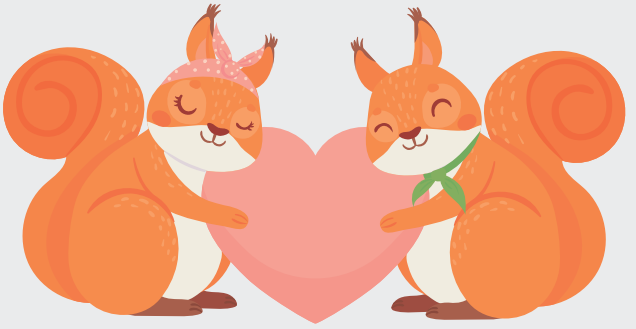
同学们，大家知道用计算机中的绘图软件可以绘制图像，但你们知道绘图软件的工作原理是什么吗？其实，通过程序编制也可以实现绘图功能，还可以绘制出动画的效果，下面就来学习如何利用计算机编程绘制图像。



1.1 课文节选案例

老舍先生的《济南的冬天》是一篇充满诗情画意的散文，作者抓住济南冬天的特点，描绘出济南冬天的独特风景，给人留下了深刻的印象。例如，文中有一段景色描绘为：“山坡上，有的地方雪厚点，有的地方草色还露着；这样，一道儿白，一道儿暗黄，给山们穿上一件带水纹的花衣。”景色如图 1-1 所示。同学们在学习这篇课文时，一定很想感受一下济南冬天的景色，看看文中所描述的景色吧？如果可以通过计算机编程来绘制场景，将有助于加深对课文的理解。下面用计算机编程来绘制该场景。

图 1-1　冬日雪景图

1.2 绘制课文中的雪景

1.2.1 编程前准备

在 Python 中可以使用 turtle 库来进行图形的绘制。turtle 库是 Python 中一个绘制图像的函数库，俗称海龟绘图，它提供了一些基本的绘图工具，可在标准的应用程序窗口中绘制各种图形。从屏幕上横轴为 x、纵轴为 y 的坐标系原点 (0,0) 位置开始，有一个像小海龟似的绘图图标，它可以根据相关函数指令和程序的控制，在这个平面坐标系中移动，从而在它爬行的路径上绘制图形，还可以通过程序控制，实现改变线段的方向、颜色、宽度等功能。下面先学习一下 turtle 库的相关操作。

1. 设置画布大小

（1）turtle.screensize(canvwidth=None, canvheight=None, bg=None)，参数分别为画布的宽 (单位：px)、高、背景颜色。

例如：

```
turtle.screensize(800,600, "blue")
turtle.screensize() # 返回默认大小 (400, 300)
```

（2）turtle.setup(width=0.5, height=0.75, startx=None,

starty=None)，参数如下。(width, height): 输入的宽和高为整数时，表示像素；为小数时，表示占据计算机屏幕的比例。(startx, starty): 这一坐标表示矩形窗口左上角顶点的位置，如果为空，则窗口位于屏幕中心。

例如：

```
turtle.setup(width=0.6,height=0.6)
turtle.setup(width=800,height=800, startx=100,starty=100)
```

2. 画笔

1）画笔的状态

在画布上，默认有一个坐标原点为画布中心的坐标轴，坐标原点上有一个面朝 x 轴正方向的小三角形。描述小三角形时使用了两个词语：坐标原点 (位置)、面朝 x 轴正方向 (方向)。turtle 绘图中，就是使用位置方向描述小三角形 (画笔) 的状态。

2）画笔的属性

（1）turtle.pensize()：设置画笔的宽度。

（2）turtle.pencolor()：没有参数传入时，返回当前画笔颜色；传入参数时，设置画笔颜色，可以是字符串如“green”“red”，也可以是 RGB 三元组。

（3）turtle.speed(speed)：设置画笔移动速度，范围为 [0,10] 中的整数，数字越大速度越快。

3）绘图命令

运用 turtle 绘图有许多命令，这些命令可以划分为三种：

画笔运动命令、画笔控制命令、全局控制命令。

（1）画笔运动命令。

```
turtle.forward(distance)     # 向当前画笔方向移动 distance
                             # 像素长度
turtle.backward(distance)    # 向当前画笔相反方向移动
                             #distance 像素长度
turtle.right(degree)         # 顺时针移动 degree
turtle.left(degree)          # 逆时针移动 degree
turtle.pendown()             # 移动时绘制图形，省略时也为绘制
turtle.goto(x,y)             # 将画笔移动到坐标为（x,y）的位置
turtle.penup()               # 提起笔移动，不绘制图形，用于另
                             # 起一个地方绘制
turtle.circle()              # 画圆，半径为正（负），表示圆心在
                             # 画笔的左边（右边）画圆
setx( )                      # 将当前 x 轴移动到指定位置
sety( )                      # 将当前 y 轴移动到指定位置
setheading(angle)            # 设置当前朝向为 angle 角度
home()                       # 设置当前画笔位置为原点，朝向东
dot(r)                       # 绘制一个指定直径的圆点
```

（2）画笔控制命令。

```
turtle.fillcolor(colorstring)   # 绘制图形的填充颜色
turtle.color(color1, color2)   # 同时设置 pencolor=color1,
```

```
                                    # fillcolor=color2
    turtle.filling()                # 返回当前是否在填充状态
    turtle.begin_fill()             # 准备开始填充图形
    turtle.end_fill()               # 填充完成
    turtle.hideturtle()             # 隐藏画笔的 turtle 形状
    turtle.showturtle()             # 显示画笔的 turtle 形状
```

（3）全局控制命令。

```
    turtle.clear()    # 清空 turtle 窗口，但是 turtle 的位置和状态
                      # 不会改变
    turtle.reset()    # 清空窗口，重置 turtle 状态为起始状态
    turtle.undo()     # 撤销上一个 turtle 动作
    turtle.isvisible()   # 返回当前 turtle 是否可见
    stamp()              # 复制当前图形
     turtle.write(s  [,font=("font-name",font_size,"font_
type")])   # 写文本，s 为文本内容，font 是字体的参数，分别为字体名称、
           # 大小和类型；font 为可选项，font 参数也是可选项
```

（4）其他命令。

```
    turtle.mainloop() 或 turtle.done()   # 启动事件循环——调用
    #Tkinter 的 mainloop() 函数。必须是图形程序中的最后一个语句
    turtle.mode(mode=None)   # 设置模式（"standard" "logo" 或
                             #"world"）并执行重置。如果没有给出
```

```
                                    # 模式，则返回当前模式
turtle.delay(delay=None)  # 设置或返回以 ms 为单位的绘图延迟
turtle.begin_poly()    # 开始记录多边形的顶点。当前的位置是多边
                                    # 形的第一个顶点
turtle.end_poly()  # 停止记录多边形的顶点。当前的位置是多边形
                                  # 的最后一个顶点。将与第一个顶点相连
turtle.get_poly()  # 返回最后记录的多边形
```

3. 命令详解

```
turtle.circle(radius, extent=None, steps=None)
```

描述：以给定半径画圆。

参数：

radius(半径)：半径为正 (负)，表示圆心在画笔的左边 (右边) 画圆。

extent：弧度，可选。

steps：作半径为 radius 的圆的内切正多边形，多边形边数为 steps；可选。

举例：

```
circle(50)           # 整圆
circle(50,steps=3)   # 三角形
circle(120, 180)     # 半圆
```

此外，除了使用 turtle 模块外，在学习图形编程时，也可使用图形库 graphics.py 来编写程序，完成简单的图形编程。可以在自己的 Python 安装目录中查找是否已有 graphics.py，如果没有这个文件，可以通过网络搜索及下载 graphic.py，将下载好的 graphics.py 文件复制到 Python\Lib 这个路径下，如图 1-2 所示。

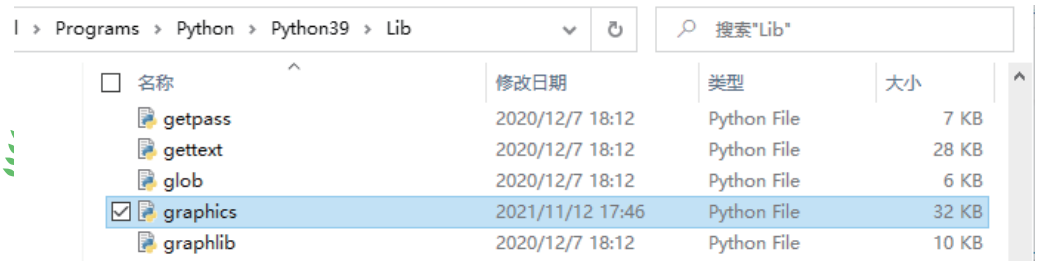


图 1-2　将 graphics 复制到 Python 的对应路径中

1.2.2　算法设计

用 Python 的 turtle 模块可以绘制很多精美的图形。turtle 库是 Python 语言中一个很流行的绘制图像的函数库，绘图时有一个小三角，在一个横轴为 x、纵轴为 y 的坐标系原点，从 (0,0) 位置开始，它根据一组函数指令的控制，在这个平面坐标系中移动，从而在它移动的路径上绘制图形。

为了与前文中的雪景相对应，先设计一段绘制“小山”的程序，勾勒出小山的轮廓；然后定义随机变量，绘制不同大小的雪花；然后运用循环语句绘制雪花飘落的效果，天空中仿佛飘落不同大小的雪花；最后绘制一些地平面的效果，营造冬天的意境。

1.3 编写程序及运行

turtle 库是 Python 语言中一个绘制图像的函数库。在使用之前首先导入 turtle 库，通过 turtle 库中的 pen() 方法，可以进行简单的绘制；使用 pendown() 方法表示落笔，可以开始绘制图形；使用 penup() 方法表示提笔；还可以用 forword() 方法使画笔前进形成一条直线；用 pensize() 方法定义笔头大小；用 pencolor() 方法定义笔刷颜色；用 hideturtle() 方法隐藏画笔标志；最后显示画布并运行画笔程序使用 turtle.done() 方法。

1.3.1 程序代码

```
import turtle as t
import random as r
# 画小山
def drawmountain():
    t.ht()  # 隐藏画笔，ht=hideturtle
    t.penup() # 提笔
    t.fd(-400)  #fd=forward，向当前画笔方向移动 -400px 长度
    t.pendown() # 落笔
    t.pensize(2)# 定义笔头大小
```

```
    t.pencolor("white")# 定义笔刷颜色为白色
    t.seth(-25)   # 设置当前朝向为 -25°
    for i in range(10): # 应用循环语句，画 9 个小山头
        t.circle(40,80) # 以给定的半径画小圆弧
        t.circle(-40,80)# 以给定的半径画小圆弧
        t.fd(40)

# 定义画雪
def drawsnow():
    t.ht()   # 隐藏笔头 ,ht=hideturtle
    t.pensize(2)   # 定义笔头大小
    for i in range(80):   # 画 79 朵雪花
        t.pencolor("white")   # 定义画笔颜色为白色
        t.pu()   # 提笔 ,pu=penup
        t.setx(r.randint(-350, 350)) # 定义 x 坐标，随机
        # 从 -350 到 350 选择
        t.sety(r.randint(1, 350))   # 定义 y 坐标，注意
        # 雪花一般在地上不会落下，所以定义是从 1 开始的
        t.pd()   # 落笔 ,pd=pendown
        dens = 6   # 雪花瓣数设为 6
        snowsize = r.randint(2, 12)   # 定义雪花大小
        for j in range(dens):   # 画 5 次，也就是一个雪花五角星
            #t.forward(int(snowsize))   #int（）表示取整数
            t.fd(int(snowsize))
```

```
            t.backward(int(snowsize))
            t.bd(int(snowsize))   # 注意没有 bd=backward,
            # 但有 fd=forward
            t.right(int(360 / dens))   # 转动角度


# 画地面线
def drawgroud():
    t.ht()        # 隐藏画笔，ht=hideturtle
    t.seth(10)   # 设置当前朝向为 10°
    for i in range(r.randint(10, 15)): #随机画几条地面线,
                                       # 但在 10~15
        # for i in range(20): # 每次操作只画 20 条地面线
        x = r.randint(-400, 350)
        y = r.randint(-280, -1)
         t.pencolor("white")
        t.pu()          # 提笔，pu=penup
        t.goto(x, y)   # 去这个坐标
        t.pd()          # 落笔，pd=pendown
        t.fd(r.randint(40, 100))   #fd=forward,向前画大小,
                                   # 随机从 40~100 选


t.setup(800, 600, 200, 200)   # 窗口大小和位置
t.tracer(True)                # 雪花和背景绘制的过程
t.bgcolor("lightblue")        #lightblue= 天蓝色
t.speed(0.3)                  # 画笔的速度
```

```
drawmountain()  # 执行画小山
drawsnow()      # 执行画雪
drawgroud()     # 执行画地面线
t.done()        # 完成
```

1.3.2 运行程序

（1）通过单击计算机 Windows 界面上的“开始”按钮，找到计算机中安装好的 Python 程序，单击 IDLE(Python) 启动编程窗口，如图 1-3 所示。

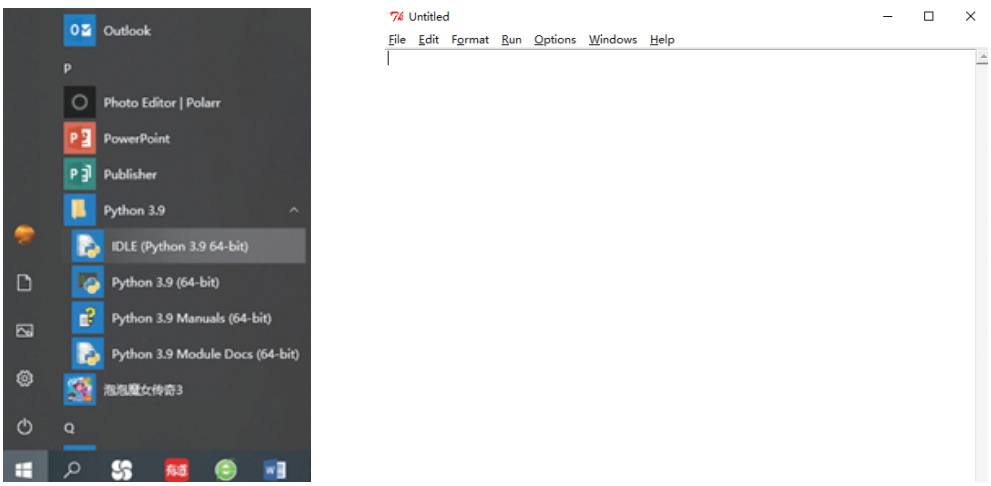


图 1-3　启动 Python 程序的编程窗口

（2）在 Python 的编程窗口中输入 1.3.1 节中的程序代码，认真检查并核对。

（3）单击菜单上的 Run → Run Module 命令，或直接按 F5 快捷键，在弹出的“保存文件”对话框中，完成保存文件操作后，调试运行该程序，如图 1-4 所示。

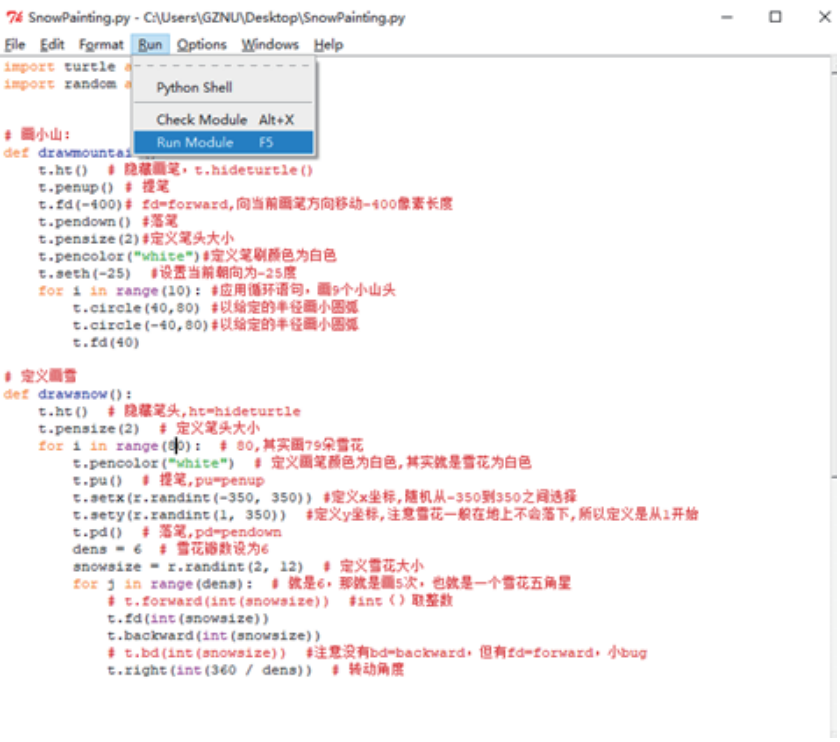


图 1-4　调试运行程序

（4）得到绘制的雪景效果图，如图 1-5 所示。

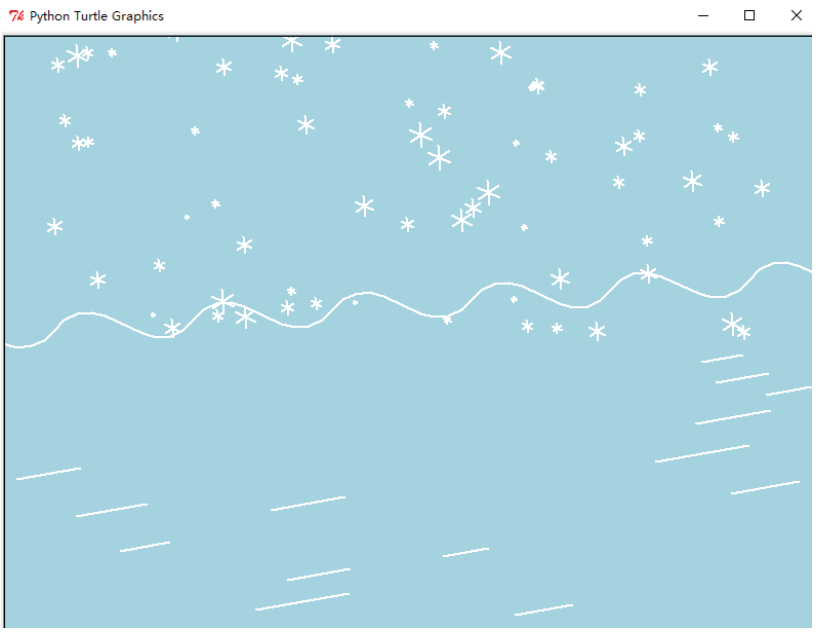


图 1-5　雪景效果图

1.4 拓展训练

在语文课本中，有很多描写景物的优美段落。例如，在一篇课文中有一段是这么写的：“山坡上卧着些小村庄，小村庄的房顶上卧着点雪，对，这是张小水墨画，也许是唐代的名手画的吧。”（选自老舍先生的散文《济南的冬天》）请同学们认真品味文中的意境，在 1.3.1 节程序代码的基础上，试着绘制几座小房子组成的小村庄。