**实 验 报 告**

**（与程序设计有关）**

**课程名称： python程序设计**

**实验题目： 类的继承和使用**

**班级学号： 2203050320**

**姓 名: 闻家尉**

**成 绩：**

**沈 阳 理 工 大 学**

**2023年 10月 27日**

|  |
| --- |
| **实验目的及要求：**  熟练掌握Python中类的继承的思想，熟练编写具有继承能力的Python程序。 |
| **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、**设计一个表示动物（Animal）的类，该类包括颜色（color）属性和叫（call）方法。再设计一个表示鱼（Fish）的类，包括尾巴（tail）和颜色（color）两个属性，及叫（call）方法。**提示：**  让Fish类继承自Animal类，重写\_\_init\_\_()和call()方法。  **2、**创建一个矩形类，包括构造方法，求周长、求面积的方法；创建一个正方形类继承矩形类，包含构造方法，求周长、求面积方法（以上方法可以继承矩形类也可以重编），输入一个矩形的长和宽，构建矩形的一个实例，求该矩形的周长与面积并输出；输入一个正方形的边长，构建正方形类的一个实例，求该正方形的周长与面积。  **3、**创建Person类，属性有姓名、年龄、性别，创建方法personInfo,打印这个人的信息.。创建Student类，继承Person类，属性有学院college，班级class，重写父类personInfo方法，调用父类方法打印个人信息外，将学生的学院、班级信息也打印出来，创建方法study参数为Teacher对象，在study函数中调用Teacher类的teach方法，接收老师教授的知识点，然后打印‘老师，xxx,我终于学会了！’xxx为老师的teach方法返回的信息（一句话即可）。重写\_\_str\_\_方法，返回某学生的信息，使print(Student类的实例对象) 能打印出该学生信息。 |

|  |
| --- |
| **程序代码或实现过程：**  1.  class Animal:  def \_\_init\_\_(self, color):  self.color = color  def call(self):  print("This is animal")  class Fish(Animal):  def \_\_init\_\_(self, color, tail):  super().\_\_init\_\_(color)  self.tail = tail  def call(self):  print("This is fish")  animal = Animal("Yellow")  fish = Fish("White","Long")  print("Animal color:",animal.color)  print("Fish color:",fish.color)  print("Fish tail:",fish.tail)  print("\n")  animal.call()  fish.call()  2.  class Rectangle:  def \_\_init\_\_(self,length,width):  self.length=length  self.width=width  def zhouchang(self):  return 2\*(self.length+self.width)  def mianji(self):  return self.length\*self.width  class Square(Rectangle):  def \_\_init\_\_(self, side):  super().\_\_init\_\_(side, side)  rectangle = Rectangle(6,2)  print("矩形周长为:", rectangle.zhouchang())  print("矩形面积为:", rectangle.mianji())  print("\n")  square = Square(3)  print("正方形周长为:", square.zhouchang())  print("正方形面积为:", square.mianji())  3.  class Person:  def \_\_init\_\_(self,name,age,gender):  self.name=name  self.age=age  self.gender=gender  def personInfo(self):  print(f"姓名：{self.name}\n年龄：{self.age}\n性别：{self.gender}")  class Student(Person):  def \_\_init\_\_(self, name, age, gender, college, cclass):  super().\_\_init\_\_(name, age, gender)  self.college = college  self.cclass = cclass    def \_\_str\_\_(self):  return f"姓名：{self.name}\n年龄：{self.age}\n性别：{self.gender}\n学院：{self.college}\n班级：{self.cclass}"  def personInfo(self):  super().personInfo()  print(f"学院：{self.college}\n班级：{self.cclass}")  def study(self, teacher):  knowledge = teacher.teach()  print(f"老师，{knowledge}, 我终于学会了！")  class Teacher:  def teach(self):  return "Python"  mate1 = Student("小明",22,"男","计算机学院","01班")  mate2 = Student("小红",20,"女","信息学院","03班")  teacher = Teacher()  mate1.personInfo()  mate1.study(teacher)  print("\n")  students = []  students.append({"姓名": mate1.name, "年龄": mate1.age, "性别": mate1.gender, "学院": mate1.college, "班级": mate1.cclass})  students.append({"姓名": mate2.name, "年龄": mate2.age, "性别": mate2.gender, "学院": mate2.college, "班级": mate2.cclass})  print("学生信息：")  for mate in students:  for key, value in mate.items():  print(f"{key}: {value}",end=" ")  print("\n")  print(mate1) |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **结果分析：**  1.    2.    3. | | | |
| **教师签字** | **D:\静电脑D盘\工作\签名\李晓静.png** | **日 期** | **2023年10月28日** |