**实 验 报 告**

**（与程序设计有关）**

**课程名称： python程序设计**

**实验题目：**  类的定义和使用

**班级学号：** 2203050320

**姓 名:** 闻家尉

**成 绩：**

**沈 阳 理 工 大 学**

**2023年 10 月 17 日**

|  |
| --- |
| **实验目的及要求：**熟练Python中面向对象的编程思想，能设计简单的类，封装所有数据和方法 |
| **软硬件环境：**   1. 硬件环境：PC机一台   2、软件环境：Windows 7/8/10、PyCharm 2023.1.3 |
| **算法或原理分析（实验内容）：**  1设计一个Circle（圆）类，包括圆心位置、半径、颜色等属性。编写构造方法和其他方法，计算周长和面积。请编写程序验证类的功能。  2设计一个课程类，包括课程编号、课程名称、任课教师、上课地点等属性，把上课地点变量设为私有的，增加构造方法和显示课程信息的方法。  3 设计一个表示学生的（Student）类，该类的属性有姓名（name）、年龄（age）、成绩（scores）（成绩包含语文、数学和英语三科成绩，每科成绩的类型为整数），另外有3个方法。获取学生姓名的方法：get\_name( )，返回类型为String。获取学生年龄的方法：get\_age( )方法，返回类型为int。返回3门科目中最高的分数：get\_course( )，返回类型为int。（成绩可以使用字典。） |

|  |  |  |  |
| --- | --- | --- | --- |
| **程序代码或实现过程：**  **题目一**  import math  class Circle:  def \_\_init\_\_(self, x, y, r, color):  self.x=x  self.y=y  self.radius=r  self.color=color  def Area(self):  return 3.14\*self.radius\*\*2  def Color(self):  return self.color  sx=float(input())  sy=float(input())  rr=float(input())  col=input()  circle=Circle(sx,sy,rr,col)  area=circle.Area()  print(f"圆的颜色是: {circle.color}")  print(f"圆的半径是: {circle.radius}")  print(f"圆的圆心在: {circle.x,circle.y}")  print(f"圆的面积是: {area}")  **题目二**  class Course:  def \_\_init\_\_(self, code, name, teacher, location):  self.course\_code = code  self.course\_name = name  self.teacher = teacher  self.\_\_location = location  def getlocation(self):  return self.\_\_location  def setlocation(self, new\_location):  self.\_\_location = new\_location  def display(self):  print(f"课程编号: {self.course\_code}")  print(f"课程名称: {self.course\_name}")  print(f"任课教师: {self.teacher}")  print(f"上课地点: {self.\_\_location}")  course=Course("111", "cs", "张老师", "xx111")  course.display()  location=course.getlocation()  print(f"上课地点: {location}")  course.setlocation("xx222")  print(f"修改后的上课地点: {course.getlocation()}")  **题目三**  class Student:  def \_\_init\_\_(self, name, age, scores):  self.name = name  self.age = age  self.scores = scores  def getname(self):  return self.name  def getage(self):  return self.age  def gethighest(self):  highest=max(self.scores.values())  return highest  scores={"语文": 60,"数学":70,"英语": 80}  mate=Student("hello", 18, scores)  name=mate.getname()  age=mate.getage()  highest=mate.gethighest()  print(f"学生姓名: {name}")  print(f"学生年龄: {age}")  print(f"最高分数: {highest}") | | | |
| **结果分析：** | | | | |
| **教师签字** | **D:\静电脑D盘\工作\签名\李晓静.png** | **日 期** | 2023年10月18日 | |