

实验四

1.

题目：

编写代码，实现一个栈（Stack）的类。栈是只能在一端插入和删除数据的序列。它按照先进后出的原则存储数据，先进入的数据被压入栈底，最后的数据在栈顶，需要读数据的时候从栈顶开始弹出数据（最后一个数据被第一个读出来）。

思路：

在数据结构中，栈是一种具有FILO性质的线性结构，下面在python中定义一个栈类。该类拥有成员变量stack，数据类型为列表，作为数据域用于存储栈的数据元素。同时，该栈按照经典的栈结构拥有push()（入栈），pop()（出栈），peek()（取栈顶），is_empty()（判空），size()（求栈长）。在构造函数中，将实例对象的stack初始为空列表，对于该类的大多数成员函数都可以通过列表的基本方法实现。

代码：

```
class Stack:
    def __init__(self):
        self.stack = []

    def push(self, item):
        self.stack.append(item)

    def pop(self):
        if not self.is_empty():
            return self.stack.pop()
        else:
            raise IndexError("Stack is empty")

    def peek(self):
        if not self.is_empty():
            return self.stack[-1]
        else:
            raise IndexError("Stack is empty")

    def is_empty(self):
        return len(self.stack) == 0

    def size(self):
        return len(self.stack)

stack = Stack()
stack.push(1)
stack.push(2)
stack.push(3)
print(stack.pop()) # 输出: 3
print(stack.peek()) # 输出: 2
print(stack.is_empty()) # 输出: False
print(stack.size()) # 输出: 2
```

运行结果：

实例化一个stack对象，通过入栈方法在栈顶依次插入1,2,3，然后测试该栈类的各个方法：

```
PS C:\Users\gylin\Desktop\buffer\course\python\实验4> python -u "c:\Users\gylin\Desktop\buffer\course\python\实验4\1.py"
3
2
False
2
```

2.

题目：

编写代码，定义一个形状基类，有2个属性：面积和周长，以及两个无返回值的方法：area()和perimeter()，分别计算形状的面积和周长，从基类派生出三个子类：三角形、矩形、圆，重载基类的两个方法。

思路：

定义Shape类，拥有area_value, perimeter_value成员变量存储对象的面积和周长。同时定义空成员函数area(), perimeter()用于重载。在Triangle, Rectangle, Circle类中各自依照数学定义，定义构造函数，重载area(), perimeter()方法。

代码：

```
import math

class Shape:
    def __init__(self):
        self.area_value = 0
        self.perimeter_value = 0

    def area(self):
        pass

    def perimeter(self):
        pass

class Triangle(Shape):
    def __init__(self, base, height):
        super().__init__()
        self.base = base
        self.height = height

    def area(self):
        self.area_value = 0.5 * self.base * self.height

    def perimeter(self):
        self.perimeter_value = self.base + 2 * math.sqrt((0.5 * self.base)**2 + self.height**2)

class Rectangle(Shape):
    def __init__(self, length, width):
        super().__init__()
        self.length = length
        self.width = width
```

```

def area(self):
    self.area_value = self.length * self.width

def perimeter(self):
    self.perimeter_value = 2 * (self.length + self.width)

class Circle(Shape):
    def __init__(self, radius):
        super().__init__()
        self.radius = radius

    def area(self):
        self.area_value = math.pi * self.radius**2

    def perimeter(self):
        self.perimeter_value = 2 * math.pi * self.radius

triangle = Triangle(4, 5)
triangle.area()
triangle.perimeter()
print("Triangle Area:", triangle.area_value)
print("Triangle Perimeter:", triangle.perimeter_value)

rectangle = Rectangle(3, 6)
rectangle.area()
rectangle.perimeter()
print("Rectangle Area:", rectangle.area_value)
print("Rectangle Perimeter:", rectangle.perimeter_value)

circle = Circle(2)
circle.area()
circle.perimeter()
print("Circle Area:", circle.area_value)
print("Circle Perimeter:", circle.perimeter_value)

```

运行结果：

实例化三个类，并各自求面积，求周长得：

```

PS C:\Users\gylin\Desktop\buffer\course\python\实验4> python -u "c:\Users\gylin\Desktop\buffer\course\python\实验4\2.py"
Triangle Area: 10.0
Triangle Perimeter: 14.770329614269007
Rectangle Area: 18
Rectangle Perimeter: 18
Circle Area: 12.566370614359172
Circle Perimeter: 12.566370614359172
PS C:\Users\gylin\Desktop\buffer\course\python\实验4> 

```