Assignment #5

一、问题描述

参照七段数码管示例程序，实现对“0~9、A-E”十六进制符号的绘制。

1．用户可通过命令行交互输入任意多个字符，输入字符 A~E 大小写均可；如 2F9ce8、D35a …

2．输入其它字符需给出错误提示，如 6M35、96G…

3．扩展功能：可通过命令行交互设置数码管颜色、长度宽度等参数。输入其它字符可给出错误提示，并要求用户重新输入。

二、算法原理

此程序的功能是通过 Turtle 图形库实现的。在每一步中，海龟根据它的当前位置和方向画一条线段，然后改变它的方向并继续画线，直到达到所需的形状。用户可以输入字符、选择画笔颜色以及设置数码管的长度和宽度。

三、代码

```
import turtle


# 绘制水平段
def draw_horizontal_line(draw, horizontal_length):
    turtle.pendown() if draw else turtle.penup()
    turtle.forward(horizontal_length)
    turtle.right(90)


# 绘制垂直段
def draw_vertical_line(draw, vertical_length):
    turtle.pendown() if draw else turtle.penup()
    turtle.forward(vertical_length)
    turtle.right(90)


# 组合绘制数码管
def draw_character(character, horizontal_length, vertical_length):
```

```
    draw_horizontal_line(character in ['2', '3', '4', '5', '6', '8', '9',
'a', 'b', 'd', 'e', 'f'], horizontal_length)
    draw_vertical_line(character in ['0', '1', '3', '4', '5', '6', '7',
'8', '9', 'a', 'b', 'd'], vertical_length)
    draw_horizontal_line(character in ['0', '2', '3', '5', '6', '8', '9',
'b', 'c', 'd', 'e'], horizontal_length)
    draw_vertical_line(character in ['0', '2', '6', '8', 'a', 'b', 'c',
'd', 'e', 'f'], vertical_length)
    turtle.left(90)
    draw_vertical_line(character in ['0', '4', '5', '6', '8', '9', 'a',
'b', 'c', 'e', 'f'], vertical_length)
    draw_horizontal_line(character in ['0', '2', '3', '5', '6', '7', '8',
'9', 'a', 'c', 'e', 'f'], horizontal_length)
    draw_vertical_line(character in ['0', '1', '2', '3', '4', '7', '8',
'9', 'a', 'd'], vertical_length)
    turtle.left(180)
    turtle.penup()

    turtle.fd(horizontal_length / 4)  # 根据水平段长度设置数码管间隔


def set_pen_attributes(valid_chars):
    while True:
        date = input("请输入要显示的字符：")
        if all(char in valid_chars for char in date):
            break
        else:
            print(f"无效字符，请仅输入有效字符：{valid_chars}")

    while True:
        pen_color = input("请输入画笔颜色（例如：red、blue）：")
        if pen_color in ['red', 'blue', 'green', 'black', 'purple',
'orange']:
            break
        else:
            print("无效颜色，请输入有效颜色。")

    while True:
        try:
            horizontal_length = int(input("请输入数码管长度（整数）："))
            if horizontal_length > 0:
                break
            else:
                print("数码管长度必须是正整数。")
```

```
        except ValueError:
            print("请输入有效的数码管长度。")

    while True:
        try:
            vertical_length = int(input("请输入数码管宽度（整数）："))
            if vertical_length > 0:
                break
            else:
                print("数码管宽度必须是正整数。")
        except ValueError:
            print("请输入有效的数码管宽度。")

    return date, pen_color, horizontal_length, vertical_length


def draw_string(string, horizontal_length, vertical_length):
    string = string.lower()
    for char in string:
        draw_character(char, horizontal_length, vertical_length)


def main():
    turtle.speed(7)
    turtle.setup(800, 350, 200, 200)
    turtle.penup()
    turtle.fd(-350)
    turtle.pensize(5)

    valid_chars = "0123456789abcdefABCDEF"
    date, pen_color, horizontal_length, vertical_length =
set_pen_attributes(valid_chars)
    turtle.pencolor(pen_color)
    draw_string(date, horizontal_length, vertical_length)
    turtle.hideturtle()
    turtle.pendown()
    turtle.done()


if __name__ == "__main__":
    main()
```

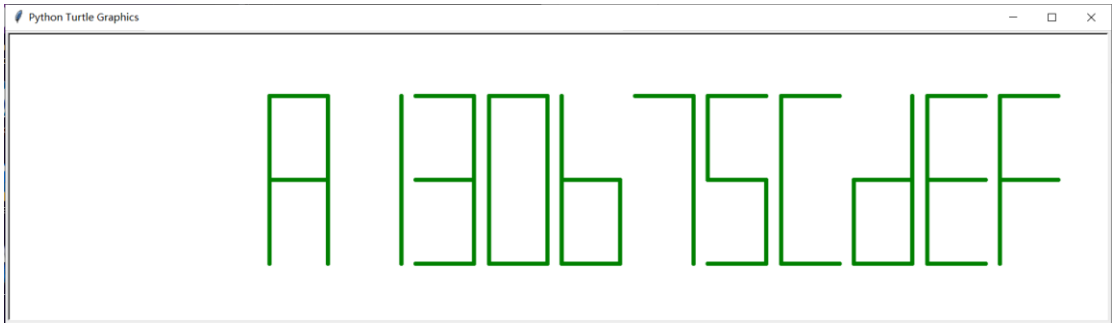
四、运行结果

输入错误提示效果:

```
PS C:\Users\livvta> & C:/Users/livvta/AppData/Local/Programs/Python/Python311/python.exe
c:/Users/livvta/OneDrive/EDU/Python/任务5_/main.py
请输入要显示的字符：PYthon123.
无效字符，请仅输入有效字符：0123456789abcdefABCDEF
请输入要显示的字符：abCDEF456
请输入画笔颜色（例如：red、blue）：RAD1
无效颜色，请输入有效颜色。
请输入画笔颜色（例如：red、blue）：red
请输入数码管长度（整数）：-60
数码管长度必须是正整数。
请输入数码管长度（整数）：六十
请输入有效的数码管长度。
请输入数码管长度（整数）：90
请输入数码管宽度（整数）：.qwe
请输入有效的数码管宽度。
请输入数码管宽度（整数）：90
PS C:\Users\livvta>
```

运行效果 1:

```
PS C:\Users\livvta> & C:/Users/livvta/AppData/Local/Programs/Python/Python311/python.exe
请输入要显示的字符：a130b75cdef
请输入画笔颜色（例如：red、blue）：green
请输入数码管长度（整数）：70
请输入数码管宽度（整数）：100
```



运行效果 2:

```
PS C:\Users\livvta> & C:/Users/livvta/AppData/Local/Programs/Python/Python311/python.exe
c:/Users/livvta/OneDrive/EDU/Python/任务5_/main.py
请输入要显示的字符：AE86F45
请输入画笔颜色（例如：red、blue）：purple
请输入数码管长度（整数）：100
请输入数码管宽度（整数）：70
```

五、分析

draw_horizontal_line 函数：

该函数用于绘制水平线段。

接受两个参数，draw 用于表示是否要绘制线段，horizontal_length 表示水平线段的长度。

如果 draw 为 True，则函数调用 Turtle 库的 pendown 函数将画笔放下，然后向前移动 horizontal_length 距离，最后右转 90 度。

如果 draw 为 False，则函数调用 penup 将画笔抬起，然后向前移动 horizontal_length 距离，最后右转 90 度。

draw_vertical_line 函数：

该函数用于绘制垂直线段。

与 draw_horizontal_line 函数类似。接受两个参数，draw 用于表示是否要绘制线段，vertical_length 表示垂直线段的长度。

draw_character 函数：

该函数根据输入的字符，绘制对应的数码管部分。

接受三个参数，character 表示要绘制的字符，horizontal_length 表示水平线段的长度，vertical_length 表示垂直线段的长度。

通过条件表达式（序列类型通用操作符 in），检查 character 是否在提供的字符集中。如果 character 包含在这个字符集中，表达式的结果将为 True，否则为 False。以此来决定是否要绘制字符相应的线段。根据七点管字符的特性，调用 draw_horizontal_line 和 draw_vertical_line 来绘制七点管对应的线段。

最后，函数将 Turtle 画笔左转 90 度并抬起画笔，向前移动 horizontal_length / 4 长度，以准备绘制下一个字符。

set_pen_attributes 函数：

该函数用于获取用户输入的字符、画笔颜色、数码管长度和宽度。

函数通过 input 获取用户输入数据。并通过无限循环 `try-except` 验证用户

的输入，确保输入的参数是有效的。

1. 获取要显示的字符，检查字符是否全部属于预定义的 valid_chars = "0123456789abcdefABCDEF"中有效字符集。若不属于，会提示错误信息，并要求用户重新输入。

2. 获取画笔颜色，确保用户输入的颜色是预定义的有效颜色['red', 'blue', 'green', 'black', 'purple', 'orange']之一。若输入不合法，会提示错误信息，并要求用户重新输入。

3. 获取数码管长度，确保用户输入的值是正整数。

通过 horizontal_length = int(input("请输入数码管长度（整数）: ")) 将用户输入的字符串转换为整数类型并赋值给变量 horizontal_length。在这行代码中，input() 函数用于从用户处获取输入，int() 函数用于将输入转换为整数类型。由于用户的输入可能会引发异常，所以使用 try 块来捕获潜在的异常。

通过 if horizontal_length > 0 语句，检查用户输入的 horizontal_length 是否大于零。如果是，表示用户提供了有效的输入，程序会执行下一步操作，即 break 语句中断循环，使程序跳出这个无限循环。

如果用户输入的值小于等于零（即无效输入），则程序会执行 else 块内的代码。即 print("数码管长度必须是正整数。")这行代码会显示一条错误消息，告诉用户数码管的长度必须是正整数。

except ValueError 是一个异常处理块，捕获由于用户输入无效数据（非整数）而引发的 ValueError 异常。如果用户输入的不是整数，程序会执行 except 块内的代码，即打印错误消息。

4. 获取数码管宽度，确保用户输入的值是正整数。

原理与获取数码管长度代码类似。

draw_string 函数：

该函数用于按顺序绘制整个输入字符串。

首先将输入字符串转换为小写字符，以便处理大小写字母。

然后通过循环，将字符串中的每个字符，调用 draw_character 函数来绘制每个字符的数码管表示。

main 函数：

此函数为程序的主要入口。

设置了 Turtle 图形的初始参数，包括画笔的速度、画布的大小，画笔的大小等。

通过调用 set_pen_attributes 函数获取用户输入的参数，包括字符、画笔颜色、数码管长度和宽度。

最后，调用 draw_string 函数来绘制输入的字符，完成数码管的绘制。

六、结论

此代码成功地使用 Turtle 库创建了一个简单的数码管式数字显示程序。用户可以输入要显示的字符，选择画笔颜色，以及数码管的长度和宽度。根据这些输入，程序生成了相应的数码管字符并将其绘制出来。

优点：

1. 使用了函数来封装绘制水平段、垂直段和数码管的逻辑。使每个函数都具有清晰的职责，提高代码的复用性和模块化。
2. 使用了列表推导式来判断字符是否在数码管的各个位置，提高代码的简洁性和效率。
3. 在处理输入错误时使用了无限循环。如果用户不小心输错了字符，不需要重新运行整个程序，只需根据提示重新输入即可。提高了程序的容错性。
4. 根据水平段长度设置数码管间隔（horizontal_length / 4），使程序在显示不同宽度数码管时更美观。
5. 使用了 input 函数来让用户输入要显示的字符、画笔颜色、数码管长度和宽度，提高代码的交互性和灵活性。

不足：

虽然此程序完成了基本和拓展任务，但仍有一些改进的空间。比如没有对用户输入的数码管长度和宽度进行合理性检查，如果用户输入了一个过大或者过小的值，程序会输出不美观或者不完整的结果。