程序设计思维大作业报告

王天宇 2000013108

June 24, 2022

Contents

1 介绍

本项目为 2022 春季程序设计思维 python 流程图生成大作业。实现了给定 python 代码，程序自动生成对应流程图的功能，并配置了相应的前端网页实现。项目后端由 python 实现代码的转换，借由 python 的 flask 框架完成服务器部分，前端 html+js 实现最后的流程图绘制。

2 项目结构

2.1 myflowchart

后端 python 的实现。

包含三个文件，其中类的继承关系是：Node 是所有类型的基类，Ast_node 是面向 ast 所有的节点类型的基类。其他的类继承自这两个类。flowchart 类是最终的封装。

2.1.1 node.py

面向 flowchart.js，实现了对应的节点类别，以及对应的输入输出功能。

类名和 flowchart.js 中的 NodeType 的一一对应：

1. StartNode – start
2. EndNode – end
3. OperationNode – operation
4. InputOutputNode – inputoutput
5. SubroutineNode – subroutine
6. ConditionNode – condition

实现了图类（节点组 NodesGroup 类），方便在代码流程图上进行遍历等操作。

2.1.2 ast_node.py

面向 _ast 包的类定义，将 _ast 包中读取出来的对象转化为自定义的类。

类名和 ast 中的 Node 的对应：

1. IfCondition & If – If
2. LoopCondition & Loop – For, While
3. Try – Try
4. FunctionDefStart & FunctionDefEnd & FunctionDefArgsInput & FunctionDef – FunctionDef
5. CommonOperation – Expression, Assign
6. Return & ReturnOutput – Return
7. YieldOutput – Yield

2.1.3 myflowchart.py

最终的类封装，利用 Node 和 AstNode 类生成的 NodesGroup 输出解析后的语句。

2.2 static

前端静态部分。

2.2.1 img

网页中用到的图片，包括网页背景以及帮助页面的示例图。

2.2.2 js

用到的 js 库，包括 bootstrap、jquery、dropzone、raphael、flowchart、ace 等。

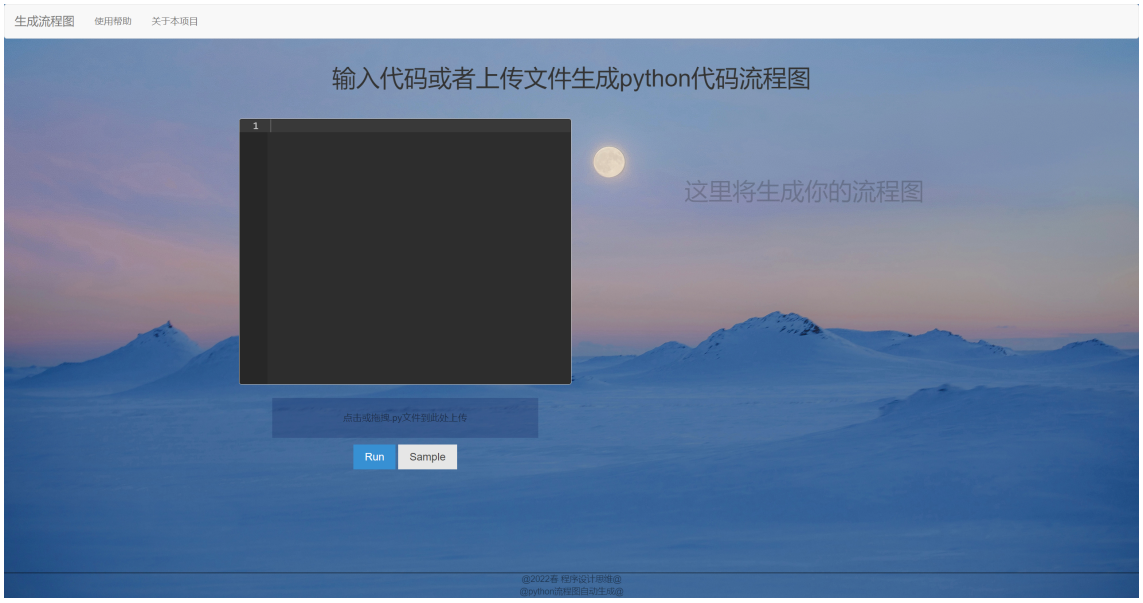
2.2.3 about.html

“关于”页面，展示了本项目的简要信息。

2.2.4 help.html

“帮助”页面，展示了网页前端的使用教程。

2.2.5 home.html



主界面，以下是主要功能：

1. 导航栏
2. 代码输入框
3. 代码拖拽上传
4. 运行、下载 svg、下载 png、展示样例代码
5. 流程图展示区域

flowchart_server.py

服务器，采用 python 的 flask 框架。

3 运行项目

1. 安装 requirements.txt 中要求的 python 包
2. 运行 flowchart_server.py
3. 根据端口打开网页即可使用

4 项目原理

通过解析 ast 包生成的 python 语法树模型，得到语句的内容、类型、作用域、上下关系，并以此构建我们的流程图。

4.1 节点定义

语句的定义部分通过 python 的面向对象设计完成，我定义了与 flowchart.js 中 NodeType 的一一对应的类，借助 ast 包的 unparse 来从 ast 树模型中获取原来的 python 代码，再从中解析出语句类型，由时间戳和语句类型生成语句的 id，这样就可以完成 flowchart 的第一部分，即节点定义部分。

4.2 节点关系

在最后的流程图中，每个节点就是一条 python 语句，而每个边就是语句之间的执行关系。

我们的流程图是一种近似树形的图。大多数的语句都是顺序执行的，包括常规的赋值、计算、函数调用、条件分支等；只有循环语句可能会在图中添加反向边（而且 python 没有 goto 语句）所以，根据 ast 包中每条语句的作用域，我们把一条语句看作父节点，作用域内的语句看作它的后代，可以得出以下结论：

1. 儿子节点都是顺序执行的。
2. 每个完整的代码块的作用域都是一个子图，包括一个开始节点（执行此代码块的第一条语句），以及一个或多个结束节点（此代码块执行的最后一条语句）而每个代码块之间的连接为上一个代码块的结束节点连接到此代码块的开始节点。
3. 此代码块与其他代码块的执行顺序由此子树根节点的语句类型决定。

根据以上结论，我们很方便的可以从 ast 树模型中提取出语句的执行关系。这样就可以完成 flowchart.js 的第二部分，即语句连接部分。

4.3 流程图生成

流程图的生成交由 flowchart.js 完成，我们将最后生成的解析语句传递给前端，在前端直接调用 flowchart.js 生成最终的流程图并显示。

References

[1] *flowchart.js* by Adriano Raiano, https://github.com/adrai/flowchart.js

[2] *pyflowchart* by cdfmlr, https://github.com/cdfmlr/pyflowchart/

[3] *Static Modification of Python With Python: The AST Module*, https://blueprintforge.com/blog/2012/02/27/static-modification-of-python-with-python-the-ast-module/

[4] python 3.10.5 documentation, https://docs.python.org/3/library/ast.html

[5] bootstrap documentation, https://getbootstrap.com/docs/5.2/getting-started/introduction/

[6] dropzone.js documentation, https://docs.dropzone.dev/

[7] Ace API Reference, https://ace.c9.io/#nav=api