Chap17_TimePoint_Theory&Exercises_Part2

May 25, 2020

1 Chapter 17. Classes and Methods

2 Second Part: Type-based dispatch, Polymorphism

2.1 Exercise 17.5

Write an "add" method for Points that works either a point onject or a tuple.

```
In [6]: class Point(object):
            def __init__(self, x=0, y=0):
                self.x = x
                self.y = y

            def __add__(self, other):
                point_ = Point()
                if isinstance(other, Point):
                    point_.x += self.x + other.x
                    point_.y += self.y + other.y
                    return point_
                elif type(other) == tuple:
                    point_.x += self.x + other[0]
                    point_.y += self.y + other[1]
                return point_

            def __radd__(self, other):
                return self.__add__(other)

            def __str__(self):
                return "(%s, %s)" % (self.x, self.y)

        point1 = Point(1, 6)
        point2 = (5, 2)
        point3 = point1 + point2
        point4 = point2 + point1
        print point3, point4
```

```
(6, 8) (6, 8)
```

2.2 Exercise 17.6

Download the code from this chapter (http://thinkpython.com/code/Time2.py) and...
Solution: http://thinkpython.com/code/Time2_soln.py

2.3 Exercise 17.7

class Kangaroo(object):

```
"""a cautionary tale about one of the most common,
   and difficult to find, errors in Python"""
```

http://thinkpython.com/code/BadKangaroo.py http://thinkpython.com/code/GoodKangaroo.py

2.4 Exercise 17.8

Use the Visual Python module that provides 3-D graphics.
http://thinkpython.com/code/color_space.py