断，应该有game\_judge()方法来实现，同时还应该有游戏开始的方法实现调用游戏结果判断方法game\_judge()，同时进行下一次游戏循环的提示。具体的类图及关系如下图所示。    请编程实现此游戏功能。  提示：可能用到一个列表中出现最多的元素的判断方法，可以使用max函数具体参看下面的代码。 | | | | | | | |
| 四、实验步骤  1.  import math  *# Circle 类的定义* class Circle:  def \_\_init\_\_(self, radius):  self.radius = radius   def get\_perimeter(self):  return 2 \* math.pi \* self.radius   def get\_area(self):  return math.pi \* (self.radius \*\* 2)  *# Course 类的定义* class Course:  def \_\_init\_\_(self, number, name, teacher, location):  self.number = number  self.name = name  self.teacher = teacher  self.\_\_location = location *# 私有属性* def show\_info(self):  print(f"课程编号: {self.number}")  print(f"课程名称: {self.name}")  print(f"任课教师: {self.teacher}")  print(f"上课地点: {self.\_\_location}")  *# 创建 Circle 对象并输出圆的周长和面积* circle = Circle(5) *# 半径为5* print("圆的周长:", circle.get\_perimeter()) print("圆的面积:", circle.get\_area()) print()  *# 创建 Course 对象并显示课程信息* course = Course("CS101", "Python程序设计", "李老师", "101教室") course.show\_info()  2.  import random  *# 人类玩家类* class Player:  def \_\_init\_\_(self):  *# 字典存储三种手势* self.dict = {0: "剪刀", 1: "石头", 2: "布"}   def gesture(self):  *# 人工输入手势* while True:  try:  player\_choice = int(input("请输入(0剪刀，1石头，2布):"))  if player\_choice in [0, 1, 2]:  return player\_choice  else:  print("输入错误，请输入0，1或2。")  except ValueError:  print("输入无效，请输入数字0，1或2。")  *# 计算机玩家类，继承自Player类* class AIPlayer(Player):  def \_\_init\_\_(self):  super().\_\_init\_\_()  *# 存储人工已经出过的手势数据* self.play\_data = []   def ai\_gesture(self):  *# 如果没有数据，随机选择手势* if not self.play\_data:  return random.randint(0, 2)  else:  *# 计算人类玩家最常出的手势* most\_common\_gesture = max(self.play\_data, key=self.play\_data.count)  *# 根据人类玩家最常出的手势，选择可以克制的手势* if most\_common\_gesture == 0: *# 人类最常出剪刀* return 1 *# 电脑出石头* elif most\_common\_gesture == 1: *# 人类最常出石头* return 2 *# 电脑出布* elif most\_common\_gesture == 2: *# 人类最常出布* return 0 *# 电脑出剪刀  # 游戏类，控制游戏流程* class Game:  def \_\_init\_\_(self):  self.player = Player()  self.ai\_player = AIPlayer()   *# 游戏结果判断* def game\_judge(self, player\_choice, ai\_choice):  if player\_choice == ai\_choice:  return "平局"  elif (player\_choice == 0 and ai\_choice == 2) or \  (player\_choice == 1 and ai\_choice == 0) or \  (player\_choice == 2 and ai\_choice == 1):  return "恭喜，你赢了！"  else:  return "你输了！"   *# 游戏开始* def game\_start(self):  while True:  *# 人类玩家出手* player\_choice = self.player.gesture()  *# 记录人类玩家出过的手势* self.ai\_player.play\_data.append(player\_choice)   *# 计算机玩家出手* ai\_choice = self.ai\_player.ai\_gesture()   *# 输出双方手势* print(f"你出的手势是: {self.player.dict[player\_choice]}")  print(f"电脑出的手势是: {self.player.dict[ai\_choice]}")   *# 判断结果* result = self.game\_judge(player\_choice, ai\_choice)  print(result)   *# 询问是否继续游戏* continue\_game = input("是否继续(y/n): ").lower()  if continue\_game != 'y':  break  *# 开始游戏* if \_\_name\_\_ == "\_\_main\_\_":  game = Game()  game.game\_start() | | | | | | | |
| 五．实验结果  1.    2. | | | | | | | |
| 六、实验中遇到的问题及解决方法 | | | | | | | |