



# Code Visualization

Milestone 6



## Progress Matrix

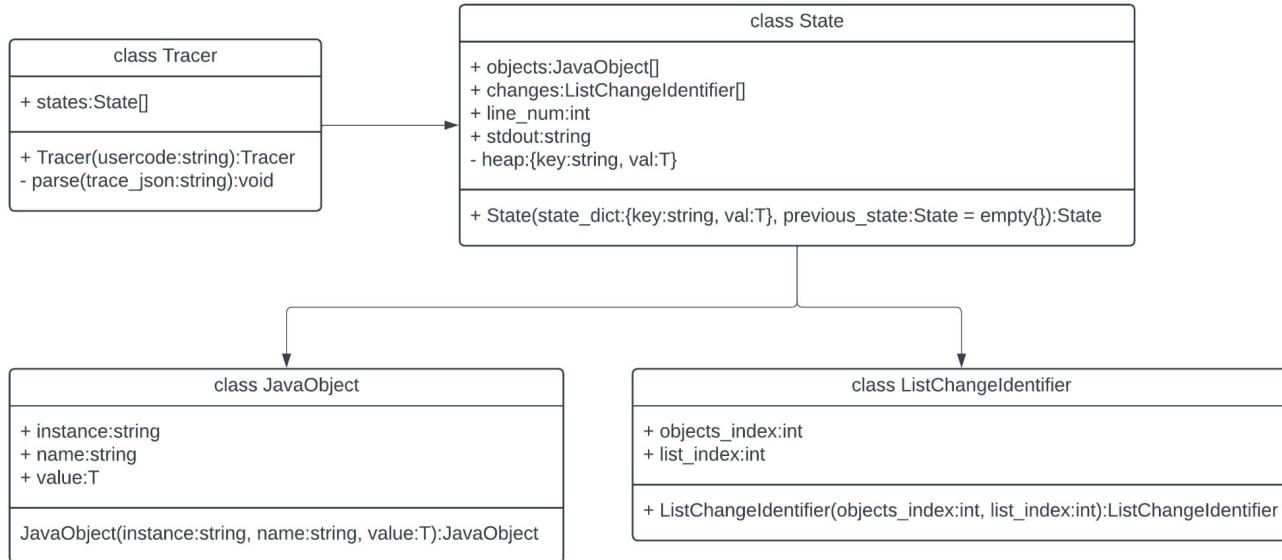
Task	Completion %	Curtice	Josh	Catherine
1. Add LinkedList functionality to parser	100%	100%	0%	0%
2. Add finishing touches to GUI	100%	0%	100%	0%
3. Create user/developer manual	100%	50%	50%	0%
4. Create demo video	100%	0%	0%	100%

---

# 1. Add LinkedList functionality to parser

Curtice Gough

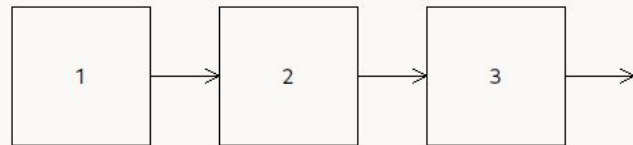
# Recall



```
git diff trace.py
```

```
70 +
71 +         elif heap_item[0] == 'INSTANCE':
72 +             if heap_item[1] == 'LinkedList': # handle LinkedList
73 +                 list_var_old = None
74 +                 for obj in previous_state.objects:
75 +                     if obj.name == varname:
76 +                         list_var_old = obj.value
77 +                         break
78 +
79 +                 list_var = []
80 +
81 +                 if heap_item[2][1] != None: # if head != null
82 +                     node = self.heap[str(heap_item[2][1][1])]
83 +                     list_var.append(node[2][1])
84 +
85 +                 while(node[3][1] != None): # Iterate through LinkedList
86 +                     node = self.heap[str(node[3][1][1])]
87 +                     list_var.append(node[2][1])
88 +
89 +                 self.objects.append(JavaObject('LIST', varname, list_var))
90 +                 for i in range(len(list_var)):
91 +                     try:
92 +                         if list_var[i] != list_var_old[i]: # Changed item
93 +                             print(f"{varname}[{i}] changed from {list_var_old[i]} to {list_var[i]}")
94 +                             self.changes.append(ListChangeIdentifier(len(self.objects) - 1, i))
95 +                     except: # New item
96 +                         self.changes.append(ListChangeIdentifier(len(self.objects) - 1, i))
```

```
public class Main {  
    public static void main(String[] args) {  
        LinkedList<Integer> list = new LinkedList<>();  
  
        list.add(1);  
        list.add(2);  
        list.add(3);  
  
        System.out.println("Size of the list: " + list.size());  
        System.out.println("Elements of the list:");  
        list.printList();  
    }  
}
```



All Variables in the state will be shown here

All Stdout will be shown here

Execute

Step Forwards

Step Backwards

---

## 2. Add finishing touches to GUI

Joshua Hartzfeld



## Unexpected issues

After many hours of troubleshooting i was unable to implement a working line tracing solution in PyQt6 and PyQt5.

After isolating the project to just a code editor I was still unable to use crucial properties of QCursor QTextDocument, and QTextBlock. Some methods outline in the documentation were unusable causing the GUI to crash.

This has been outlined for any future groups to figure out.



---

# 3. Create user/developer manual

Curtice Gough  
Joshua Hartzfeld

<https://curtico.github.io/code-visualization/docs/manual.pdf>

The image shows a file explorer on the left side of a code editor. The file explorer lists several files: main.tex, screenshot.png, screenshot2.png, screenshot3.png, system-architecture.png, and trace-uml.png. Below the file explorer is a 'File outline' section with a tree view. The tree view is expanded to show the following structure:

- Introduction
  - Overview
  - Core Features
  - User Demographics
- Installation
  - System Requirements
    - Operating System
    - Python Libraries
    - Java JDK
  - Compiling from Source
- Getting Started
  - Code Editor
  - Data Structures View
  - STDOUT
- Supported Data Structures
  - Primitives
  - Arrays

```
1 \documentclass{scrreprt}
2 \usepackage{listings}
3 \usepackage{underscore}
4 \usepackage{graphicx}
5 \usepackage{xcolor}
6 \usepackage[bookmarks=true]{hyperref}
7 \usepackage[utf8]{inputenc}
8 \usepackage[english]{babel}
9 \hypersetup{
10   bookmarks=false, % show bookmarks bar?
11   pdftitle={User/Developer Manual}, % title
12   pdfauthor={Curtice Gough}, % author
13   pdfsubject={TeX and LaTeX}, % subject of the document
14   pdfkeywords={TeX, LaTeX, graphics, images}, % list of keywords
15   colorlinks=true, % false: boxed links; true: colored links
16   linkcolor=blue, % color of internal links
17   citecolor=black, % color of links to bibliography
18   filecolor=black, % color of file links
19   urlcolor=purple, % color of external links
20   linktoc=page % only page is linked
21 }%
22 \def\myversion{1.0 }
23 \date{}
24 %\title{}
25
26 %}
27 \usepackage{hyperref}
28
29 \definecolor{codegreen}{rgb}{0,0.6,0}
30 \definecolor{codegray}{rgb}{0.5,0.5,0.5}
31 \definecolor{codepurple}{rgb}{0.58,0,0.82}
32 \definecolor{backcolour}{rgb}{0.95,0.95,0.92}
33
34 \lstdefinestyle{mystyle}{
35   backgroundcolor=\color{backcolour},
36   commentsstyle=\color{codegreen},
37   keywordsstyle=\color{magenta},
38   numberstyle=\tiny\color{codegray},
39   stringstyle=\color{codepurple},
40   basicstyle=\ttfamily\footnotesize,
41   breakatwhitespace=false,
42   breaklines=true,
43   captionpos=b,
44   keepspaces=true,
45   number=left
```

## Contents

<b>1 Introduction</b>	<b>3</b>
1.1 Overview	3
1.2 Core Features	3
1.3 User Demographics	3
<b>2 Installation</b>	<b>4</b>
2.1 System Requirements	4
2.1.1 Operating System	4
2.1.2 Python Libraries	4
2.1.3 Java JDK	4
2.2 Compiling from Source	4
<b>3 Getting Started</b>	<b>5</b>
3.1 Code Editor	5
3.2 Data Structures View	6
3.3 STDOUT	6
<b>4 Supported Data Structures</b>	<b>7</b>
4.1 Primitives	7
4.2 Arrays	7
4.3 Linked Lists	8
4.4 Custom Data Structures	9
<b>5 Further Development</b>	<b>11</b>
5.1 System Architecture	11
5.2 Parsing Traceprinter Output	12
5.2.1 JSON Structure	12
5.2.2 trace.py UML	15
5.3 GUI Development	15
5.3.1 PyQt6 Widgets	15

---

# 4. Create demo video

Catherine DiResta



# Demo Video

<https://drive.google.com/file/d/1rMWXDUIukP47zJGj0nZI-pQLEcK8j-1s/view?usp=sharing>

**Thank You**

