



图 4-30：过程 Flower 及其绘制结果

我们可以继续使用过程 **Flower** 构建更复杂的图形。但是无论有多么复杂，其思路依然是清晰的：从最初的过程 **Leaf** 到调用它的 **Leaves** 过程，再到调用它们绘制树枝的过程 **Branch**，最后到调用过程 **Branch** 绘制一朵看似复杂的花的过程 **Flower**。你甚至可以继续创建一棵花团锦簇的树，然后让这些树组成一个大花园。

通过这个案例，你明白什么叫作自底向上的分析了吗？我们一开始并不关注整个问题的复杂性，而是先从更小、更容易管理的片段着手，然后将这些片段整合在一起，从而解决了整个问题。使用这种思维方式，我们可以先创建过程解决简单的问题，再创建更复杂的过程调用之前的过程。

本章小结

本章介绍了许多本书广泛使用的基本概念。首先是消息广播机制，它能让角色之间进行交流，并同步各个角色之间的行为。接着讲解了结构化程序设计，同时使用广播机制模拟了过程，随后使用 Scratch 2 的新特性新建功能块创建过程，并讨论了如何传递参数以及过程调用过程，即过程的嵌套。最后介绍了两种分析问题的思维方式，一种是自顶向下地把问题分解为更小、更易管理的片段，另一种是自底向上地把已经解决的小片段整合在一起。

下一章将学习在任何编程语言中都最重要的概念：变量。理解它是进阶为熟练的程序员至关重要的一步。

练习题

1. 编写过程，绘制你名字中的各个字母，过程名是绘制的字母。例如，用过程 **A** 绘制字母 **A**。然后编写一段脚本调用这些过程，把你的名字绘制在舞台上。