



การใช้ AI ใน การเขียนโค้ดและทดสอบโปรแกรม
Leveraging AI for Coding and Software Testing

จัดทำโดย

นายชนินทร์ รัญสิริพัฒนาดา	รหัสนักศึกษา 653380125-2
นายชลพัฒน์ ปั่นมนี	รหัสนักศึกษา 653380126-0
นายปวิณวัฒน์ สุขร่วม	รหัสนักศึกษา 653380136-7

อาจารย์ประจำวิชา

ผศ. ดร.ชิตสุรา สุ่มเล็ก

เอกสารนี้เป็นส่วนหนึ่งของวิชา การประกันคุณภาพซอฟต์แวร์
(Software Quality Assurance)
ประจำภาคเรียนที่ 1 ปีการศึกษา 2567

ชื่อหัวข้อโครงการภาษาไทย	: การใช้ AI ในการเขียนโค้ดและทดสอบโปรแกรม
ชื่อหัวข้อโครงการภาษาอังกฤษ	: Leveraging AI for Coding and Software Testing
ชื่อผู้จัดทำโครงการ	: นายชนินทร์ รัญสิริพัฒนาрадา รหัสนักศึกษา 653380125-2
	นายชลพัฒน์ ปั่นมนี รหัสนักศึกษา 653380126-0
	นายปริณวัฒน์ สุขรุ่ม รหัสนักศึกษา 653380136-7
อาจารย์ประจำวิชา	: ผศ. ดร.ชิตสุรา สุ่มเล็ก
สาขาวิชา	: วิทยาการคอมพิวเตอร์ คณะวิทยาลัยการคุณพิวเตอร์
ปีการศึกษา	: 2567

บทคัดย่อ

ในยุคปัจจุบัน AI ได้เข้ามายึดทบทำสำคัญในวงการพัฒนาซอฟต์แวร์ โดยเฉพาะด้านการเขียนโค้ดและการทดสอบ ซึ่งคาดหวังว่าจะช่วยเพิ่มทั้งประสิทธิภาพและความถูกต้องของซอฟต์แวร์อย่างมีนัยสำคัญ งานวิจัยนี้จึงเน้นศึกษาผลกระทบของ AI ที่มีต่อกระบวนการพัฒนาซอฟต์แวร์ โดยเปรียบเทียบโค้ดที่สร้างจาก ChatGPT4.0, Gemini 1.5 Pro, Gemini 1.5 Flash และ GitHub Copilot เพื่อวิเคราะห์ประสิทธิผลและความถูกต้องของ Generative AI แต่ละตัว การวิจัยนี้ครอบคลุมการประเมินคุณภาพโค้ดในหลากหลายด้าน เช่น การตรวจจับข้อบกพร่อง การปฏิบัติตามมาตรฐานการเขียนโค้ด และการประเมินประสิทธิภาพการทำงานของโปรแกรม ผลการวิจัยจะช่วยให้เราเข้าใจถึงจุดเด่นและข้อจำกัดของการใช้ Generative AI ในการพัฒนาและทดสอบซอฟต์แวร์ ทั้งในเรื่องความรวดเร็ว ความแม่นยำ และคุณภาพโดยรวม ข้อมูลจากการวิจัยนี้จะเป็นประโยชน์อย่างมากต่อผู้พัฒนาและผู้บริหารซอฟต์แวร์ในการตัดสินใจเลือกใช้เทคโนโลยี AI ที่เหมาะสมและมีประสิทธิภาพ ในการบูรณาการพัฒนาซอฟต์แวร์ต่อไป

ผลการทดสอบ

Prompt ในการ Generate Code ในแต่ละรอบ

Design Pattern : Flyweight Pattern

ภาษาโปรแกรมที่ใช้ : Java

รอบที่ 1

Write Java code that use flyweight pattern with requirement below

"ระบบแสดงผลข้อความ(Text Rendering System)"

1. การจัดการฟอนต์ (Font Management)

- การโหลดฟอนต์หลายประเภท: ระบบต้องรองรับการแสดงผลฟอนต์หลายแบบ เช่น Sans-serif, Serif, Monospace รวมถึงฟอนต์พิเศษ เช่น ฟอนต์ที่สนับสนุนหลายภาษา
- ขนาดฟอนต์ (Font Size): ควรสามารถแสดงข้อความในขนาดฟอนต์ที่แตกต่างกัน เช่น ใหญ่ขึ้นหรือเล็กลงตามความต้องการ
- การปรับรูปแบบฟอนต์: รองรับการแสดงผลฟอนต์ที่มีรูปแบบหลากหลาย เช่น Bold, Italic, Underline

2. การจัดการสีข้อความ (Text Color)

- ต้องสามารถกำหนดสีของข้อความได้ เช่น สีดำ สีแดง หรือสีอื่นๆ

3. การจัดเรียงข้อความ (Text Alignment)

- การจัดซิดซ้าย (Left Alignment): การจัดข้อความให้อยู่ทางด้านซ้ายของพื้นที่แสดงผล
- การจัดกึ่งกลาง (Center Alignment): ข้อความที่อยู่กลางหน้าจอหรือพื้นที่
- การจัดซิดขวา (Right Alignment): ข้อความที่ถูกจัดซิดทางขวา
- Justify: การจัดข้อความให้เรียงเต็มพื้นที่ทั้งซ้ายและขวา

4. การเว้นระยะระหว่างบรรทัด (Line Spacing)

- ควรรองรับการปรับระยะห่างระหว่างบรรทัด เพื่อความสวยงามและง่ายต่อการอ่าน

5. การจัดการระยะห่างระหว่างตัวอักษร (Kerning/Letter Spacing)

- ควรสามารถปรับระยะห่างระหว่างตัวอักษรแต่ละตัวได้ เพื่อให้การแสดงผลดูสวยงามและมีอ้าวซีพมากขึ้น

6. การรองรับหลายภาษา (Multilingual Support)

- ระบบต้องรองรับการแสดงผลข้อความในหลายภาษา รวมถึงภาษาที่ต้องการการจัดเรียงพิเศษ เช่น ภาษาอาหรับที่ต้องแสดงจากขวาไปซ้าย หรือภาษาจีนที่อาจมีการจัดรูปแบบตัวอักษรแนวตั้ง

7. การจัดการข้อความในหลายรูปแบบ (Rich Text Support)

- รองรับการแสดงข้อความที่มีการเน้นหลายรูปแบบในประโยชน์เดียว เช่น ข้อความที่มีตัวหนา ตัวเอียง หรือการขีดเส้นใต้ในส่วนเดียวกัน

- การแสดงผลที่มีการแทรกภาพ หรือสัญลักษณ์ในข้อความ

8. การจัดการข้อความในพื้นที่จำกัด (Text Wrapping)

- ระบบต้องรองรับการตัดคำ (word wrapping) เมื่อข้อความยาวเกินพื้นที่ที่กำหนด เช่น การตัดคำในกรอบข้อความ หรือในคอลัมน์

- การตัดคำและการแสดงผลข้อความในหลายบรรทัดในกรณีที่ข้อความมีความยาวเกินกว่าพื้นที่แสดงผล

9. การจัดการตัวอักษรพิเศษและ Unicode (Special Characters & Unicode Support)

- ควรรองรับการแสดงผลตัวอักษรพิเศษ เช่น เครื่องหมายทางคณิตศาสตร์, สัญลักษณ์, อิมจิ, และ Unicode characters

Write Junit that have requirement below

- Branch coverage 100%

- Statement coverage 100%

รอบที่ 2

Use Java language to write a code that use flyweight pattern with requirement below

"ระบบแสดงผลข้อความ(Text Rendering System)"

1. การจัดการฟอนต์ (Font Management)

- การโหลดฟอนต์หลายประเภท: ระบบต้องรองรับการแสดงผลฟอนต์หลายแบบ เช่น Sans-serif, Serif, Monospace รวมถึงฟอนต์พิเศษ เช่น ฟอนต์ที่สนับสนุนหลายภาษา

- ขนาดฟอนต์ (Font Size): ควรสามารถแสดงข้อความในขนาดฟอนต์ที่แตกต่างกัน เช่น ใหญ่ขึ้นหรือเล็กลงตามความต้องการ

- การปรับรูปแบบฟอนต์: รองรับการแสดงผลฟอนต์ที่มีรูปแบบหลากหลาย เช่น Bold, Italic, Underline

2. การจัดการสีข้อความ (Text Color)

- ต้องสามารถกำหนดสีของข้อความได้ เช่น สีดำ สีแดง หรือสีอื่นๆ

3. การจัดเรียงข้อความ (Text Alignment)

- การจัดซ้าย (Left Alignment): การจัดข้อความให้อยู่ทางด้านซ้ายของพื้นที่แสดงผล

- การจัดกลาง (Center Alignment): ข้อความที่อยู่กลางหน้าจอหรือพื้นที่

- การจัดขวา (Right Alignment): ข้อความที่ถูกจัดซิดทางขวา

- Justify: การจัดข้อความให้เรียงเต็มพื้นที่ทั้งซ้ายและขวา

4. การเว้นระยะระหว่างบรรทัด (Line Spacing)

- ควรรองรับการปรับระยะห่างระหว่างบรรทัด เพื่อความสวยงามและง่ายต่อการอ่าน

5. การจัดการระยะห่างระหว่างตัวอักษร (Kerning/Letter Spacing)

- ควรสามารถปรับระยะห่างระหว่างตัวอักษรแต่ละตัวได้ เพื่อให้การแสดงผลดูสวยงามและมีอาชีพมากขึ้น

6. การรองรับหลายภาษา (Multilingual Support)

- ระบบต้องรองรับการแสดงผลข้อความในหลายภาษา รวมถึงภาษาที่ต้องการการจัดเรียงพิเศษ เช่น ภาษาอาหรับที่ต้องแสดงจากขวาไปซ้าย หรือภาษาจีนที่อาจมีการจัดรูปแบบตัวอักษรแนวตั้ง

7. การจัดการข้อความในหลายรูปแบบ (Rich Text Support)

- รองรับการแสดงข้อความที่มีการเน้นหลายรูปแบบในประโยคเดียว เช่น ข้อความที่มีตัวหนา ตัวเอียง หรือการขีดเส้นใต้ในส่วนเดียวกัน

- การแสดงผลที่มีการแทรกภาพ หรือสัญลักษณ์ในข้อความ

8. การจัดการข้อความในพื้นที่จำกัด (Text Wrapping)

- ระบบต้องรองรับการตัดคำ (word wrapping) เมื่อข้อความยาวเกินพื้นที่ที่กำหนด เช่น การตัดคำในกรอบข้อความ หรือในคอลัมน์

- การตัดคำและการแสดงผลข้อความในหลายบรรทัดในกรณีที่ข้อความมีความยาวเกินกว่าพื้นที่แสดงผล

9. การจัดการตัวอักษรพิเศษและ Unicode (Special Characters & Unicode Support)

- ควรรองรับการแสดงผลตัวอักษรพิเศษ เช่น เครื่องหมายทางคณิตศาสตร์, สัญลักษณ์ อิมิจิ, และ Unicode characters

Write Junit with requirements below

- Branch coverage 100%

- Statement coverage 100%

รอบที่ 3

Write a code that use flyweight pattern with Java that have requirement below

"ระบบแสดงผลข้อความ(Text Rendering System)"

1. การจัดการฟอนต์ (Font Management)

- การโหลดฟอนต์หลายประเภท: ระบบต้องรองรับการแสดงผลฟอนต์หลายแบบ เช่น Sans-serif, Serif, Monospace รวมถึงฟอนต์พิเศษ เช่น ฟอนต์ที่สนับสนุนหลายภาษา

- ขนาดฟอนต์ (Font Size): ความสามารถแสดงข้อความในขนาดฟอนต์ที่แตกต่างกัน เช่น ใหญ่ขึ้นหรือเล็กลงตามความต้องการ

- การปรับรูปแบบฟอนต์: รองรับการแสดงผลฟอนต์ที่มีรูปแบบหลากหลาย เช่น Bold, Italic, Underline

2. การจัดการสีข้อความ (Text Color)

- ต้องสามารถกำหนดสีของข้อความได้ เช่น สีดำ สีแดง หรือสีอื่นๆ

3. การจัดเรียงข้อความ (Text Alignment)

- การจัดซิดซ้าย (Left Alignment): การจัดข้อความให้อยู่ทางด้านซ้ายของพื้นที่แสดงผล

- การจัดกึ่งกลาง (Center Alignment): ข้อความที่อยู่กลางหน้าจอหรือพื้นที่

- การจัดซิดขวา (Right Alignment): ข้อความที่ถูกจัดซิดทางขวา

- Justify: การจัดข้อความให้เรียงเต็มพื้นที่ทั้งซ้ายและขวา

4. การเว้นระยะระหว่างบรรทัด (Line Spacing)

- ควรรองรับการปรับระยะห่างระหว่างบรรทัด เพื่อความสวยงามและง่ายต่อการอ่าน

5. การจัดการระยะห่างระหว่างตัวอักษร (Kerning/Letter Spacing)

- ความสามารถปรับระยะห่างระหว่างตัวอักษรแต่ละตัวได้ เพื่อให้การแสดงผลดูสวยงามและมีอาชีพมากขึ้น

6. การรองรับหลายภาษา (Multilingual Support)

- ระบบต้องรองรับการแสดงผลข้อความในหลายภาษา รวมถึงภาษาที่ต้องการการจัดเรียงพิเศษ เช่น ภาษาอาหรับที่ต้องแสดงจากขวาไปซ้าย หรือภาษาจีนที่อาจมีการจัดรูปแบบตัวอักษรแนวตั้ง

7. การจัดการข้อความในหลายรูปแบบ (Rich Text Support)

- รองรับการแสดงข้อความที่มีการเน้นหลายรูปแบบในประโยคเดียว เช่น ข้อความที่มีตัวหนา ตัวเอียง หรือการขีดเส้นใต้ในส่วนเดียวกัน

- การแสดงผลที่มีการแทรกภาพ หรือสัญลักษณ์ในข้อความ

8. การจัดการข้อความในพื้นที่จำกัด (Text Wrapping)

- ระบบต้องรองรับการตัดคำ (word wrapping) เมื่อข้อความยาวเกินพื้นที่ที่กำหนด เช่น การตัดคำในกรอบข้อความ หรือในคอลัมน์

- การตัดคำและการแสดงผลข้อความในหลายบรรทัดในกรณีที่ข้อความมีความยาวเกินกว่าพื้นที่แสดงผล

9. การจัดการตัวอักษรพิเศษและ Unicode (Special Characters & Unicode Support)

- ควรรองรับการแสดงผลตัวอักษรพิเศษ เช่น เครื่องหมายทางคณิตศาสตร์, สัญลักษณ์, อิมิจิ, และ Unicode characters

Write JUnit to test requirement below

- Branch coverage 100%

- Statement coverage 100%

Design Pattern : Interpreter Pattern

ภาษาโปรแกรมที่ใช้ : Java

รอบที่ 1

Write a code that use Interpreter pattern with java that have requirement below

1. การจัดการนิพจน์ทางคณิตศาสตร์พื้นฐาน:

- รองรับการประเมินค่าของนิพจน์พื้นฐาน เช่น การบวก, ลบ, คูณ, และหาร
- สามารถใช้วงเล็บเพื่อกำหนดลำดับการคำนวณได้

2. การจัดการข้อผิดพลาด:

- สามารถจัดการกับข้อผิดพลาดสำหรับนิพจน์ที่ไม่ถูกต้องหรือไม่สามารถประเมินค่าได้

รอบที่ 2

Write a code that use Interpreter pattern with java that have requirement below

1. การจัดการนิพจน์ทางคณิตศาสตร์พื้นฐาน:

- รองรับการประเมินค่าของนิพจน์พื้นฐาน เช่น การบวก, ลบ, คูณ, และหาร
- สามารถใช้วงเล็บเพื่อกำหนดลำดับการคำนวณได้

2. การจัดการข้อผิดพลาด:

- สามารถจัดการกับข้อผิดพลาดสำหรับนิพจน์ที่ไม่ถูกต้องหรือไม่สามารถประเมินค่าได้

รอบที่ 3

Write a code that use Interpreter pattern with java that have requirement below

1. การจัดการนิพจน์ทางคณิตศาสตร์พื้นฐาน:

- รองรับการประเมินค่าของนิพจน์พื้นฐาน เช่น การบวก, ลบ, คูณ, และหาร
- สามารถใช้วงเล็บเพื่อกำหนดลำดับการคำนวณได้

2. การจัดการข้อผิดพลาด:

- สามารถจัดการกับข้อผิดพลาดสำหรับนิพจน์ที่ไม่ถูกต้องหรือไม่สามารถประเมินค่าได้

3. การจัดการตัวแปร:

- สามารถจัดการกับตัวแปรและประเมินค่าตามนิพจน์ที่กำหนด

4. การประเมินค่าหلامยระดับ:

- รองรับการประเมินค่าที่มีความซับซ้อนสูง เช่น นิพจน์ที่มีการรวมหلامยประเกท

5. การใช้งาน:

- แสดงวิธีการใช้ Interpreter ในการประเมินค่าของนิพจน์ที่หلامยประเกท

Additionally, provide JUnit tests ensuring:

- 100% branch coverage.

- 100% statement coverage.

Design Pattern : Flyweight Pattern

ภาษาโปรแกรมที่ใช้ : Python

รอบที่ 1

Write python code that use flyweight pattern with requirement below

"ระบบแสดงผลข้อความ(Text Rendering System)"

1. การจัดการฟอนต์ (Font Management)

- การโหลดฟอนต์หلامยประเกท: ระบบต้องรองรับการแสดงผลฟอนต์หلامยแบบ เช่น Sans-serif, Serif, Monospace รวมถึงฟอนต์พิเศษ เช่น ฟอนต์ที่สนับสนุนหلامภาษา
- ขนาดฟอนต์ (Font Size): ควรสามารถแสดงข้อความในขนาดฟอนต์ที่แตกต่างกัน เช่น ใหญ่ขึ้นหรือเล็กลงตามความต้องการ
- การปรับรูปแบบฟอนต์: รองรับการแสดงผลฟอนต์ที่มีรูปแบบหلام เช่น Bold, Italic, Underline

2. การจัดการสีข้อความ (Text Color)

- ต้องสามารถกำหนดสีของข้อความได้ เช่น สีดำ สีแดง หรือสีอื่นๆ

3. การจัดเรียงข้อความ (Text Alignment)

- การจัดซิดซ้าย (Left Alignment): การจัดข้อความให้อยู่ทางด้านซ้ายของพื้นที่การแสดงผล
- การจัดกึ่งกลาง (Center Alignment): ข้อความที่อยู่กลางหน้าจอหรือพื้นที่
- การจัดซิดขวา (Right Alignment): ข้อความที่ถูกจัดซิดทางขวา
- Justify: การจัดข้อความให้เรียงเต็มพื้นที่ทั้งซ้ายและขวา

4. การเว้นระยะระหว่างบรรทัด (Line Spacing)

- ควรรองรับการปรับระยะห่างระหว่างบรรทัด เพื่อความสวยงามและง่ายต่อการอ่าน

5. การจัดการระยะห่างระหว่างตัวอักษร (Kerning/Letter Spacing)

- ควรสามารถปรับระยะห่างระหว่างตัวอักษรแต่ละตัวได้ เพื่อให้การแสดงผลดูสวยงามและมีอ้าชีพมากขึ้น

6. การรองรับหلامภาษา (Multilingual Support)

- ระบบต้องรองรับการแสดงผลข้อความในหลายภาษา รวมถึงภาษาที่ต้องการการจัดเรียงพิเศษ เช่น ภาษาอาหรับที่ต้องแสดงจากขวาไปซ้าย หรือภาษาจีนที่อาจมีการจัดรูปแบบตัวอักษรแนวตั้ง

7. การจัดการข้อความในหลายรูปแบบ (Rich Text Support)

- รองรับการแสดงข้อความที่มีการเน้นหลายรูปแบบในประโยชน์เดียว เช่น ข้อความที่มีตัวหนา ตัวเอียง หรือการขีดเส้นใต้ในส่วนเดียวกัน

- การแสดงผลที่มีการแทรกภาพ หรือสัญลักษณ์ในข้อความ

8. การจัดการข้อความในพื้นที่จำกัด (Text Wrapping)

- ระบบต้องรองรับการตัดคำ (word wrapping) เมื่อข้อความยาวเกินพื้นที่ที่กำหนด เช่น การตัดคำในกรอบข้อความ หรือในคอลัมน์

- การตัดคำและการแสดงผลข้อความในหลายบรรทัดในกรณีที่ข้อความมีความยาวเกินกว่าพื้นที่แสดงผล

9. การจัดการตัวอักษรพิเศษและ Unicode (Special Characters & Unicode Support)

- ควรรองรับการแสดงผลตัวอักษรพิเศษ เช่น เครื่องหมายทางคณิตศาสตร์, สัญลักษณ์ อิมิจิ, และ Unicode characters

Write pytest that have requirement below

- Branch coverage 100%

- Statement coverage 100%

รอบที่ 2

Use python language to write a code that use flyweight pattern with requirement below

"ระบบแสดงผลข้อความ(Text Rendering System)"

1. การจัดการฟอนต์ (Font Management)

- การโหลดฟอนต์หลายประเภท: ระบบต้องรองรับการแสดงผลฟอนต์หลายแบบ เช่น Sans-serif, Serif, Monospace รวมถึงฟอนต์พิเศษ เช่น ฟอนต์ที่สนับสนุนหลายภาษา

- ขนาดฟอนต์ (Font Size): ควรสามารถแสดงข้อความในขนาดฟอนต์ที่แตกต่างกัน เช่น ใหญ่ขึ้นหรือเล็กลงตามความต้องการ

- การปรับรูปแบบฟอนต์: รองรับการแสดงผลฟอนต์ที่มีรูปแบบหลากหลาย เช่น Bold, Italic, Underline

2. การจัดการสีข้อความ (Text Color)

- ต้องสามารถกำหนดสีของข้อความได้ เช่น สีดำ สีแดง หรือสีอื่นๆ

3. การจัดเรียงข้อความ (Text Alignment)

- การจัดชิดซ้าย (Left Alignment): การจัดข้อความให้อยู่ทางด้านซ้ายของพื้นที่แสดงผล

- การจัดกึ่งกลาง (Center Alignment): ข้อความที่อยู่กลางหน้าจอหรือพื้นที่

- การจัดชิดขวา (Right Alignment): ข้อความที่ถูกจัดชิดทางขวา

- Justify: การจัดข้อความให้เรียงเต็มพื้นที่ทั้งซ้ายและขวา

4. การเว้นระยะระหว่างบรรทัด (Line Spacing)

- ควรรองรับการปรับระยะห่างระหว่างบรรทัด เพื่อความสวยงามและง่ายต่อการอ่าน

5. การจัดการระยะห่างตัวอักษร (Kerning/Letter Spacing)

- ความสามารถปรับระยะห่างระหว่างตัวอักษรแต่ละตัวได้ เพื่อให้การแสดงผลดูสวยงามและมีอ้าวีพมากขึ้น

6. การรองรับหลายภาษา (Multilingual Support)

- ระบบต้องรองรับการแสดงผลข้อความในหลายภาษา รวมถึงภาษาที่ต้องการการจัดเรียงพิเศษ เช่น ภาษาอาหรับที่ต้องแสดงจากขวาไปซ้าย หรือภาษาจีนที่อาจมีการจัดรูปแบบตัวอักษรแนวตั้ง

7. การจัดการข้อความในหลายรูปแบบ (Rich Text Support)

- รองรับการแสดงข้อความที่มีการเน้นหลายรูปแบบในประโยคเดียว เช่น ข้อความที่มีตัวหนา ตัวเอียง หรือการขีดเส้นใต้ในส่วนเดียวกัน

- การแสดงผลที่มีการแทรกภาพ หรือสัญลักษณ์ในข้อความ

8. การจัดการข้อความในพื้นที่จำกัด (Text Wrapping)

- ระบบต้องรองรับการตัดคำ (word wrapping) เมื่อข้อความยาวเกินพื้นที่ที่กำหนด เช่น การตัดคำในกรอบข้อความ หรือในคอลัมน์

- การตัดคำและการแสดงผลข้อความในหลายบรรทัดในกรณีที่ข้อความมีความยาวเกินกว่าพื้นที่แสดงผล

9. การจัดการตัวอักษรพิเศษและ Unicode (Special Characters & Unicode Support)

- ควรรองรับการแสดงผลตัวอักษรพิเศษ เช่น เครื่องหมายทางคณิตศาสตร์, สัญลักษณ์ อิมิจิ, และ Unicode characters

Write pytest with requirements below

- Branch coverage 100%

- Statement coverage 100%

รอบที่ 3

Write a code that use flyweight pattern with python that have requirement below

"ระบบแสดงผลข้อความ(Text Rendering System)"

1. การจัดการฟอนต์ (Font Management)

- การโหลดฟอนต์หลายประเภท: ระบบต้องรองรับการแสดงผลฟอนต์หลายแบบ เช่น Sans-serif, Serif, Monospace รวมถึงฟอนต์พิเศษ เช่น ฟอนต์ที่สนับสนุนหลายภาษา

- ขนาดฟอนต์ (Font Size): ความสามารถแสดงข้อความในขนาดฟอนต์ที่แตกต่างกัน เช่น ใหญ่ขึ้นหรือเล็กลงตามความต้องการ
- การปรับรูปแบบฟอนต์: รองรับการแสดงผลฟอนต์ที่มีรูปแบบหลากหลาย เช่น Bold, Italic, Underline

2. การจัดการสีข้อความ (Text Color)

- ต้องสามารถกำหนดสีของข้อความได้ เช่น สีดำ สีแดง หรือสีอื่นๆ

3. การจัดเรียงข้อความ (Text Alignment)

- การจัดซิดซ้าย (Left Alignment): การจัดข้อความให้อยู่ทางด้านซ้ายของพื้นที่แสดงผล
- การจัดกึ่งกลาง (Center Alignment): ข้อความที่อยู่กลางหน้าจอหรือพื้นที่
- การจัดซิดขวา (Right Alignment): ข้อความที่ถูกจัดซิดทางขวา
- Justify: การจัดข้อความให้เรียงเต็มพื้นที่ทั้งซ้ายและขวา

4. การเว้นระยะระหว่างบรรทัด (Line Spacing)

- ควรรองรับการปรับระยะห่างระหว่างบรรทัด เพื่อความสวยงามและง่ายต่อการอ่าน

5. การจัดการระยะห่างตัวอักษร (Kerning/Letter Spacing)

- ความสามารถปรับระยะห่างระหว่างตัวอักษรแต่ละตัวได้ เพื่อให้การแสดงผลดูสวยงามและมีอาชีพมากขึ้น

6. การรองรับหลายภาษา (Multilingual Support)

- ระบบต้องรองรับการแสดงผลข้อความในหลายภาษา รวมถึงภาษาที่ต้องการการจัดเรียงพิเศษ เช่น ภาษาอาหรับที่ต้องแสดงจากขวาไปซ้าย หรือภาษาจีนที่อาจมีการจัดรูปแบบตัวอักษรแนวตั้ง

7. การจัดการข้อความในหลายรูปแบบ (Rich Text Support)

- รองรับการแสดงข้อความที่มีการเน้นหลายรูปแบบในประโยคเดียว เช่น ข้อความที่มีตัวหนา ตัวเอียง หรือการขีดเส้นใต้ในส่วนเดียวกัน

- การแสดงผลที่มีการแทรกภาพ หรือสัญลักษณ์ในข้อความ

8. การจัดการข้อความในพื้นที่จำกัด (Text Wrapping)

- ระบบต้องรองรับการตัดคำ (word wrapping) เมื่อข้อความยาวเกินพื้นที่ที่กำหนด เช่น การตัดคำในกรอบข้อความ หรือในคอลัมน์

- การตัดคำและการแสดงผลข้อความในหลายบรรทัดในกรณีที่ข้อความมีความยาวเกินกว่าพื้นที่แสดงผล

9. การจัดการตัวอักษรพิเศษและ Unicode (Special Characters & Unicode Support)

- ควรรองรับการแสดงผลตัวอักษรพิเศษ เช่น เครื่องหมายทางคณิตศาสตร์, สัญลักษณ์, อิโมจิ, และ Unicode characters

Write pytest to test requirement below

- Branch coverage 100%
- Statement coverage 100%

Design Pattern : Interpreter Pattern

ภาษาโปรแกรมที่ใช้ : Python

รอบที่ 1

Write a code that use Interpreter pattern with python that have requirement below

1. การจัดการนิพจน์ทางคณิตศาสตร์พื้นฐาน:

- รองรับการประเมินค่าของนิพจน์พื้นฐาน เช่น การบวก, ลบ, คูณ, และหาร
- สามารถใช้วงเล็บเพื่อกำหนดลำดับการคำนวณได้

2. การจัดการข้อผิดพลาด:

- สามารถจัดการกับข้อผิดพลาดสำหรับนิพจน์ที่ไม่ถูกต้องหรือไม่สามารถประเมินค่าได้

3. การจัดการตัวแปร:

- สามารถจัดการกับตัวแปรและประเมินค่าตามนิพจน์ที่กำหนด

4. การประเมินค่าหลายระดับ:

- รองรับการประเมินค่าที่มีความซับซ้อนสูง เช่น นิพจน์ที่มีการรวมหลายประเภท

5. การใช้งาน:

- แสดงวิธีการใช้ Interpreter ในการประเมินค่าของนิพจน์ที่หลากหลาย

Additionally, provide pytest tests ensuring:

- 100% branch coverage.
- 100% statement coverage.

รอบที่ 2

Write a code that use Interpreter pattern with python that have requirement below

1. การจัดการนิพจน์ทางคณิตศาสตร์พื้นฐาน:

- รองรับการประเมินค่าของนิพจน์พื้นฐาน เช่น การบวก, ลบ, คูณ, และหาร
- สามารถใช้วงเล็บเพื่อกำหนดลำดับการคำนวณได้

2. การจัดการข้อผิดพลาด:

- สามารถจัดการกับข้อผิดพลาดสำหรับนิพจน์ที่ไม่ถูกต้องหรือไม่สามารถประเมินค่าได้

3. การจัดการตัวแปร:

- สามารถจัดการกับตัวแปรและประเมินค่าตามนิพจน์ที่กำหนด

4. การประเมินค่าหลายระดับ:

- รองรับการประเมินค่าที่มีความซับซ้อนสูง เช่น นิพจน์ที่มีการรวมหลายประเภท

5. การใช้งาน:

- แสดงวิธีการใช้ Interpreter ในการประเมินค่าของนิพจน์ที่หลากหลาย

Additionally, provide pytest tests ensuring:

- 100% branch coverage.
- 100% statement coverage.

รอบที่ 3

Write a code that use Interpreter pattern with python that have requirement below

1. การจัดการนิพจน์ทางคณิตศาสตร์พื้นฐาน:

- รองรับการประเมินค่าของนิพจน์พื้นฐาน เช่น การบวก, ลบ, คูณ, และหาร
- สามารถใช้วงเล็บเพื่อกำหนดลำดับการคำนวณได้

2. การจัดการข้อผิดพลาด:

- สามารถจัดการกับข้อผิดพลาดสำหรับนิพจน์ที่ไม่ถูกต้องหรือไม่สามารถประเมินค่าได้

3. การจัดการตัวแปร:

- สามารถจัดการกับตัวแปรและประเมินค่าตามนิพจน์ที่กำหนด

4. การประเมินค่าหลายระดับ:

- รองรับการประเมินค่าที่มีความซับซ้อนสูง เช่น นิพจน์ที่มีการรวมหลายประเภท

5. การใช้งาน:

- แสดงวิธีการใช้ Interpreter ในการประเมินค่าของนิพจน์ที่หลากหลาย

Additionally, provide pytest tests ensuring:

- 100% branch coverage.
- 100% statement coverage.

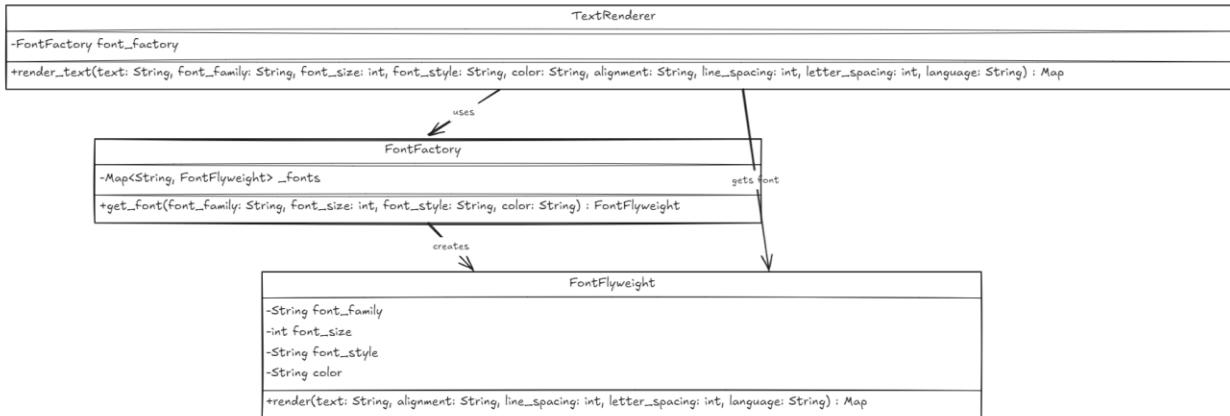
Diagram ที่ได้จากการ Generate Code

ภาษาโปรแกรมที่ใช้: Python

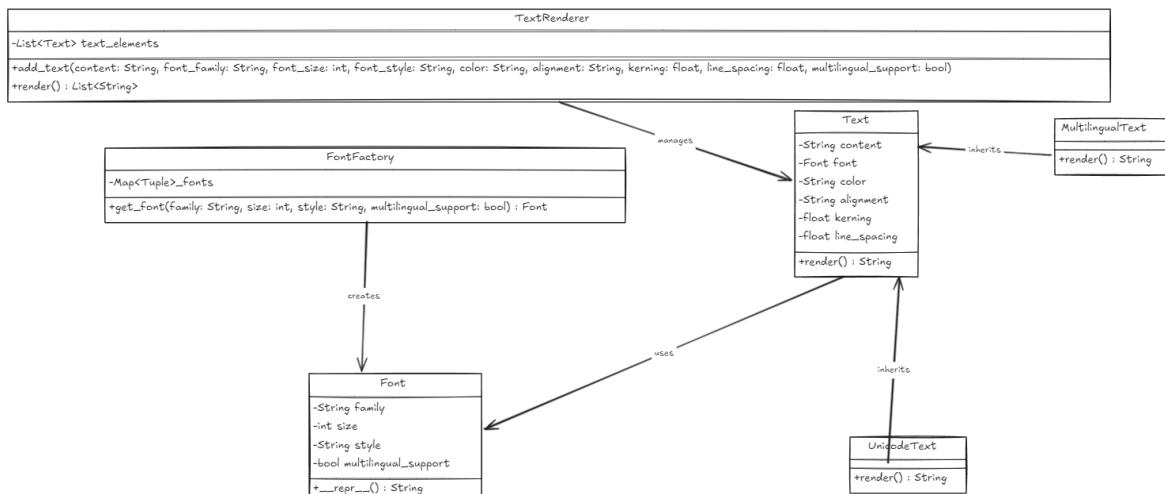
AI ที่ใช้: ChatGPT4.0

Design pattern: Flyweight

รอบที่ 1



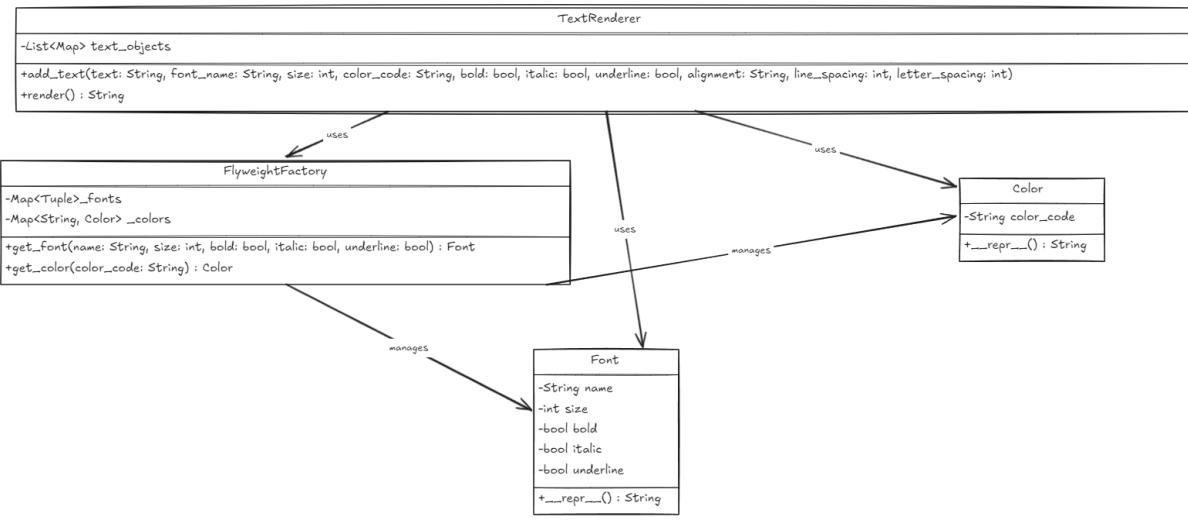
รอบที่ 2



Activate Windo
Go to Settings to act

รอบที่ 3

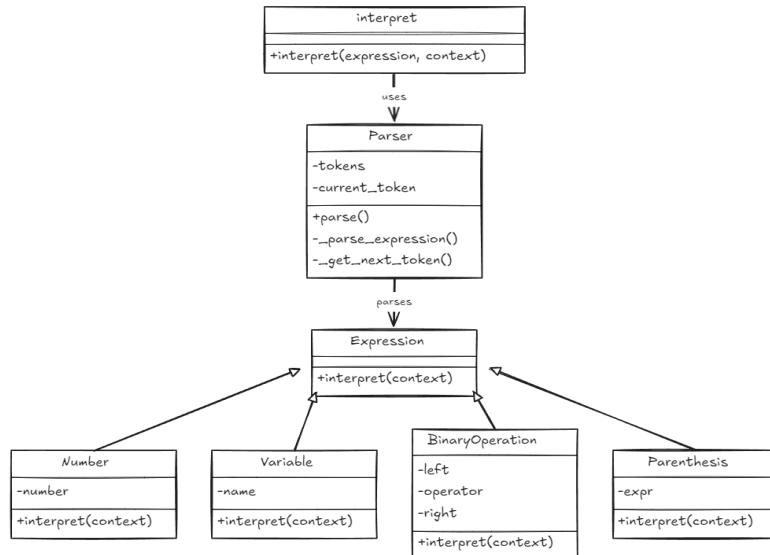
Diagram



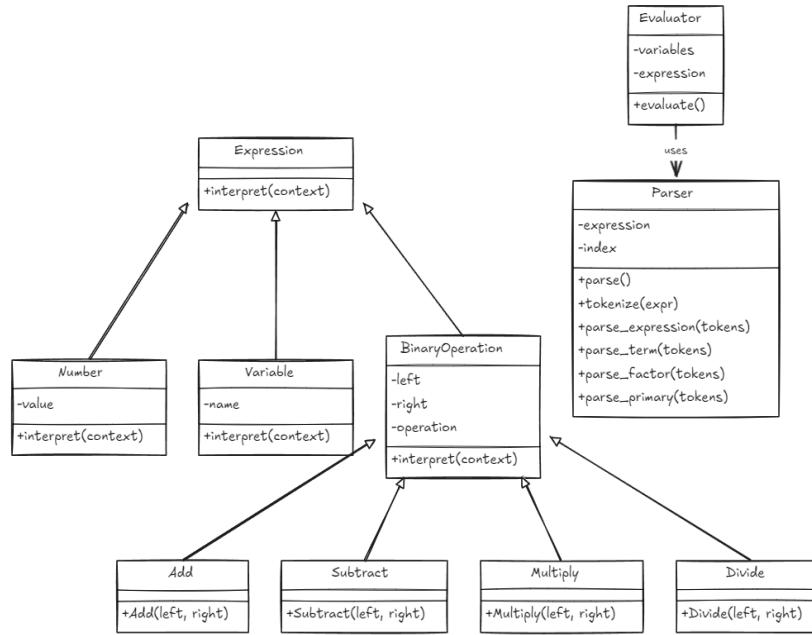
AI ที่ใช้: ChatGPT4.0

Design pattern: Interpreter

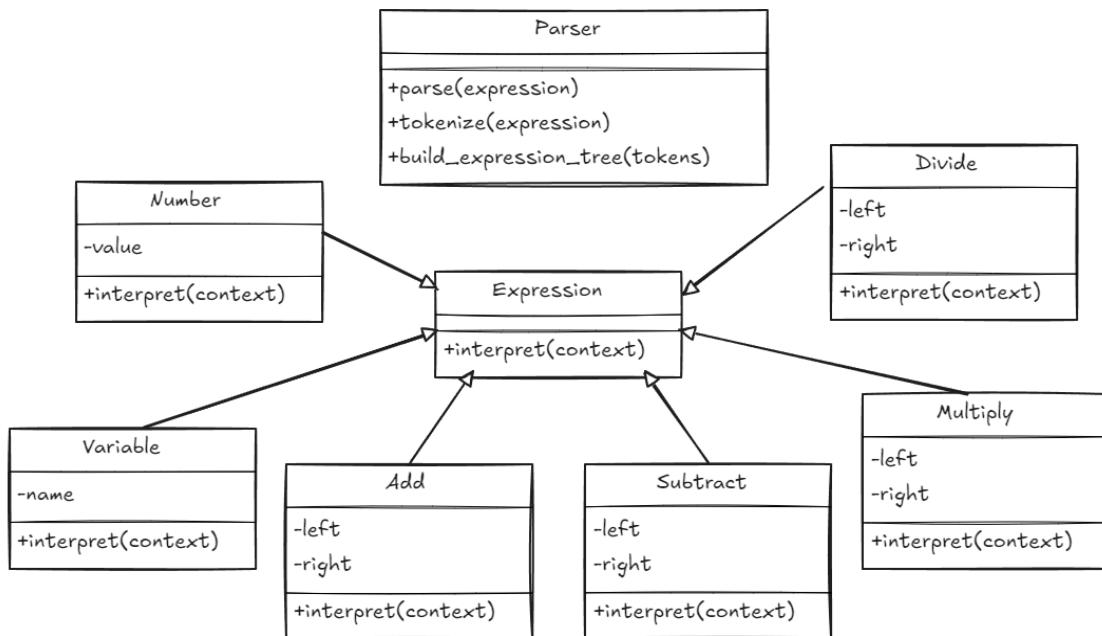
รอบที่ 1



รูปที่ 2



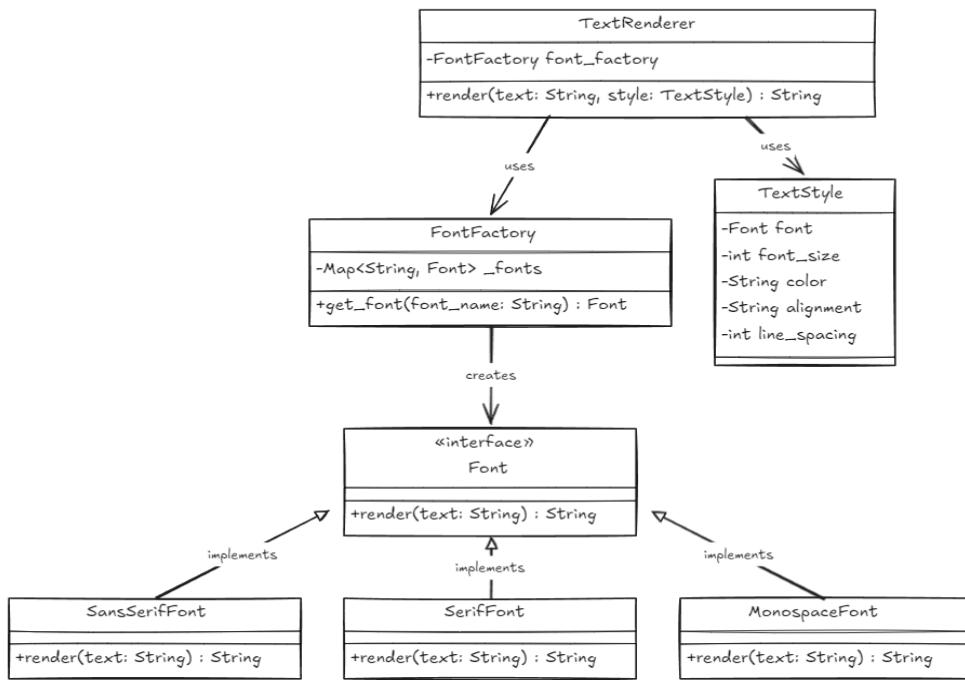
รูปที่ 3



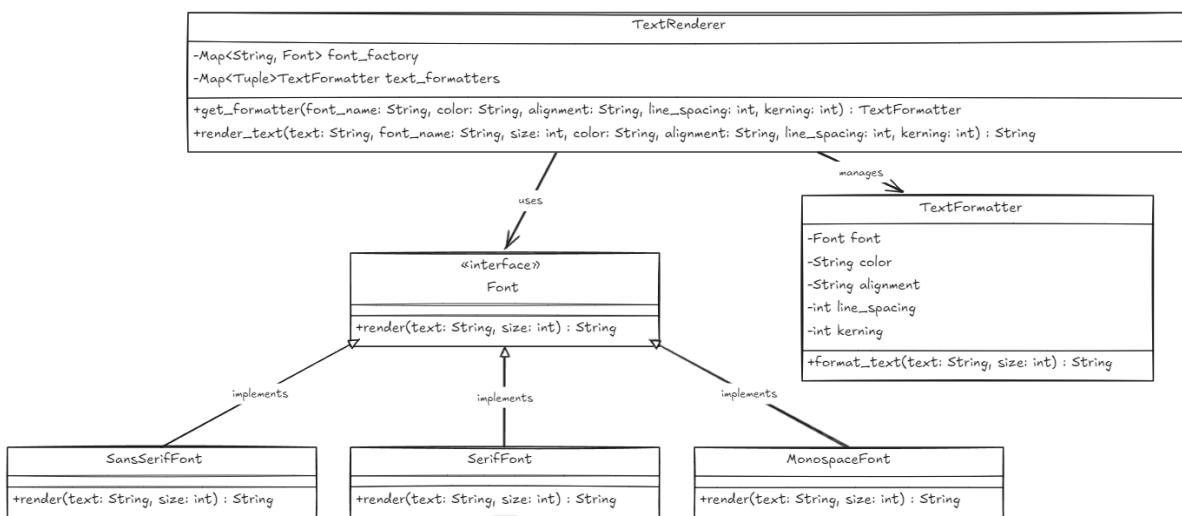
AI ที่ใช้: Gemini 1.5 Flash

Design pattern: Flyweight

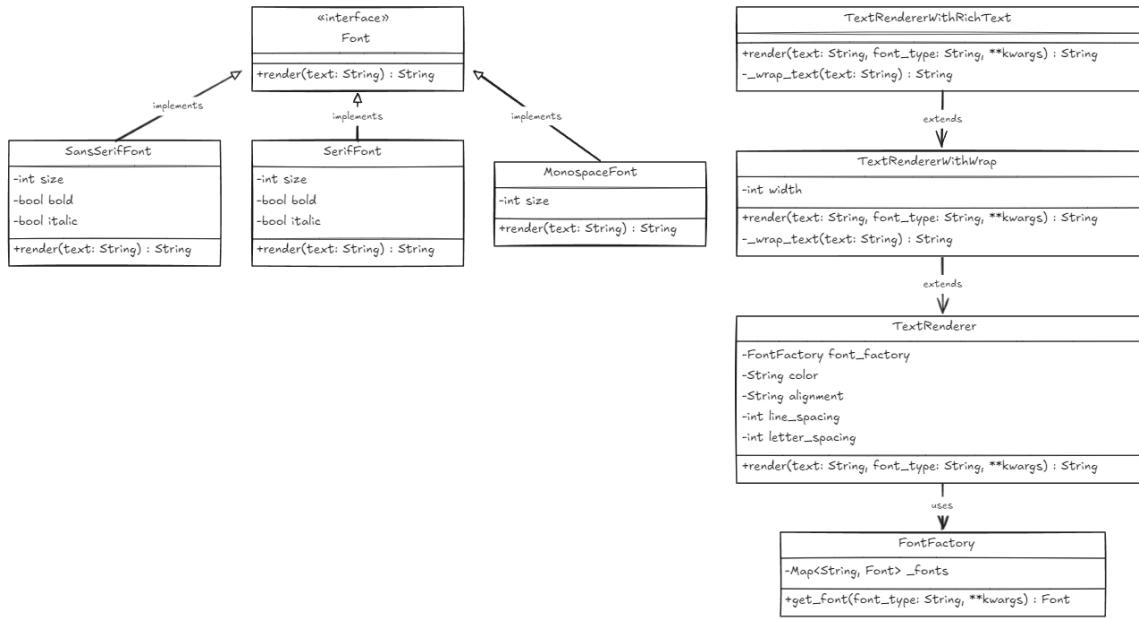
รอบที่ 1



รอบที่ 2



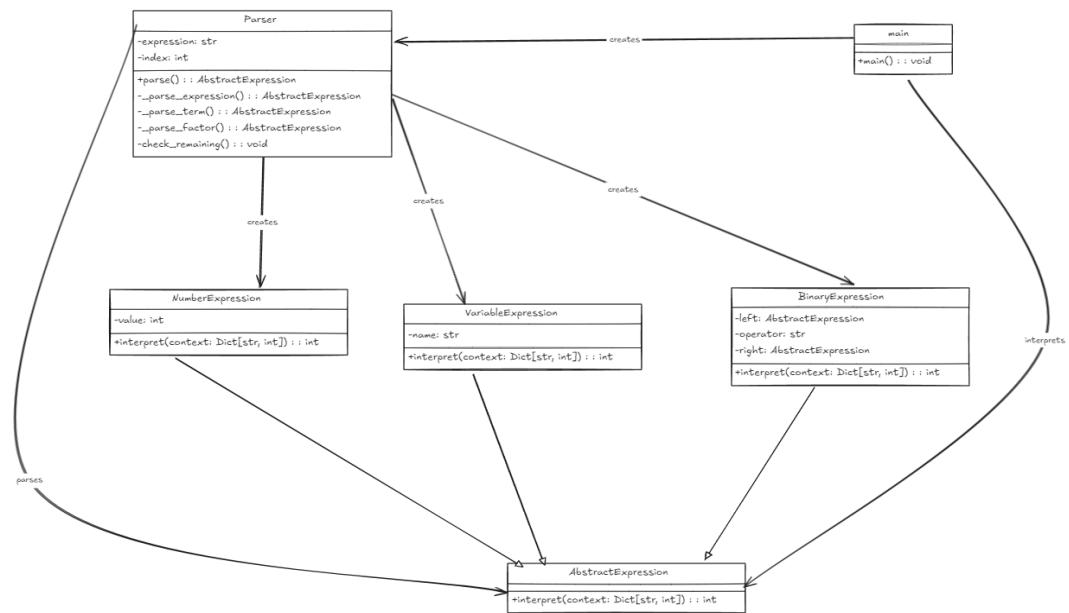
รอบที่ 3



AI ที่ใช้: Gemini 1.5 Flash

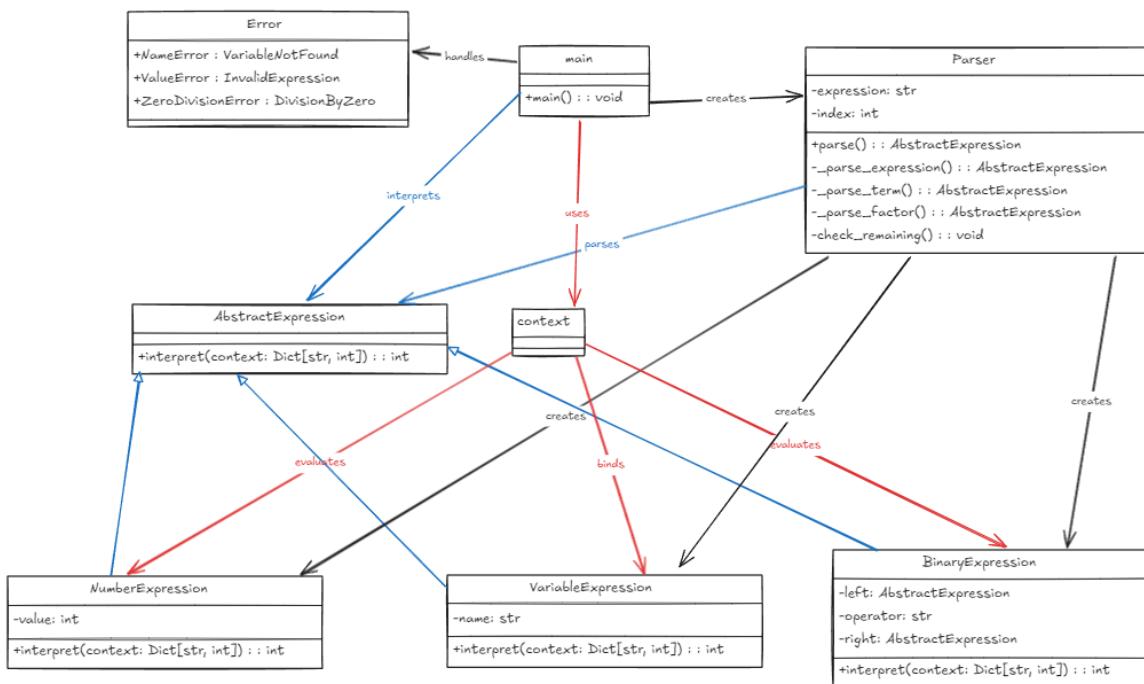
Design pattern: Interpreter

รอบที่ 1

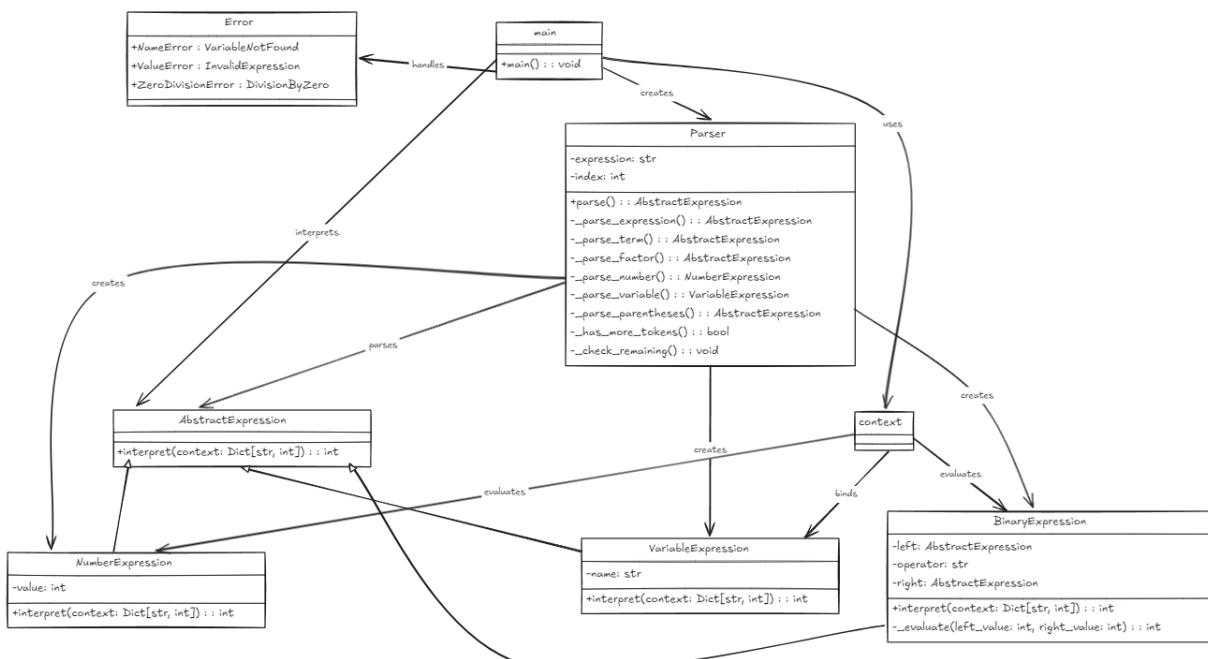


รอบที่ 2

- * ความลับที่เป็นรากที่เก็บของ context
- * ความลับที่เป็นรากที่เก็บของ context



รอบที่ 3

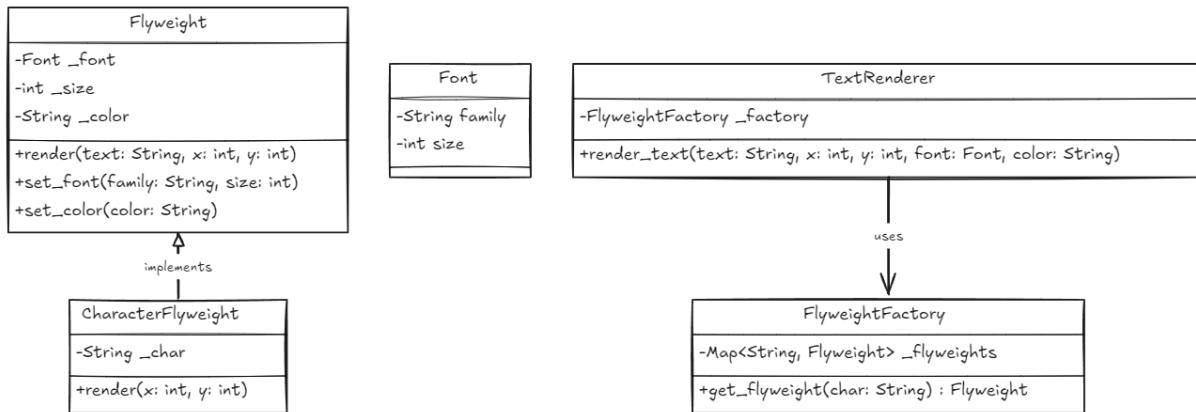


AI ที่ใช้: Gemini 1.5 Pro

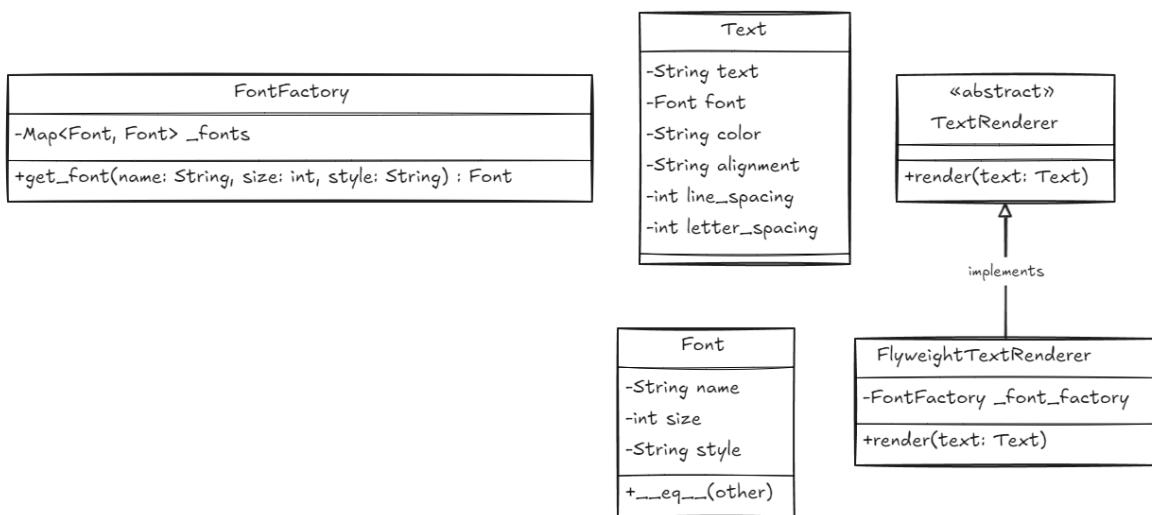
Design pattern: Flyweight

รอบที่ 1

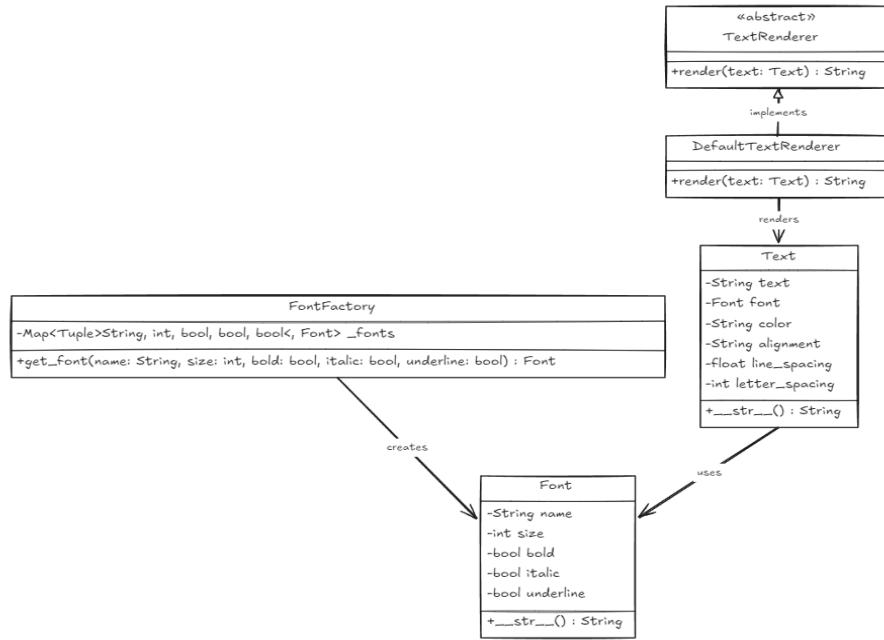
Diagram



รอบที่ 2



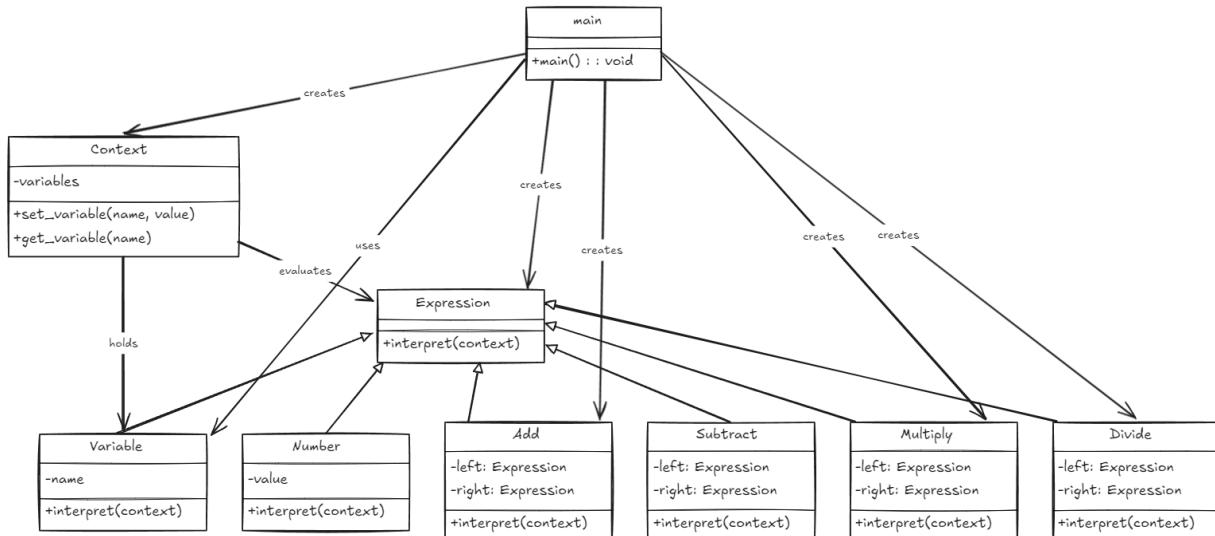
รอบที่ 3



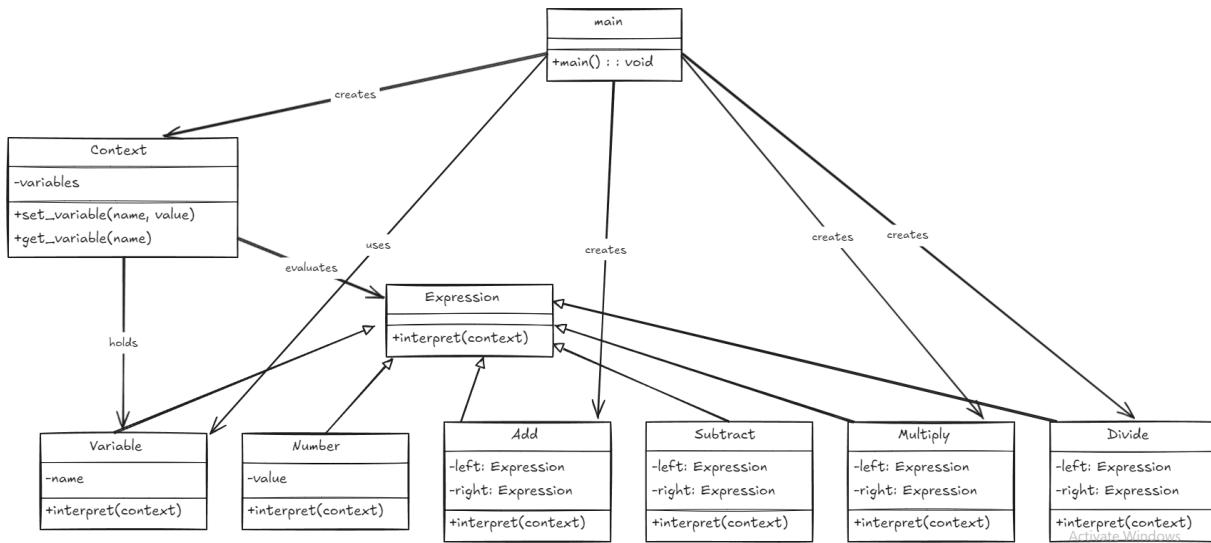
AI ที่ใช้: Gemini 1.5 Pro

Design pattern: Interpreter

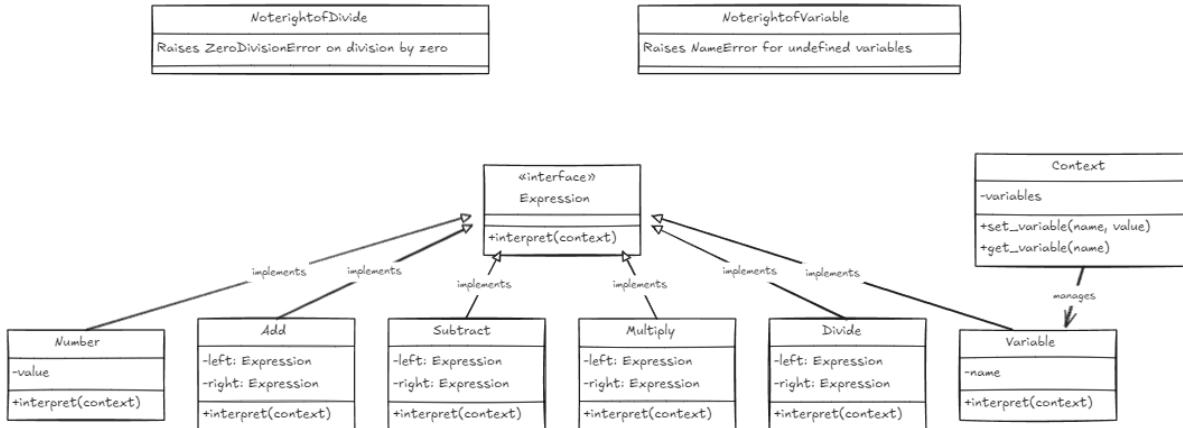
รอบที่ 1



รูปที่ 2



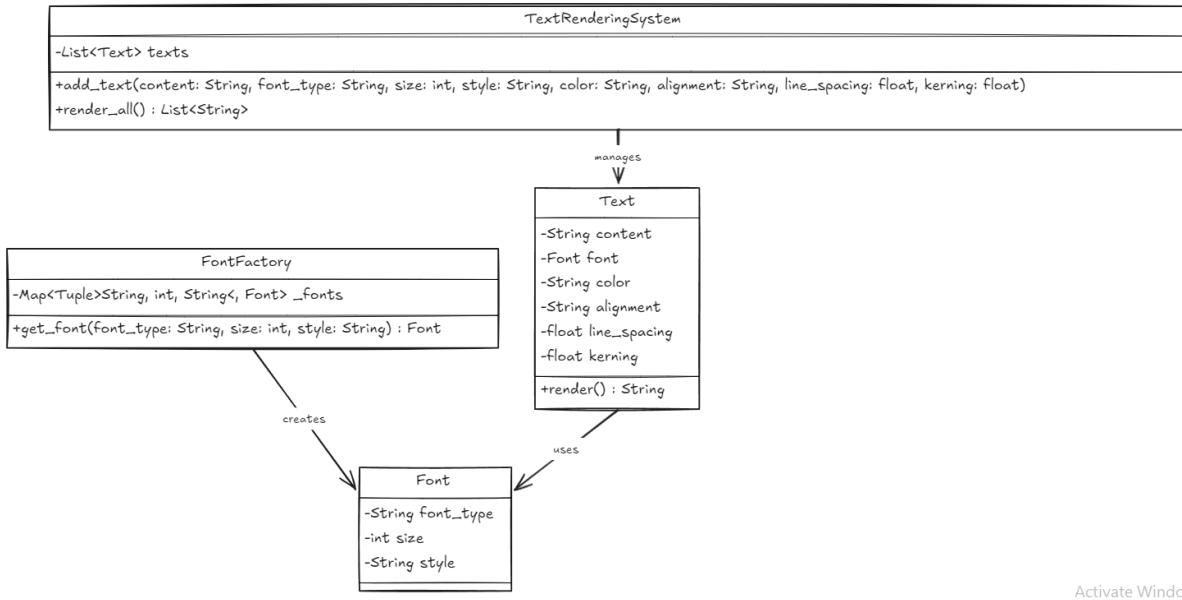
รูปที่ 3



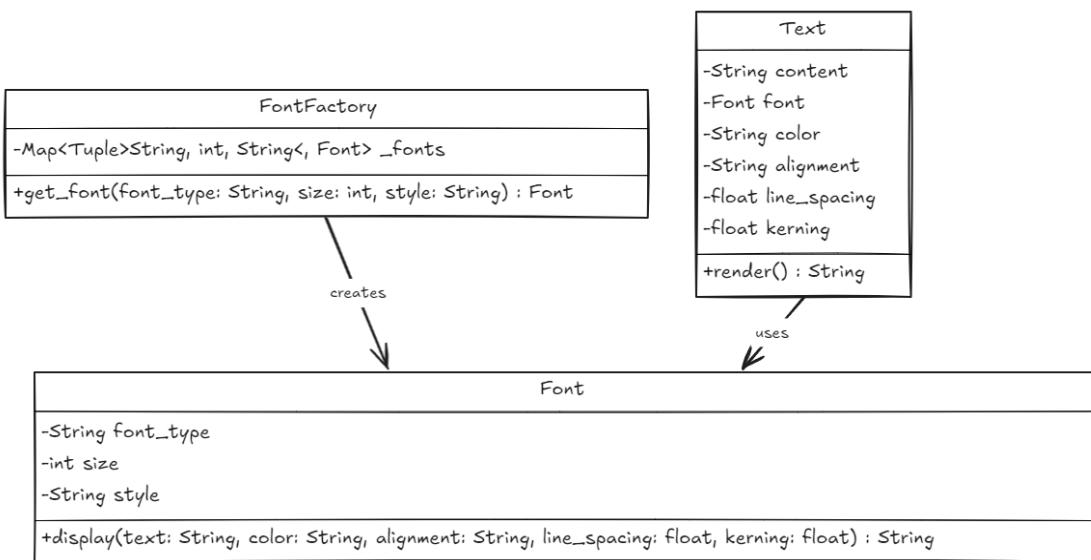
AI ที่ใช้: GithubCopilot

Design pattern: Flyweight

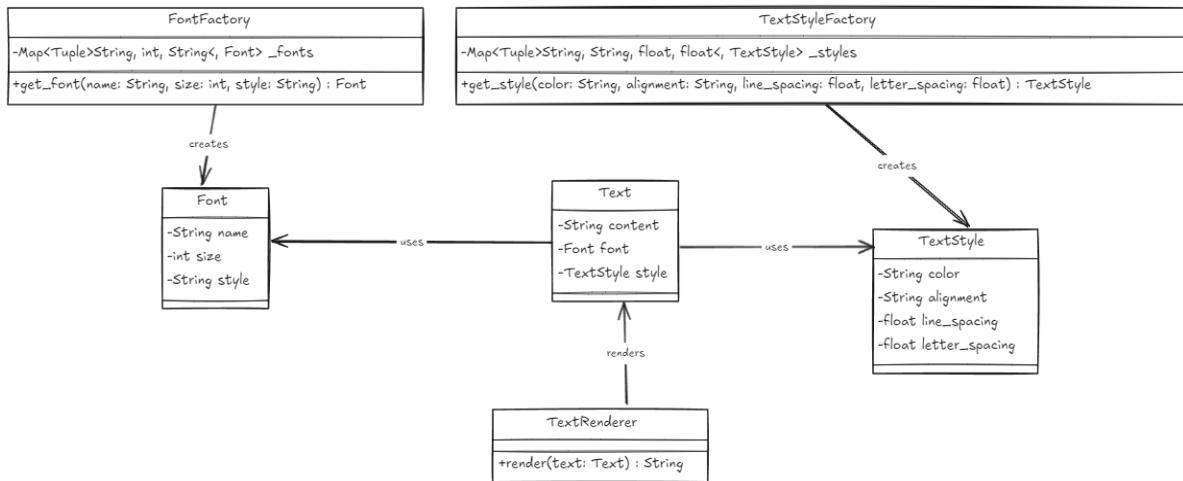
รอบที่ 1



รอบที่ 2



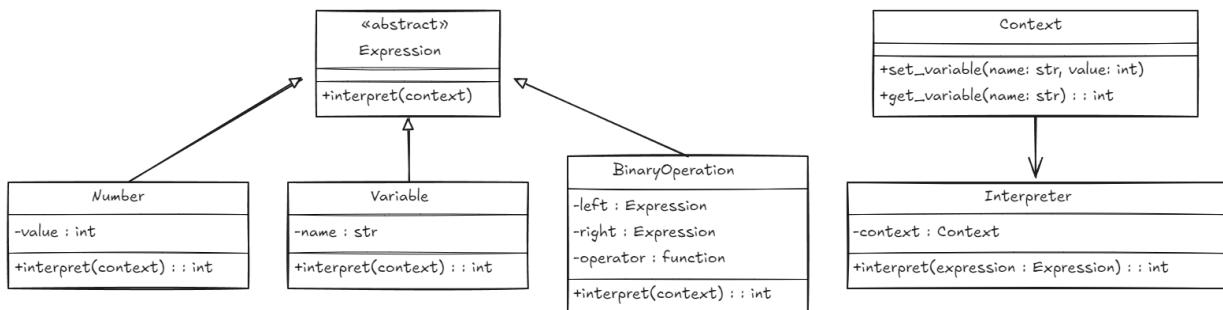
รอบที่ 3



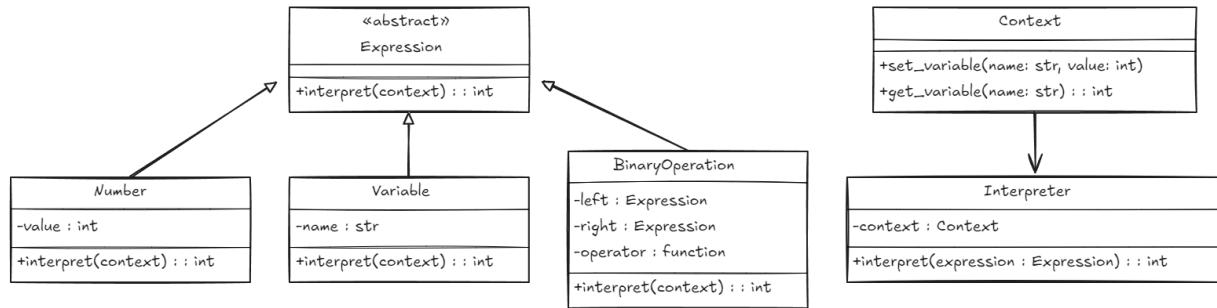
AI ที่ใช้: Github Copilot

Design pattern: Interpreter

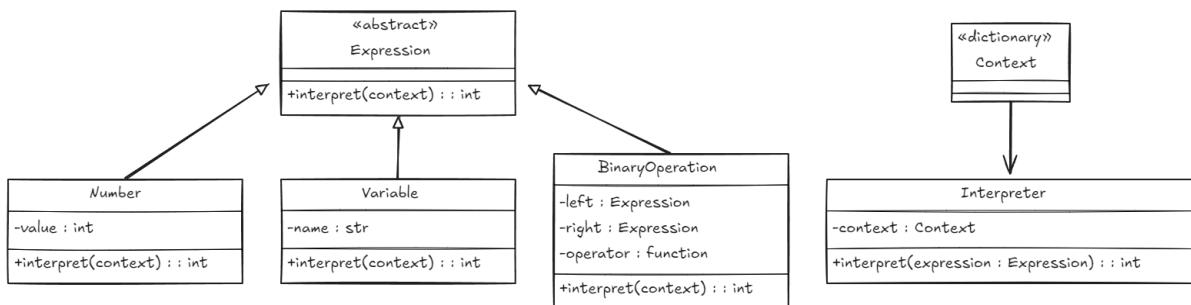
รอบที่ 1



รูปที่ 2



รูปที่ 3

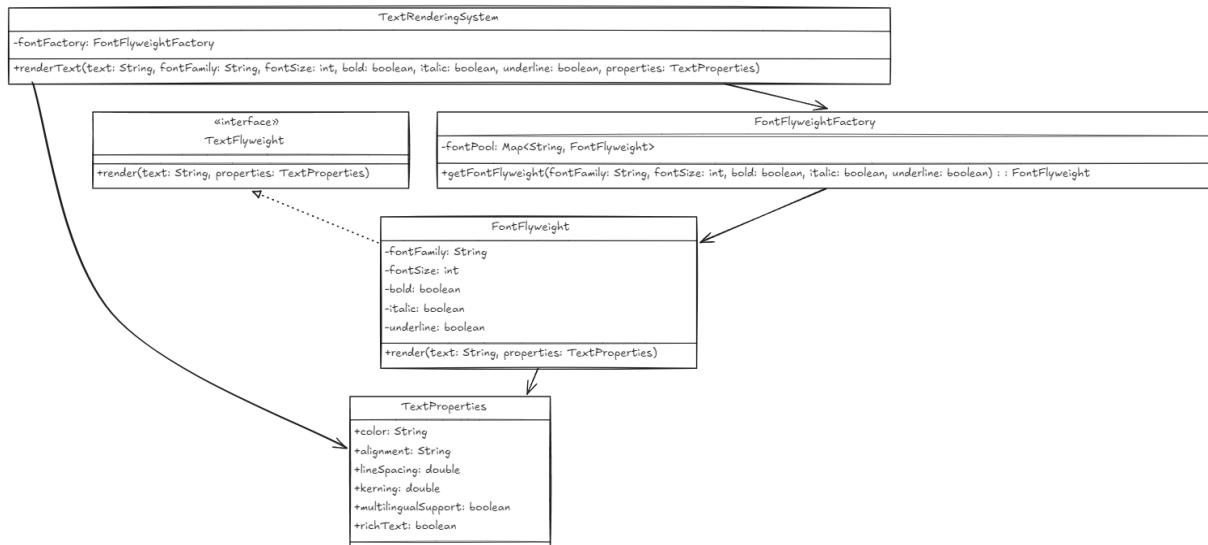


ภาษาโปรแกรมที่ใช้: Java

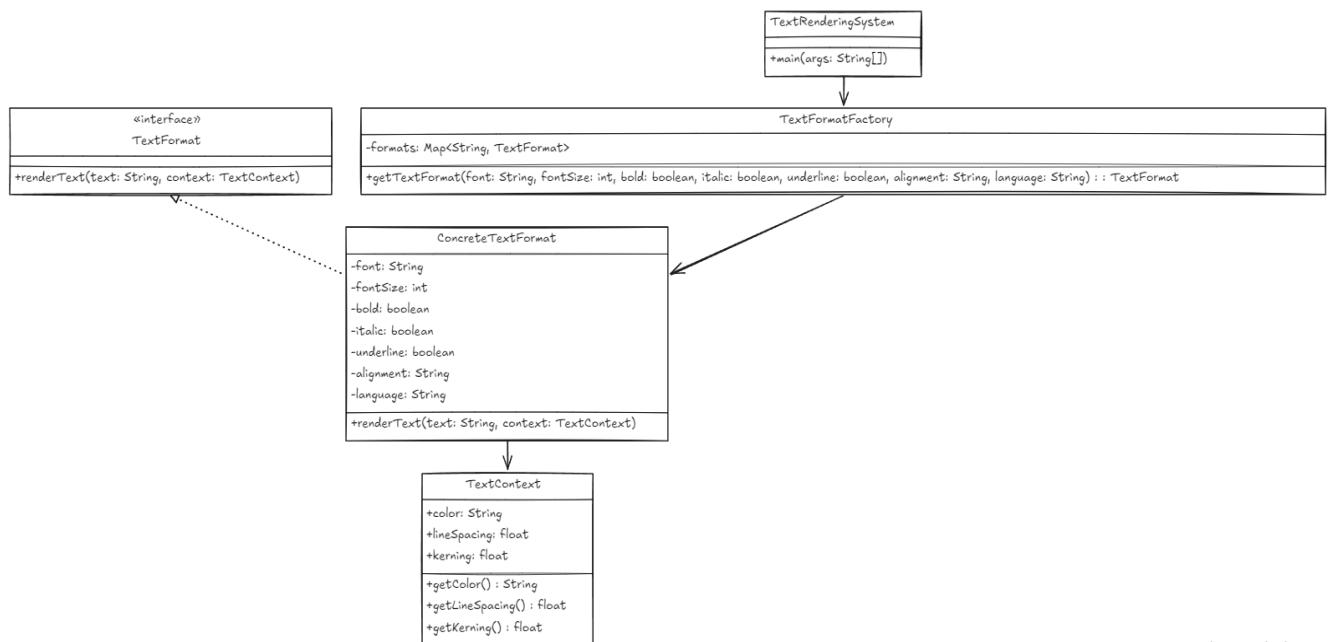
AI ที่ใช้: ChatGPT4.0

Design pattern: Flyweight

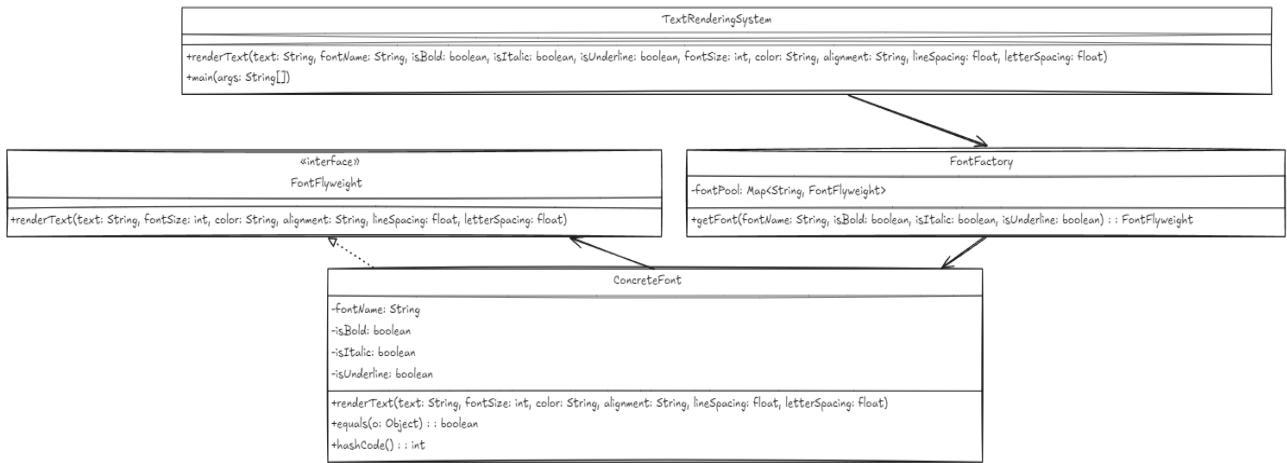
รอบที่ 1



รอบที่ 2



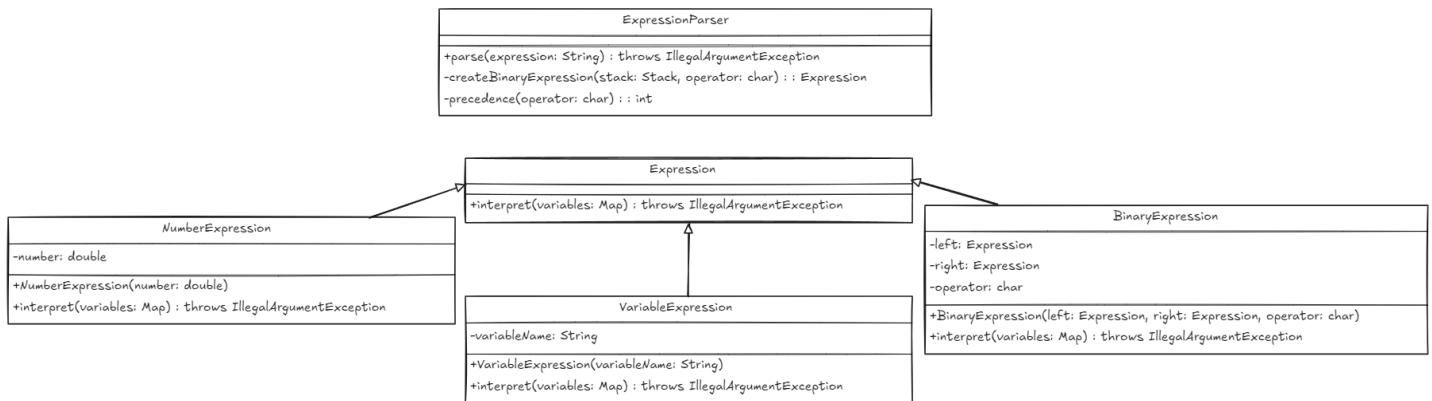
รอบที่ 3



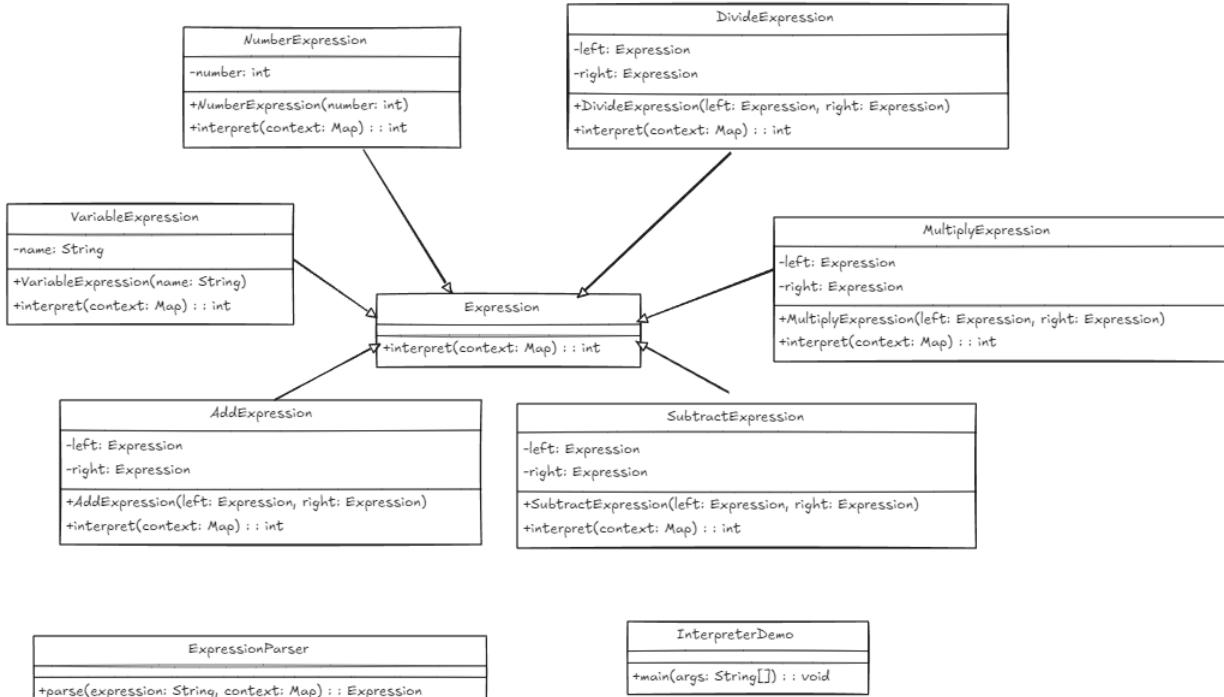
AI ที่ใช้: ChatGpt4.0

Design pattern: Interpreter

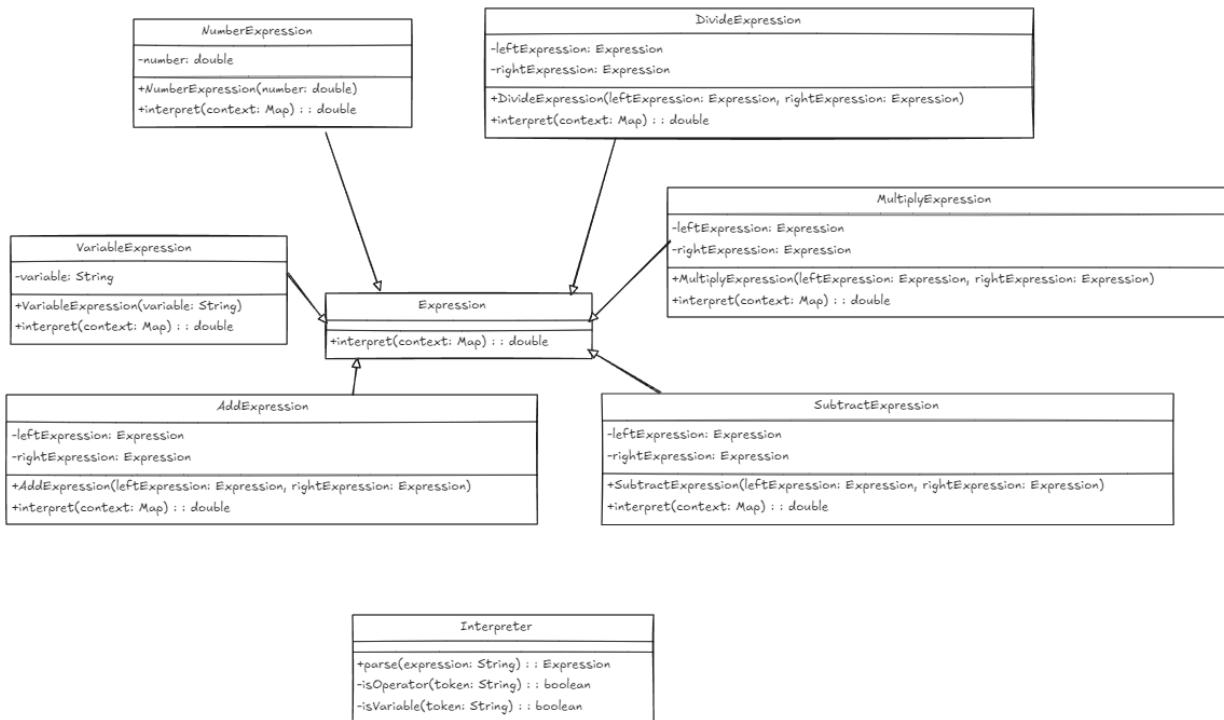
รอบที่ 1



รอบที่ 2



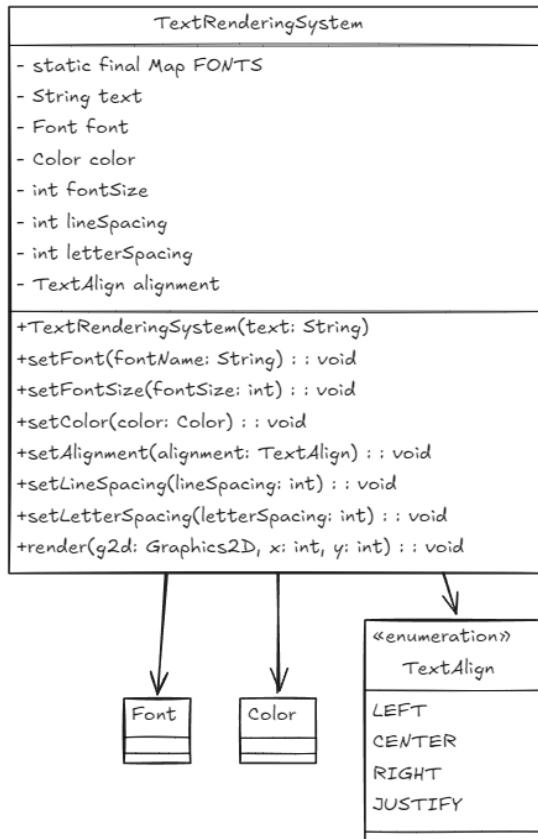
รอบที่ 3



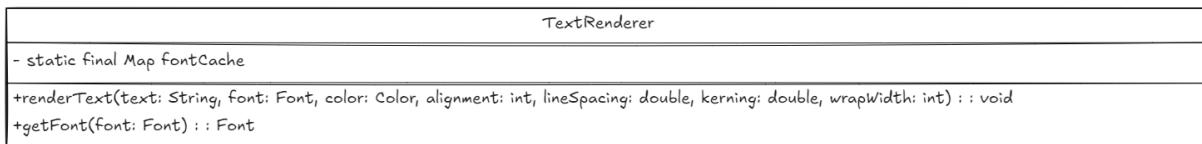
AI ที่ใช้: Gemini 1.5 Flash

Design pattern: Flyweight

รอบที่ 1



รอบที่ 2



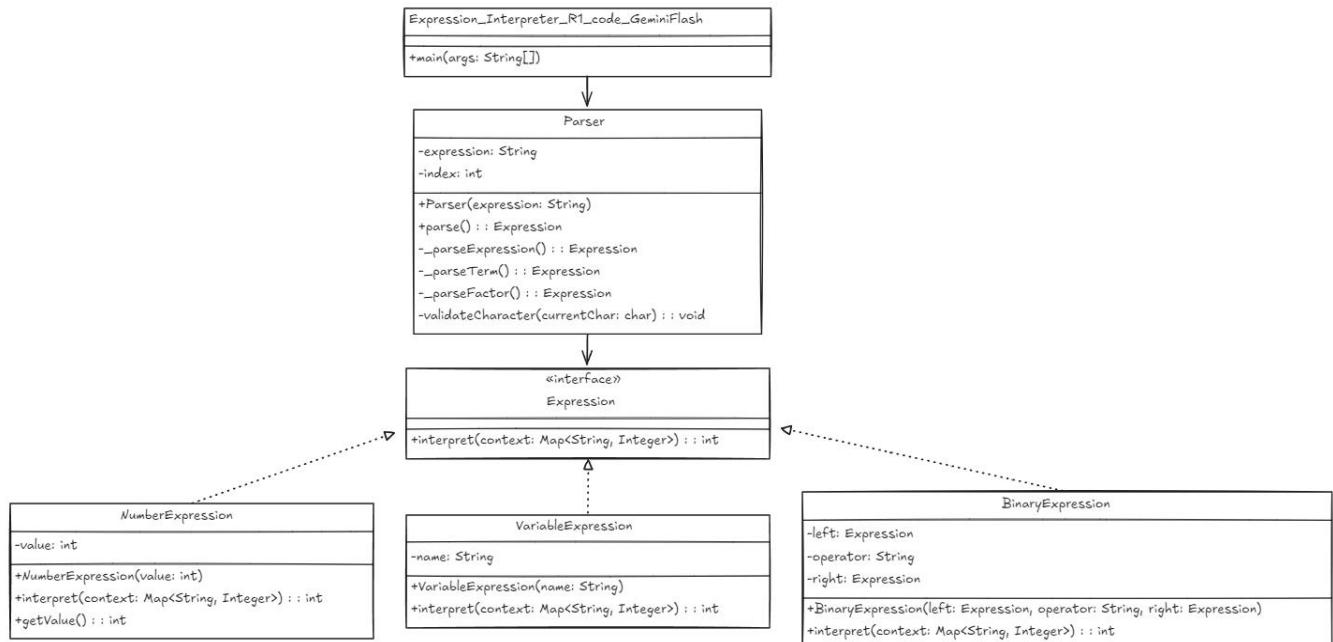
รอบที่ 3

TextRenderer
- static final Map FONT_CACHE
- static final Map COLOR_CACHE
+renderText(text: String, font: Font, color: Color, alignment: int, lineSpacing: double, letterSpacing: double, language: String, richText: boolean) : void

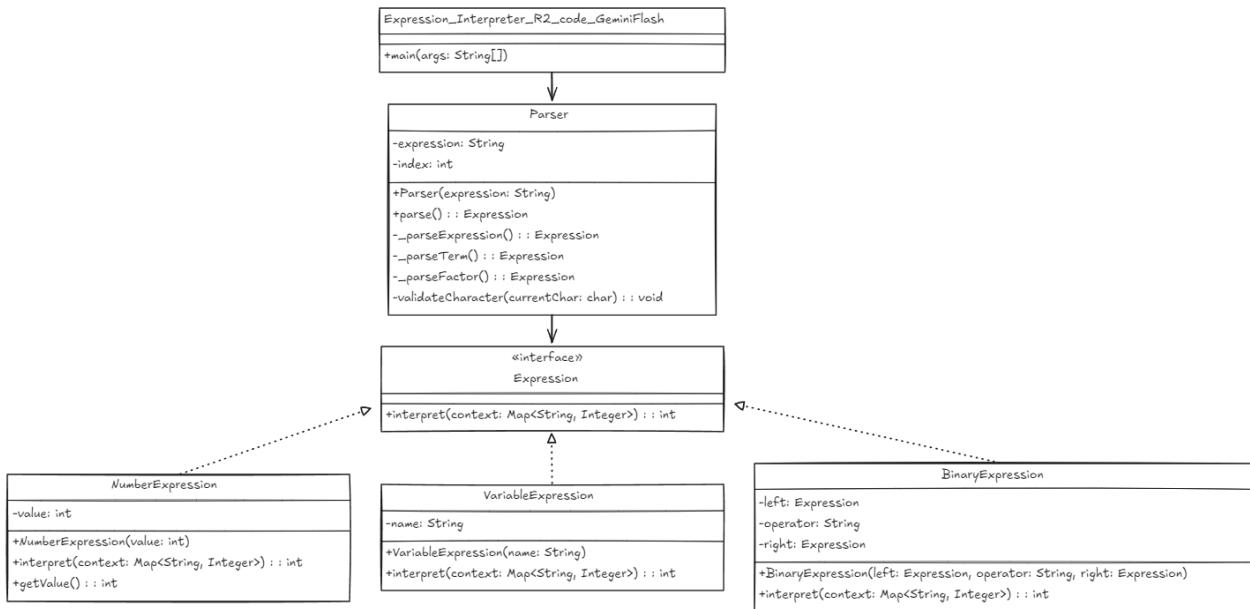
AI ที่ใช้: Gemini 1.5 Flash

Design pattern: Interpreter

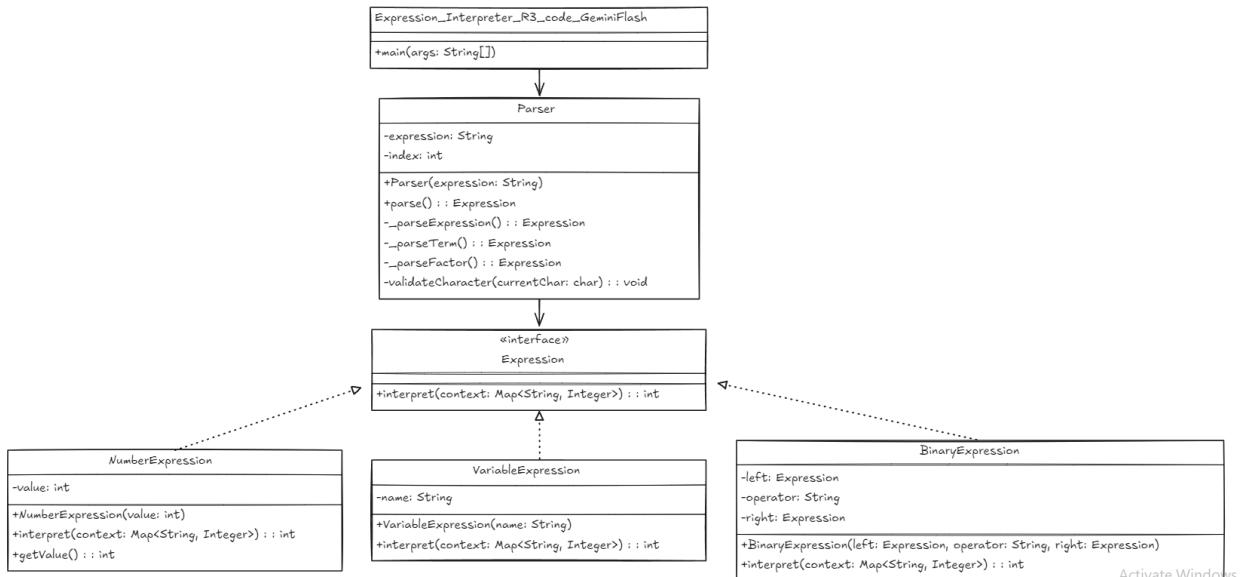
รอบที่ 1



รอบที่ 2



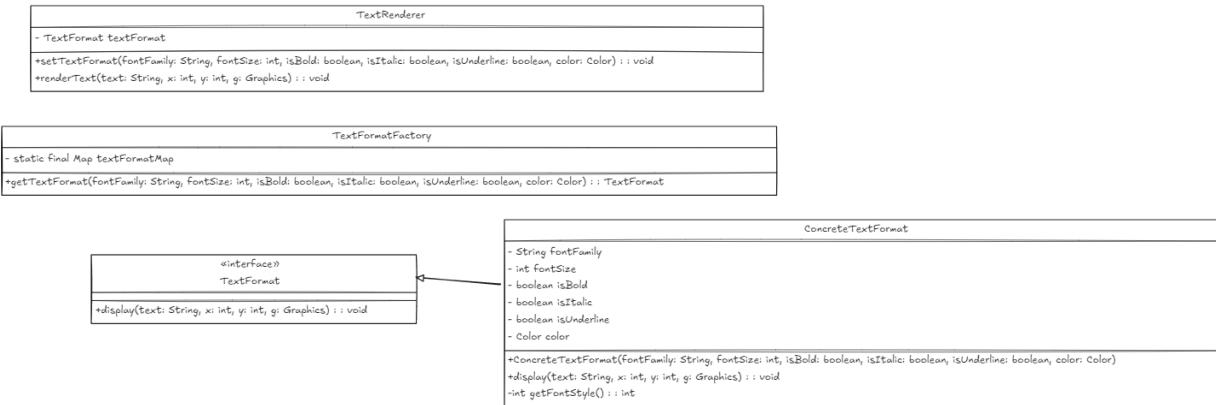
รอบที่ 3



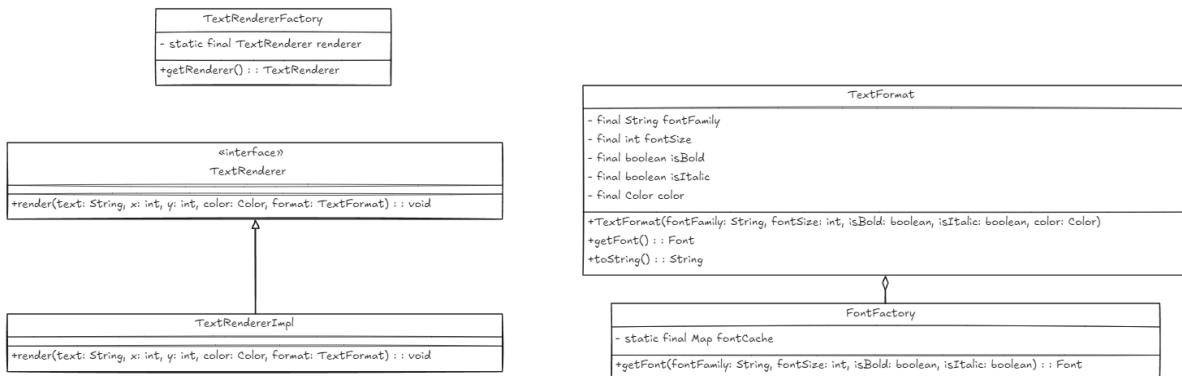
AI ที่ใช้: Gemini 1.5 Pro

Design pattern: Flyweight

รอบที่ 1

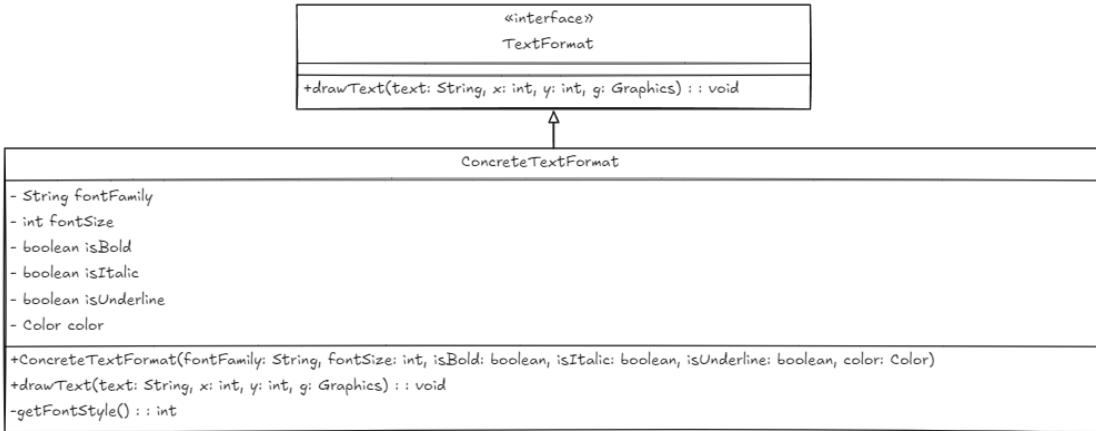


รอบที่ 2



รอบที่ 3

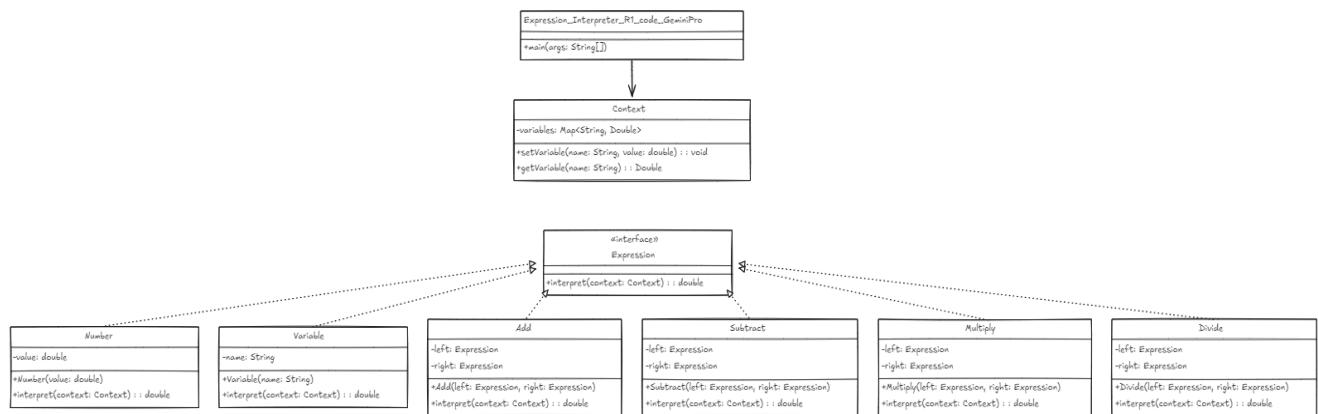
TextFormatFactory
- static final HashMap textFormatMap
+getTextFormat(fontFamily: String, fontSize: int, isBold: boolean, isItalic: boolean, isUnderline: boolean, color: Color) : TextFormat



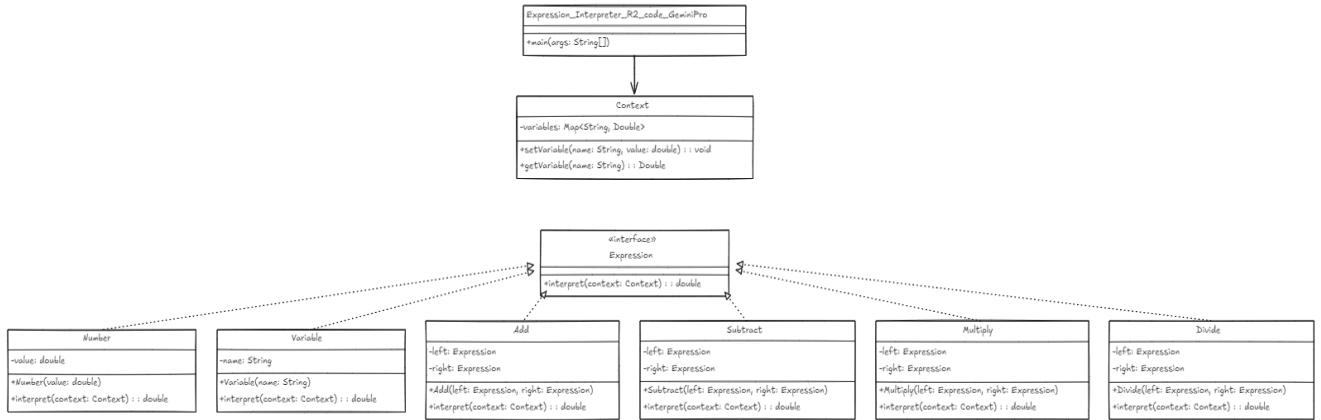
AI ที่ใช้: Gemini 1.5 Pro

Design pattern: Interpreter

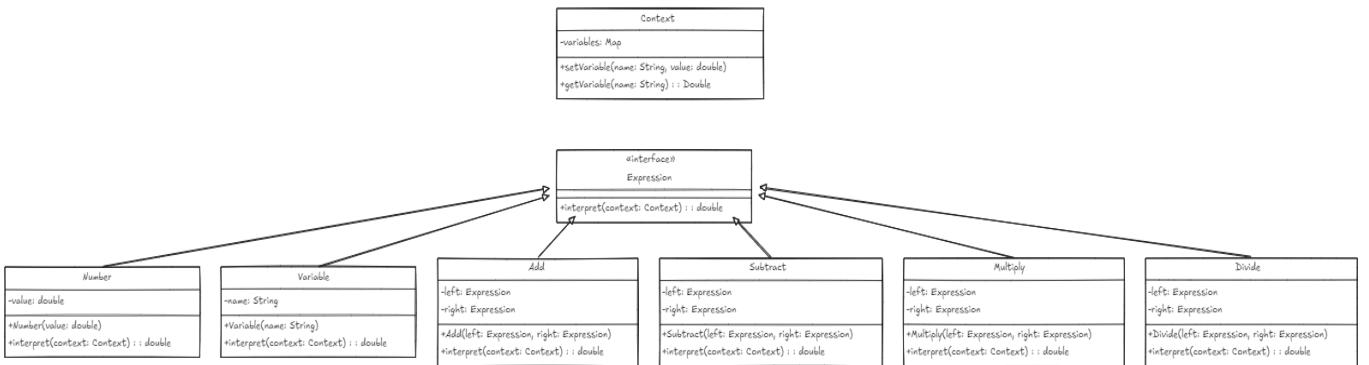
รอบที่ 1



รอบที่ 2



รอบที่ 3

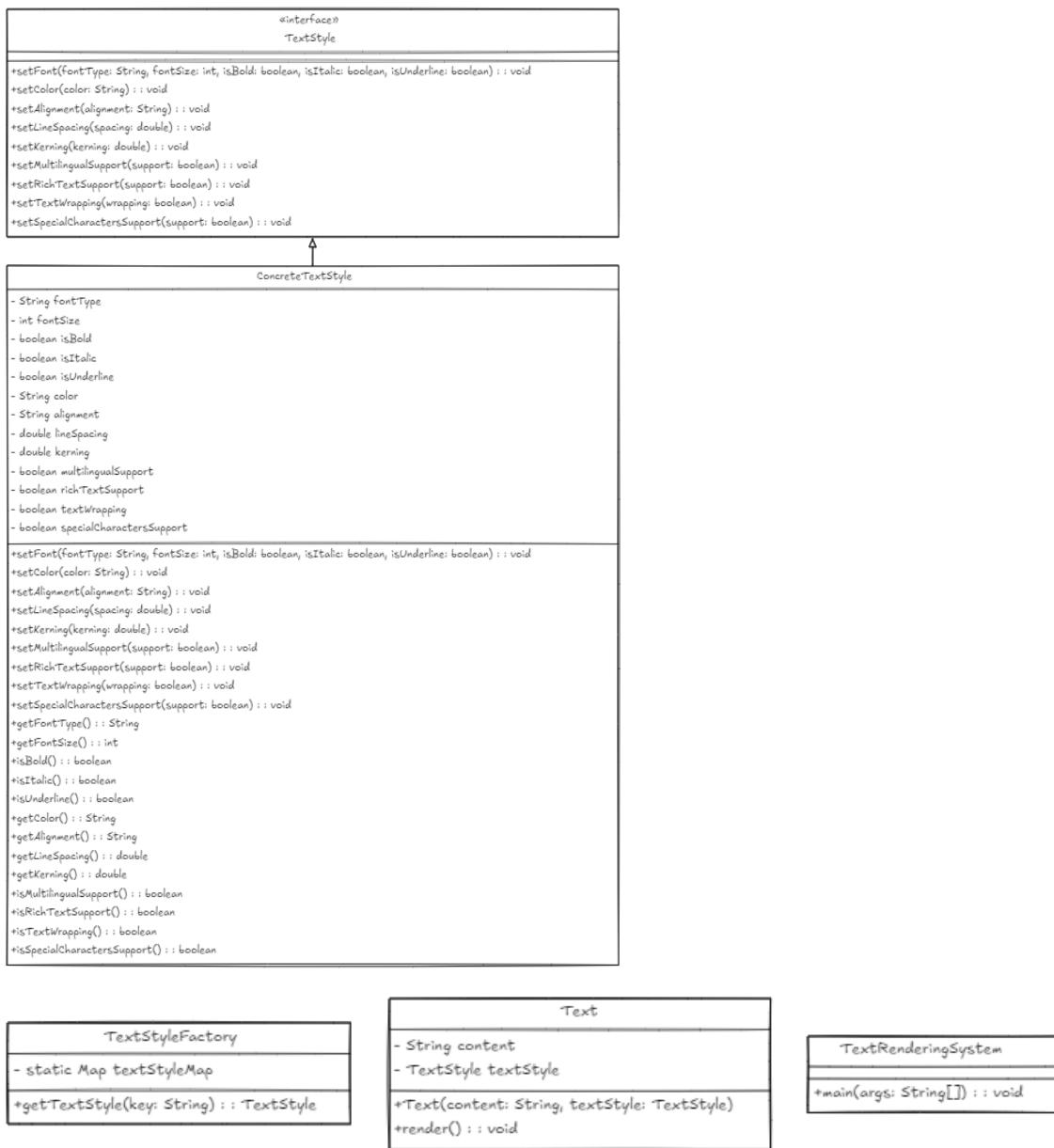


AI ที่ใช้: Github Copilot

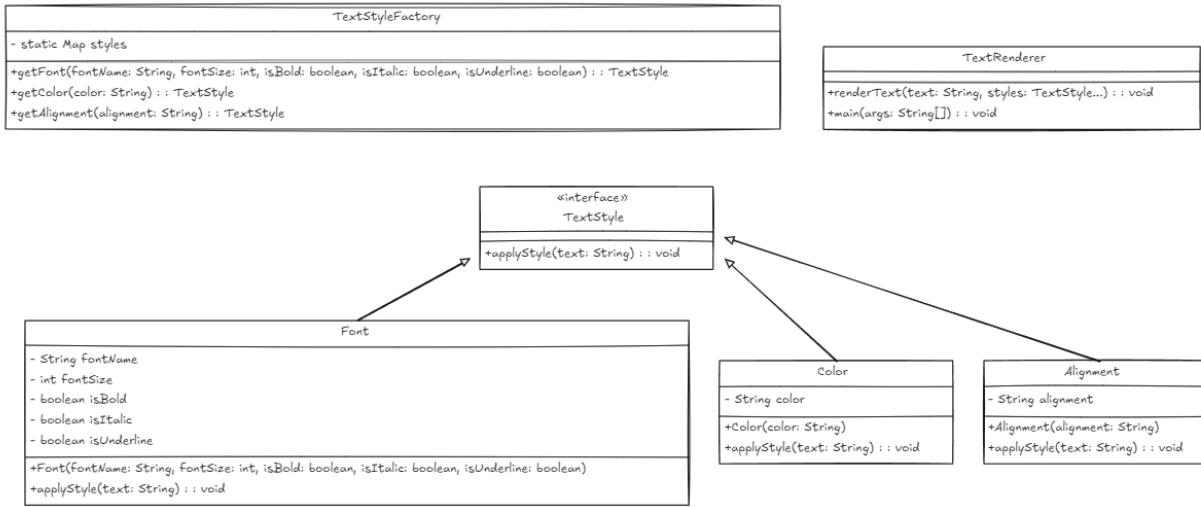
Design pattern: Flyweight

รอบที่ 1

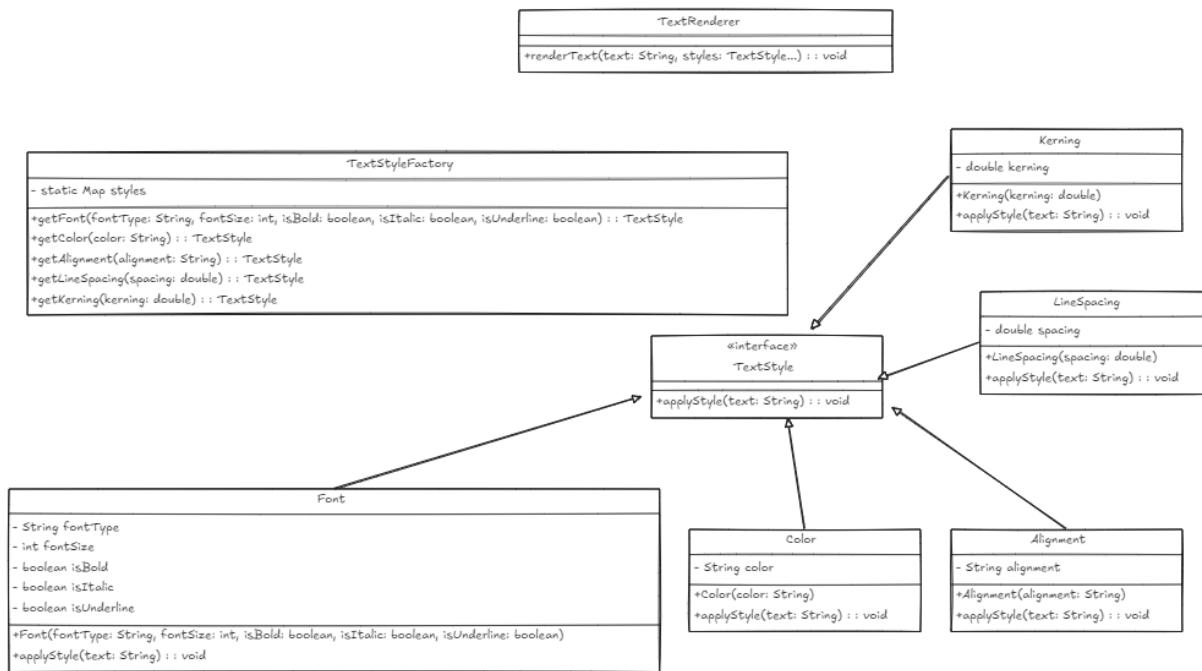
Diagram



รอบที่ 2



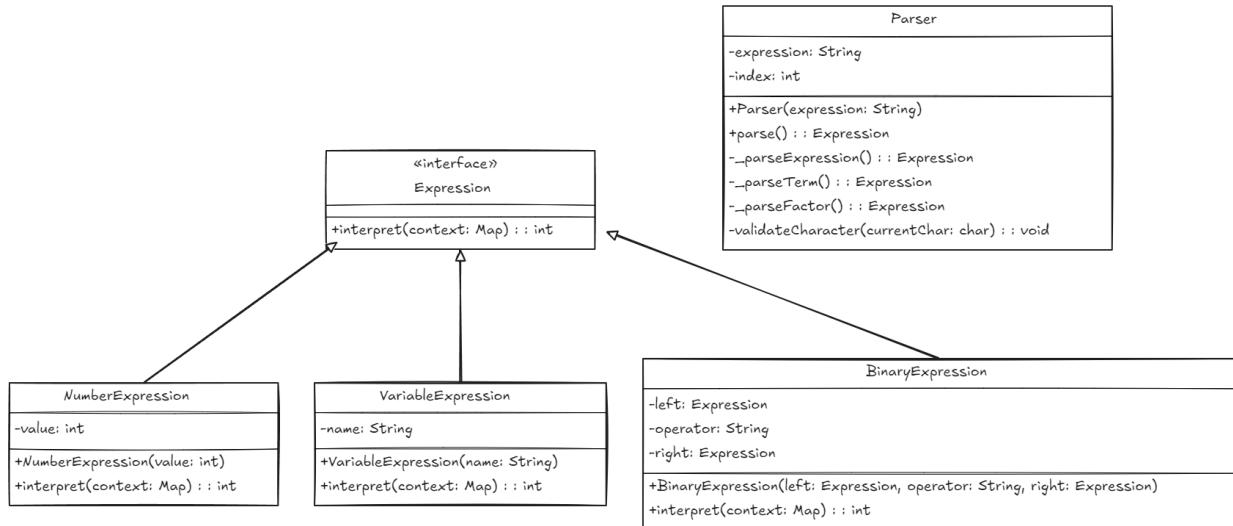
รอบที่ 3



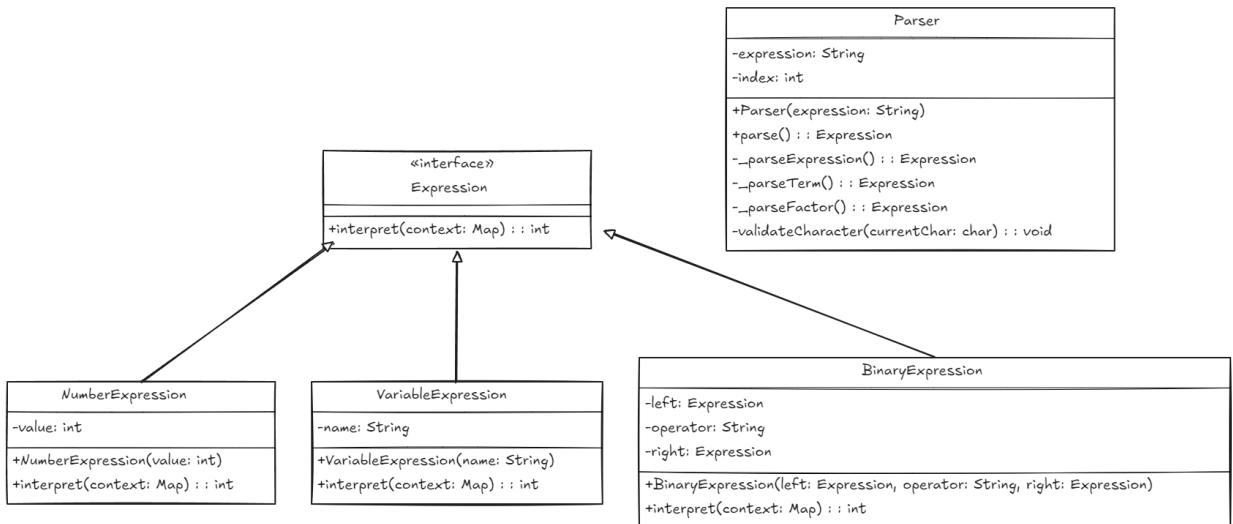
AI ที่ใช้: Github Copilot

Design pattern: Interpreter

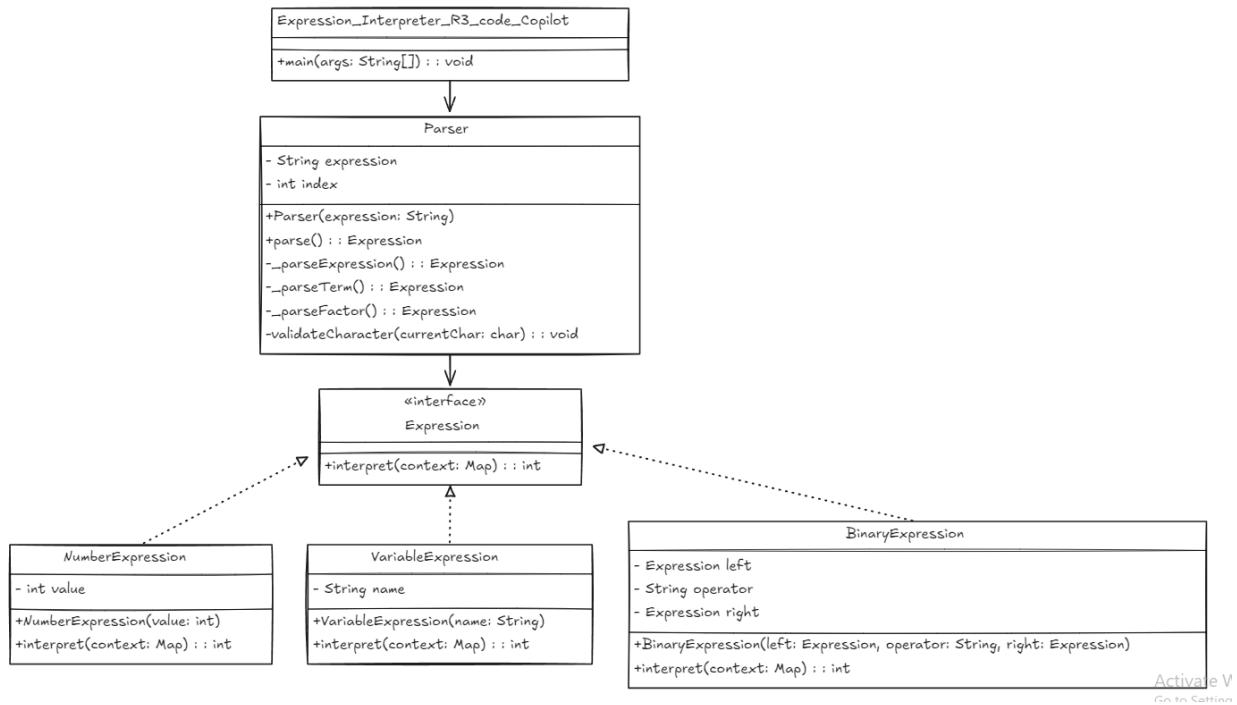
รอบที่ 1



รอบที่ 2



รอบที่ 3



ผลการทดสอบการนำ Code ไป Run

ภาษาโปรแกรมที่ใช้	Generative AI	Design Pattern	รอบที่	requirement	ภาษาโปรแกรมและ Design pattern	ทดสอบ Run	ผลลัพธ์
Java	ChatGPT4.0	Flyweight	1	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
				ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
			3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
	Interpreter		1	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
				ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง

		3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
Gemini 1.5Flash	Flyweight	1	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
		2	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
		3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
	Interpreter	1	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ไม่ถูกต้อง เพราะว่า การจัดลำดับของการประมวลผล (operator precedence) ไม่ถูกต้อง ทำให้การประมวลผลของการคำนวณไม่ถูกต้องได้

			2	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ไม่ถูกต้อง เพราะว่า การจัดลำดับของการประมวลผล (operator precedence) ไม่ถูกต้อง ทำให้การประมวลผลของการคำนวณไม่ถูกต้องได้
			3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ไม่ถูกต้อง เพราะว่า การจัดลำดับของการประมวลผล (operator precedence) ไม่ถูกต้อง ทำให้การประมวลผลของการคำนวณไม่ถูกต้องได้
Gemini 1.5 Pro	Flyweight	1	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง	
		2	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง	

			3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
			1	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
	Interpreter	2		ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
		3		ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
Github Copilot	Flyweight	1		ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
		2		ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design	สามารถ Run ได้	ถูกต้อง

			pattern ที่ถูกต้อง		
	3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
Interpreter	1	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ไม่ถูกต้อง เพราะว่า การจัดลำดับของการประมวลผล (operator precedence) ไม่ถูกต้อง ทำให้การประมวลผลของการคำนวณไม่ถูกต้องได้
	2	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ไม่ถูกต้อง เพราะว่า การจัดลำดับของการประมวลผล (operator precedence) ไม่ถูกต้อง ทำให้การประมวลผลของการคำนวณไม่ถูกต้องได้
	3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design	สามารถ Run ได้	ไม่ถูกต้อง เพราะว่า การจัดลำดับของการประมวลผล (operator

					pattern ที่ถูกต้อง		precedence) ไม่ ถูกต้อง ทำให้การประมวลผล ของการคำนวณไม่ ถูกต้องได้
Python	ChatGPT4.0	Flyweight	1	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
			2	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
			3	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
	Interpreter	Interpreter	1	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ไม่ถูกต้อง คำนวณไม่ ถูกต้อง
			2	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design	สามารถ Run ได้	ถูกต้อง

				pattern ที่ถูกต้อง		
		3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ไม่ถูกต้อง คำนวนไม่มีถูกต้อง
Gemini 1.5Flash	Flyweight	1	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง แต่ไม่แสดง Font ที่ใช้
		2	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
		3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
	Interpreter	1	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ไม่ถูกต้อง เพราะว่า การจัดลำดับของการประมวลผล (operator precedence) ไม่ถูกต้อง ทำให้การประมวลผล

						ของการคำนวณไม่ ถูกต้องได้
		2	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ไม่ถูกต้อง เพราะว่า การจัดลำดับของการ ประมวลผล (operator precedence) ไม่ ถูกต้อง ทำให้การประมวลผล ของการคำนวณไม่ ถูกต้องได้
		3	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ไม่ถูกต้อง เพราะว่า การจัดลำดับของการ ประมวลผล (operator precedence) ไม่ ถูกต้อง ทำให้การประมวลผล ของการคำนวณไม่ ถูกต้องได้
Gemini 1.5 Pro	Flyweight	1	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
		2	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design	โค้ดมี TypeError: unhashable type แก้ได้ด้วยการ เพิ่ม	ถูกต้อง

				pattern ที่ถูกต้อง	ฟังก์ชัน hash เข้าไป ใน class Font	
		3	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
		1	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
	Interpreter	2	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
		3	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
Github Copilot	Flyweight	1	ตรง requirement ที่ กำหนด	ใช้ภาษา โปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง

			2	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
			3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง
		1	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง	
	Interpreter	2	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง	
		3	ตรง requirement ที่กำหนด	ใช้ภาษาโปรแกรมและ Design pattern ที่ถูกต้อง	สามารถ Run ได้	ถูกต้อง	

ผลการทดสอบ การ Run Test Code

ภาษา โปรแกรมที่ ใช้	Generative AI	Design Pattern	รอบ ที่	ทดสอบการ Run	คุณภาพ
Java ChatGPT4.0	Flyweight	Flyweight	1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	35/41 Statements covered (85.37%) 2/2 Branch covered (100%)
			2	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	28/37 Statements covered (75.68%) 2/2 Branch covered (100%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	20/30 Statements covered (66.67%) 2/16 Branch covered (12.50%)
	Interpreter	Interpreter	1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	64/67 Statements covered (95.52%) 42/50 Branch covered (84.00%)
			2	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	58/67 Statements covered (86.57%) 11/11 Branch covered (100%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	54/55 Statements covered (98.18%) 23/23 Branch covered (100%)

			1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	25/27 Statements covered (92.59%) 5/6 Branch covered (83.33%)
Gemini	Flyweight	2			33/34 Statements covered (97.06%) 9/10 Branch covered (90.00%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	15/16 Statements covered (93.75%) 15/18 Branch covered (83.33%)
			1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	74/94 Statements covered (78.94%) 48/59 Branch covered (81.36%)
1.5Flash	Interpreter	2			74/94 Statements covered (78.94%) 48/59 Branch covered (81.36%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	74/94 Statements covered (78.94%) 48/59 Branch covered (81.36%)
			1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	27/29 Statements covered (93.10%) 4/6 Branch covered (66.67%)
	Gemini 1.5 Pro	Flyweight	1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	27/29 Statements covered (93.10%) 4/6 Branch covered (66.67%)

			2	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	25/28 Statements covered (89.29%) 6/6 Branch covered (100%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	27/42 Statements covered (64.29%) 4/6 Branch covered (66.67%)
Interpreter	2	Interpreter	1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	39/49 Statements covered (79.59%) 4/4 Branch covered (100%)
			2	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	39/49 Statements covered (79.59%) 4/4 Branch covered (100%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	39/49 Statements covered (79.59%) 4/4 Branch covered (100%)
Github Copilot	Flyweight	Flyweight	1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	42/63 Statements covered (66.67%) 2/2 Branch covered (100%)
			2	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	35/42 Statements covered (83.33%) 8/8 Branch covered (100%)

			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	53/54 Statements covered (98.15%) 12/12 Branch covered (100%)
Interpreter			1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	71/93 Statements covered (76.34%) 46/59 Branch covered (77.97%)
			2	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	71/93 Statements covered (76.34%) 46/59 Branch covered (77.97%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	71/93 Statements covered (76.34%) 46/59 Branch covered (77.97%)
Python	ChatGPT4.0	Flyweight	1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	28/28 Statements covered (100%) 4/4 Branch covered (100%)
			2	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	43/44 Statements covered (97.72%) 5/6 Branch covered (83.33%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	42/48 Statements covered (87.5%) 7/8 Branch covered (87.5%)

			1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	28/28 Statements covered (100%) 4/4 Branch covered (100%)
Gemini 1.5Flash	Interpreter	2			43/44 Statements covered (96%) 7/8 Branch covered (88%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	42/48 Statements covered (88%) 7/8 Branch covered (88%)
			1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	31/36 Statements covered (86.1%) 3/4 Branch covered (75%)
	Flyweight	2			35/36 Statements covered (97.22%) 4/4 Branch covered (100%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	84/88 Statements covered (95.45%) 19/22 Branch covered (86.36%)
			1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	82/96 Statements covered (85.42%) 31/34 Branch covered (91.18%)

			2	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	82/96 Statements covered (85.42%) 31/34 Branch covered (91.18%)
			3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	94/108 Statements covered (87.04%) 32/36 Branch covered (88.89%)
Gemini 1.5 Pro	Flyweight	1			43/44 Statements covered (97.72%) 6/6 Branch covered (100%)
			2	เกิด ModuleNotFoundError แก้ไขได้โดยง่าย โดยเปลี่ยนชื่อ module ให้เป็นชื่อโค้ด	29/34 Statements covered (85.29%) 2/4 Branch covered (50%)
			3	เกิด ModuleNotFoundError แก้ไขได้โดยง่าย โดยเปลี่ยนชื่อ module ให้เป็นชื่อโค้ด	39/51 Statements covered (76.47%) 10/14 Branch covered (71.43%)
	Interpreter	1		สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	52/62 Statements covered (83.87%) 7/8 Branch covered (87.5%)
		2		สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	52/62 Statements covered (83.87%) 7/8 Branch covered (87.5%)

		3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	52/60 Statements covered (86.67%) 7/8 Branch covered (87.5%)
Github	Flyweight	1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	32/32 Statements covered (100%) 4/4 Branch covered (100%)
		2	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	25/25 Statements covered (100%) 3/4 Branch covered (75%)
		3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	35/35 Statements covered (100%) 8/8 Branch covered (100%)
Copilot	Interpreter	1	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	32/44 Statements covered (72.72%) 5/6 Branch covered (83.33%)
		2	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	32/44 Statements covered (72.72%) 5/6 Branch covered (83.33%)
		3	สามารถ Run ได้ (เพิ่ม การ import module ไปใช้กับไฟล์ Test)	32/44 Statements covered (72.72%) 5/6 Branch covered (83.33%)

คุณภาพของ Code และ Test code

คำอธิบายของเมตริก:

1. Cyclomatic Complexity:

- เป็นเมตริกที่วัดจำนวนเส้นทางที่สามารถเลือกได้ในโค้ดของฟังก์ชัน ยิ่งมีค่าเยอะ โค้ดก็จะยิ่งซับซ้อน และทำให้ยากต่อการทดสอบและบำรุงรักษา
- เกณฑ์: 10 (ถ้าค่าสูงกว่า 10 อาจถือว่าโค้ดซับซ้อนเกินไป)

2. Lines of Code (LOC):

- วัดจำนวนบรรทัดของโค้ดในฟังก์ชัน โดยไม่นับรวมคอมเมนต์และบรรทัดว่าง ฟังก์ชันที่มีบรรทัดมากเกินไปมักจะยากต่อการเข้าใจและบำรุงรักษา
- เกณฑ์: 50 (ถ้าบรรทัดยาวเกิน 50 อาจจะทำให้โค้ดยากต่อการจัดการ)

3. Parameter Count:

- เป็นจำนวนพารามิเตอร์ที่ฟังก์ชันรับเข้ามา ฟังก์ชันที่มีพารามิเตอร์มากเกินไปอาจจะยากต่อการเข้าใจและทดสอบ
- เกณฑ์: 4 (ถ้ามีจำนวนพารามิเตอร์มากกว่า 4 อาจบ่งบอกว่าฟังก์ชันทำงานมากเกินไป)

ภาษาโปรแกรมที่ใช้ : Python

AI: Chatgpt4.0

Design pattern: Flyweight

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
__init__	5	9	1	5	▲ 5
render	11	25	1	14	▲ 6
get_font	32	36	2	5	▲ 5
__init__	40	41	1	2	1
render_text	43	45	1	3	▲ 10

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
text_renderer	6	7	1	2	0
test_render_text_basic	9	21	1	13	1
test_render_text_multilingual	23	35	1	13	1
test_render_text_special_characters	37	49	1	13	1
test_font_factory_reuse	51	55	1	4	1
test_font_factory_new_instance	57	61	1	4	1

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
__init__	5	9	1	5	▲ 5
__repr__	11	12	1	2	1
get_font	19	23	2	5	▲ 5
__init__	27	33	1	7	▲ 7
render	35	37	1	3	1
__init__	42	43	1	2	1
add_text	45	46	1	2	▲ 10
render	51	52	2	2	1
render	57	60	2	4	1
render	65	66	1	2	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
test_font_factory	5	8	1	4	0
test_text_rendering	11	16	1	6	0
test_multilingual_text	19	23	1	5	0
test_unicode_text	26	30	1	5	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
__init__	5	10	1	6	▲ 6
__repr__	12	13	1	2	1
__init__	16	17	1	2	2
__repr__	19	20	1	2	1
__init__	24	26	1	3	1
get_font	28	32	2	5	▲ 6
get_color	34	37	2	4	2
__init__	41	43	1	3	2
add_text	45	55	1	11	▲ 11
render	57	61	2	5	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
factory	5	6	1	2	0
renderer	9	10	1	2	1
test_font_management	12	17	1	6	1
test_color_management	19	24	1	6	1
test_text_rendering	26	33	1	8	1
test_multilingual_support	35	40	1	6	1
test_alignment	42	51	1	10	1
test_line_spacing	53	56	1	4	1
test_letter_spacing	58	61	1	4	1

AI: Chatgpt

Design pattern: Interpreter

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
interpret	5	6	1	2	2
__init__	10	11	1	2	2
interpret	13	14	1	2	2
__init__	18	19	1	2	2
interpret	21	24	2	4	2
__init__	35	38	1	4	4
interpret	40	47	3	6	2
__init__	51	52	1	2	2
interpret	54	55	1	2	2
__init__	59	61	1	3	2
parse	63	64	1	2	1
_parse_expression	66	81	4	15	1
_get_next_token	83	86	1	4	1
interpret	89	99	3	10	2

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
test_basic_operations	3	7	1	5	0
test_parentheses_operations	9	12	1	4	0
test_variable_operations	14	18	1	5	0
test_error_handling	20	23	1	4	0
test_nested_expressions	25	27	1	3	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
interpret	4	5	1	2	2
__init__	8	9	1	2	2
interpret	11	12	1	2	2
__init__	15	16	1	2	2
interpret	18	21	2	4	2
__init__	24	27	1	4	4
interpret	29	35	2	7	2
__init__	38	39	1	2	3
__init__	42	43	1	2	3
__init__	46	47	1	2	3
__init__	50	51	1	2	3
__init__	54	56	1	3	2
parse	58	60	1	3	1
tokenize	62	64	1	3	2
parse_expression	66	67	1	2	2
parse_term	69	82	5	11	2
parse_factor	84	97	5	11	2
parse_primary	99	114	5	15	2
__init__	117	121	2	5	3
evaluate	123	126	1	4	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
test_valid_expressions	12	14	1	3	3
test_invalid_expressions	22	25	1	4	2
test_variable_expressions	31	33	1	3	3
test_complex_expression	38	40	1	3	3

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
interpret	3	4	1	2	2
__init__	9	10	1	2	2
interpret	12	13	1	2	2
__init__	18	19	1	2	2
interpret	21	24	2	4	2
__init__	29	31	1	3	3
interpret	33	34	1	2	2
__init__	39	41	1	3	3
interpret	43	44	1	2	2
__init__	49	51	1	3	3
interpret	43	44	1	2	2
__init__	49	51	1	3	3
interpret	53	54	1	2	2
__init__	59	61	1	3	3
interpret	63	67	2	5	2
parse	72	74	1	3	2
tokenize	76	77	1	2	2
build_expression_tree	79	102	9	24	2

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
test_number_interpret	3	5	1	3	0
test_variable_interpret	7	10	1	4	0
test_add_interpret	12	14	1	3	0
test_subtract_interpret	16	18	1	3	0
test_multiply_interpret	20	22	1	3	0
test_divide_interpret	24	26	1	3	0
test_divide_by_zero	28	31	1	4	0
test_variable_not_defined	33	36	1	4	0
test_parser_simple_expression	38	41	1	4	0
test_parser_complex_expression	43	46	1	4	0
test_parser_variable_expression	48	52	1	5	0
test_parser_empty_expression	54	57	1	4	0
test_parser_invalid_operator	59	62	1	4	0
test_parser_unexpected_token	64	67	1	4	0

AI: Gemini 1.5 Flash

Design pattern: Flyweight

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
render	6	7	1	2	2
render	10	11	1	2	2
render	14	15	1	2	2
render	18	19	1	2	2
__init__	24	29	1	6	1
get_font	31	32	1	2	2
__init__	35	40	1	6	▲ 6
__init__	43	44	1	2	2
render	46	51	2	6	3

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
font_factory	6	7	1	2	0
text_renderer	10	11	1	2	1
test_render_sans_serif	13	15	1	3	2
test_render_serif	17	19	1	3	2
test_render_monospace	21	23	1	3	2
test_render_unknown_font	25	27	1	3	2
test_render_with_line_spacing	29	31	1	3	2

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
render	6	7	1	2	3
render	10	11	1	2	3
render	14	15	1	2	3
render	18	19	1	2	3
__init__	22	27	1	6	▲ 6
format_text	29	30	1	2	3
__init__	33	39	1	7	1
get_formatter	41	45	2	5	▲ 6
render_text	47	49	1	3	▲ 8

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
renderer	5	6	1	2	0
test_render_text_sans_serif	8	11	1	4	1
test_render_text_serif	13	16	1	4	1
test_render_text_monospace	18	21	1	4	1
test_render_text_reuse_formatter	23	28	1	6	1
test_get_formatter_cache	31	34	1	4	1
test_get_formatter_new	36	39	1	4	1

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
render	8	10	1	3	2
__init__	15	18	1	4	4
render	20	22	1	3	2
__init__	27	30	1	4	4
render	32	34	1	3	2
__init__	39	40	1	2	2
render	42	44	1	3	2
get_font	51	63	5	13	3
__init__	68	73	1	6	▲ 6
render	75	79	1	5	4
__init__	84	86	1	3	4
render	88	93	1	6	4
_wrap_text	95	108	5	14	2
render	113	118	1	6	4
_wrap_text	120	133	5	14	2

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
test_font_factory	3	7	1	5	0
test_font_rendering	9	14	1	6	0
test_text_renderer_with_wrap	16	19	1	4	0
test_text_renderer_with_rich_text	21	24	1	4	0

AI: Gemini 1.5 Flash

Design pattern: Interpreter

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
interpret	6	7	1	2	3
__init__	10	11	1	2	2
interpret	13	14	1	2	3
__init__	17	18	1	2	2
interpret	20	24	2	5	3
__init__	27	30	1	4	4
interpret	32	47	6	15	3
__init__	50	52	1	3	2
parse	54	57	1	4	1
_parse_expression	59	67	3	8	1
_parse_term	69	77	3	8	1
_parse_factor	79	99	▲ 11	21	1
check_remaining	101	104	2	3	1
main	106	126	3	19	0

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
test_number_expression	11	14	1	4	1
test_variable_expression	16	19	1	4	1
test_variable_expression_not_found	21	25	1	5	1
test_binary_expression_addition	27	32	1	6	1
test_binary_expression_subtraction	34	39	1	6	1
test_binary_expression_multiplication	41	46	1	6	1
test_binary_expression_division	48	53	1	6	1
test_binary_expression_division_by_zero	55	61	1	7	1
test_parse_simple_expression	63	67	1	5	1
test_parse_variable_expression	69	73	1	5	1
test_parse_addition_expression	75	83	1	9	1
test_parse_complex_expression	85	101	1	17	1
test_interpret_complex_expression	103	108	1	6	1
test_parse_invalid_expression	110	113	1	4	1
test_parse_invalid_operator	115	118	1	4	1

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
interpret	6	7	1	2	3
__init__	10	11	1	2	2
interpret	13	14	1	2	3
__init__	17	18	1	2	2
interpret	20	24	2	5	3
__init__	27	30	1	4	4
interpret	32	47	6	15	3
__init__	50	52	1	3	2
parse	54	57	1	4	1
_parse_expression	59	67	3	8	1
_parse_term	69	77	3	8	1
_parse_factor	79	99	▲ 11	21	1

check_remaining	101	104	2	3	1
main	106	126	3	19	0

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold 50)	Parameter Count (Threshold: 4)
test_number_expression	11	14	1	4	1
test_variable_expression	16	19	1	4	1
test_variable_expression_not_found	21	25	1	5	1
test_binary_expression_addition	27	32	1	6	1
test_binary_expression_subtraction	34	39	1	6	1
test_binary_expression_multiplication	41	46	1	6	1
test_binary_expression_division	48	53	1	6	1
test_binary_expression_division_by_zero	55	61	1	7	1
test_parse_simple_expression	63	67	1	5	1
test_parse_variable_expression	69	73	1	5	1
test_parse_addition_expression	75	83	1	9	1
test_parse_complex_expression	85	101	1	17	1
test_interpret_complex_expression	103	108	1	6	1
test_parse_invalid_expression	110	113	1	4	1
test_parse_invalid_operator	115	118	1	4	1

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
interpret	6	7	1	2	3
__init__	10	11	1	2	2
interpret	13	14	1	2	3
__init__	17	18	1	2	2
interpret	20	23	2	4	3
__init__	26	29	1	4	4
interpret	31	35	1	4	3
_evaluate	37	48	6	12	3
__init__	51	53	1	3	2
parse	55	58	1	4	1
_parse_expression	60	67	3	8	1
_parse_term	69	76	3	8	1
_parse_factor	78	87	5	10	1
_parse_number	89	94	3	6	1
_parse_variable	96	99	1	4	1
_parse_parentheses	101	107	3	7	1
_has_more_tokens	109	110	1	2	1
_check_remaining	112	114	2	3	1
main	116	136	3	19	0

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
test_number_expression	11	14	1	4	1
test_variable_expression	16	19	1	4	1
test_variable_expression_not_found	21	25	1	5	1
test_binary_expression_addition	27	32	1	6	1
test_binary_expression_subtraction	34	39	1	6	1
test_binary_expression_multiplication	41	46	1	6	1
test_binary_expression_division	48	53	1	6	1
test_binary_expression_division_by_zero	55	61	1	7	1
test_parse_simple_expression	63	67	1	5	1
test_parse_variable_expression	69	73	1	5	1
test_parse_addition_expression	75	83	1	9	1
test_parse_complex_expression	85	101	1	17	1
test_interpret_complex_expression	103	108	1	6	1
test_parse_invalid_expression	110	113	1	4	1
test_parse_invalid_operator	115	118	1	4	1

AI: Gemini 1.5 Pro

Design pattern: Flyweight

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
__init__	5	8	1	4	1
render	11	12	1	2	4
set_font	14	15	1	2	3
set_color	17	18	1	2	2
__init__	22	24	1	3	3
__init__	28	30	1	3	2
render	32	35	1	4	3
__init__	39	40	1	2	1
get_flyweight	42	45	2	4	2
__init__	49	50	1	2	2
render_text	52	58	2	7	⚠ 6

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
test_character_flyweight_rendering	10	15	1	5	0
test_flyweight_factory_creation_and_reuse	18	22	1	5	0
test_text_renderer_rendering	25	30	1	5	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
__init__	5	8	1	4	4
__eq__	10	15	3	6	2
__init__	22	23	1	2	1
get_font	25	29	2	5	4
__init__	33	39	1	7	▲ 7
render	44	45	1	2	2
__init__	49	50	1	2	1
render	52	61	1	8	2

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
text_renderer	12	13	1	2	0
test_flyweight_text_renderer	46	56	2	9	▲ 5

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
__init__	4	9	1	6	▲ 6
__str__	11	19	4	9	1
get_font	26	30	2	5	▲ 6
__init__	34	40	1	7	▲ 7
__str__	42	43	1	2	1
render	48	49	1	2	2
render	53	54	1	2	2

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
font_factory	7	8	1	2	0
default_renderer	12	13	1	2	0
test_font_creation	16	22	1	7	1
test_font_styles	25	29	1	5	1
test_font_reuse	32	35	1	4	1
test_text_creation	38	46	1	9	1
test_text_rendering	49	53	1	5	2

AI: Gemini 1.5 Pro

Design pattern: Interpreter

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
interpret	5	6	1	2	2
__init__	10	11	1	2	2
interpret	13	14	1	2	2
__init__	18	20	1	3	3
interpret	22	23	1	2	2
__init__	27	29	1	3	3
interpret	31	32	1	2	2
__init__	36	38	1	3	3
interpret	40	41	1	2	2
__init__	45	47	1	3	3
interpret	49	52	2	4	2
__init__	56	57	1	2	2
interpret	59	63	2	5	2
__init__	67	68	1	2	1
set_variable	70	71	1	2	3
get_variable	73	74	1	2	2
main	77	88	1	9	0

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
test_number	12	15	1	4	0
test_add	18	21	1	4	0
test_subtract	24	27	1	4	0
test_multiply	30	33	1	4	0
test_divide	36	39	1	4	0
test_divide_by_zero	42	46	1	5	0
test_variable	49	53	1	5	0
test_undefined_variable	56	60	1	5	0
test_complex_expression	63	71	1	8	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
interpret	4	5	1	2	2
__init__	9	10	1	2	2
interpret	12	13	1	2	2
__init__	17	19	1	3	3
interpret	21	22	1	2	2
__init__	26	28	1	3	3
interpret	30	31	1	2	2
__init__	35	37	1	3	3
interpret	39	40	1	2	2
__init__	44	46	1	3	3
interpret	48	51	2	4	2
__init__	55	56	1	2	2
interpret	58	62	2	5	2
__init__	66	67	1	2	1
set_variable	69	70	1	2	3
get_variable	72	73	1	2	2
main	76	87	1	9	0

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
test_number	13	16	1	4	0
test_add	19	22	1	4	0
test_subtract	25	28	1	4	0
test_multiply	31	34	1	4	0
test_divide	37	40	1	4	0
test_divide_by_zero	43	47	1	5	0
test_variable	50	54	1	5	0
test_undefined_variable	57	61	1	5	0
test_complex_expression	64	72	1	8	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
interpret	5	6	1	2	2
__init__	10	11	1	2	2
interpret	13	14	1	2	2
__init__	18	20	1	3	3
interpret	22	23	1	2	2
__init__	27	29	1	3	3
interpret	31	32	1	2	2
__init__	36	38	1	3	3
interpret	40	41	1	2	2
__init__	45	47	1	3	3
interpret	49	52	2	4	2
__init__	56	57	1	2	2
interpret	59	63	2	5	2
__init__	67	68	1	2	1
set_variable	70	71	1	2	3
get_variable	73	74	1	2	2
main	77	89	1	10	0

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
test_number	13	16	1	4	0
test_add	19	22	1	4	0
test_subtract	25	28	1	4	0
test_multiply	31	34	1	4	0
test_divide	37	40	1	4	0
test_divide_by_zero	43	47	1	5	0
test_variable	50	54	1	5	0
test_undefined_variable	57	61	1	5	0
test_complex_expression	64	72	1	8	0

AI: Github Copilot

Design pattern: Flyweight

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
__init__	3	6	1	4	4
get_font	12	16	2	5	4
__init__	19	25	1	7	▲ 7
render	27	29	1	2	1
__init__	33	34	1	2	1
add_text	36	39	1	4	▲ 9
render_all	41	42	2	2	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
test_font_factory	7	10	1	4	0
test_text_rendering	12	15	1	4	0
test_text_rendering_system	17	24	1	8	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
__init__	2	5	1	4	4
display	7	9	1	2	▲ 6
get_font	15	19	2	5	4
__init__	22	28	1	7	▲ 9
render	30	31	1	2	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
test_text_rendering	4	6	1	3	0
test_multilingual_support	8	10	1	3	0
test_rich_text_support	12	14	1	3	0
test_text_wrapping	16	18	1	3	0
test_special_characters	20	22	1	3	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
__init__	2	5	1	4	4
__init__	8	12	1	5	▲ 5
get_font	18	22	2	5	4
get_style	28	32	2	5	▲ 5
__init__	35	38	1	4	4
render	41	43	1	2	2

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
test_text_rendering	4	10	1	7	0
test_font_factory	12	15	1	4	0
test_text_style_factory	17	20	1	4	0
test_multilingual_support	22	28	1	7	0
test_special_characters	30	36	1	7	0

AI: Github Copilot

Design pattern: Interpreter

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
interpret	7	8	1	2	2
__init__	12	13	1	2	2
interpret	15	16	1	2	2
__init__	20	21	1	2	2
interpret	23	26	2	4	2
__init__	30	33	1	4	4
interpret	35	38	1	4	2
__init__	46	47	1	2	1
interpret	49	53	2	5	2

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
test_number	5	6	1	2	0
test_variable	8	10	1	3	0
test_variable_not_found	12	16	2	5	0
test_addition	18	20	1	3	0
test_subtraction	22	24	1	3	0
test_multiplication	26	28	1	3	0
test_division	30	32	1	3	0
test_complex_expression	34	41	1	8	0
test_division_by_zero	43	48	2	6	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
interpret	7	8	1	2	2
__init__	12	13	1	2	2
interpret	15	16	1	2	2
__init__	20	21	1	2	2
interpret	23	26	2	4	2
__init__	30	33	1	4	4
interpret	35	38	1	4	2
__init__	46	47	1	2	1
interpret	49	53	2	5	2

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
test_number	5	6	1	2	0
test_variable	8	10	1	3	0
test_variable_not_found	12	16	2	5	0
test_addition	18	20	1	3	0
test_subtraction	22	24	1	3	0
test_multiplication	26	28	1	3	0
test_division	30	32	1	3	0
test_complex_expression	34	41	1	8	0
test_division_by_zero	43	48	2	6	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
interpret	7	8	1	2	2
__init__	12	13	1	2	2
interpret	15	16	1	2	2
__init__	20	21	1	2	2
interpret	23	26	2	4	2
__init__	30	33	1	4	4
interpret	35	38	1	4	2
__init__	46	47	1	2	1
interpret	49	53	2	5	2

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
test_number	5	6	1	2	0
test_variable	8	10	1	3	0
test_variable_not_found	12	16	2	5	0
test_addition	18	20	1	3	0
test_subtraction	22	24	1	3	0
test_multiplication	26	28	1	3	0
test_division	30	32	1	3	0
test_complex_expression	34	41	1	8	0
test_division_by_zero	43	48	2	6	0

ภาษาโปรแกรมที่ใช้ : Java

AI: Chatgpt

Design pattern: Flyweight

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
FontFlyweight::FontFlyweight	19	25	1	7	▲ 5
FontFlyweight::render	28	38	1	10	2
FontFlyweightFactory::getFontFlyweight	45	51	2	7	▲ 5
TextProperties::TextProperties	63	70	1	8	▲ 6
TextRenderingSystem::renderText	77	80	1	4	▲ 7
TextRenderingSystem::main	82	92	1	7	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
TextRenderingSystemTest::setUp	12	14	1	3	0
TextRenderingSystemTest::testRenderTextWithRegularFont	17	20	1	4	0
TextRenderingSystemTest::testRenderTextWithBoldFont	23	26	1	4	0
TextRenderingSystemTest::testFlyweightFactoryReuse	29	35	1	6	0
TextRenderingSystemTest::testFlyweightFactoryCreateNew	38	44	1	6	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ConcreteTextFormat::ConcreteTextFormat	21	29	1	9	▲ 7
ConcreteTextFormat::renderText	32	38	1	6	2
TextFormatFactory::getTextFormat	45	51	2	7	▲ 7
TextContext::TextContext	60	64	1	5	3
TextContext::getColor	66	68	1	3	0
TextContext::getLineSpacing	70	72	1	3	0
TextContext::getKerning	74	76	1	3	0
TextRenderingSystem::main	81	91	1	9	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
FlyweightPatternTest::setUp	12	14	1	3	0
FlyweightPatternTest::testFlyweightCreationAndReusability	17	22	1	5	0
FlyweightPatternTest::testFlyweightDifferentFormats	25	30	1	5	0
FlyweightPatternTest::testRenderTextWithContext	33	40	1	6	0
FlyweightPatternTest::testContextAttributes	43	49	1	6	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ConcreteFont::ConcreteFont	18	23	1	6	4
ConcreteFont::renderText	26	31	1	6	▲ 6
ConcreteFont::equals	34	39	7	6	1
ConcreteFont::hashCode	42	44	1	3	0
FontFactory::getFont	50	56	2	7	4
TextRenderingSystem::renderText	60	64	1	5	▲ 10
TextRenderingSystem::main	66	70	1	5	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
TextRenderingSystemTest::setUp	13	15	1	3	0
TextRenderingSystemTest::testRenderTextWithDifferentFonts	18	22	1	5	0
TextRenderingSystemTest::testRenderTextWithSameFontInstance	25	30	1	5	0
TextRenderingSystemTest::testRenderTextWithDifferentFontAttributes	33	38	1	5	0
TextRenderingSystemTest::testRenderTextAlignment	41	46	1	6	0
TextRenderingSystemTest::testRenderWithSpecialCharactersAndUnicode	49	52	1	4	0

AI: Chatgpt

Design pattern: Interpreter

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
VariableExpression::VariableExpression	28	30	1	3	1
VariableExpression::interpret	33	39	2	7	1
BinaryExpression::BinaryExpression	47	51	1	5	3
BinaryExpression::interpret	54	69	6	15	1
ExpressionParser::parse	73	125	▲ 20	46	1
ExpressionParser::createBinaryExpression	127	131	1	5	2
ExpressionParser::precedence	133	141	5	9	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
ExpressionTest::testNumberExpression	12	15	1	4	0
ExpressionTest::testVariableExpression	18	26	1	7	0
ExpressionTest::testBinaryExpressionAddition	29	32	1	4	0
ExpressionTest::testBinaryExpressionSubtraction	35	38	1	4	0
ExpressionTest::testBinaryExpressionMultiplication	41	44	1	4	0
ExpressionTest::testBinaryExpressionDivision	47	50	1	4	0
ExpressionTest::testDivisionByZero	53	56	1	4	0
ExpressionTest::testComplexExpression	59	66	1	8	0
ExpressionTest::testUnknownOperator	69	71	1	3	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
NumberExpression::NumberExpression	15	17	1	3	1
NumberExpression::interpret	20	22	1	3	1
VariableExpression::VariableExpression	29	31	1	3	1
VariableExpression::interpret	34	39	2	6	1
AddExpression::AddExpression	47	50	1	4	2
AddExpression::interpret	53	55	1	3	1
SubtractExpression::SubtractExpression	63	66	1	4	2
SubtractExpression::interpret	69	71	1	3	1
MultiplyExpression::MultiplyExpression	79	82	1	4	2
MultiplyExpression::interpret	85	87	1	3	1
DivideExpression::DivideExpression	95	98	1	4	2
DivideExpression::interpret	101	107	2	7	1
ExpressionParser::parse	112	149	7	37	2
InterpreterDemo::main	154	164	1	8	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
InterpreterTest::testAddition	10	17	1	7	0
InterpreterTest::testSubtraction	20	27	1	7	0
InterpreterTest::testMultiplication	30	37	1	7	0
InterpreterTest::testDivision	40	47	1	7	0
InterpreterTest::testDivisionByZero	50	57	1	7	0
InterpreterTest::testInvalidVariable	60	66	1	6	0
InterpreterTest::testComplexExpression	69	76	1	7	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
NumberExpression::NumberExpression	14	16	1	3	1
NumberExpression::interpret	19	21	1	3	1
VariableExpression::VariableExpression	28	30	1	3	1
VariableExpression::interpret	33	39	2	7	1
AddExpression::AddExpression	47	50	1	4	2
AddExpression::interpret	53	55	1	3	1
SubtractExpression::SubtractExpression	63	66	1	4	2
SubtractExpression::interpret	69	71	1	3	1
MultiplyExpression::MultiplyExpression	79	82	1	4	2
MultiplyExpression::interpret	85	87	1	3	1
DivideExpression::DivideExpression	95	98	1	4	2
DivideExpression::interpret	101	107	2	7	1
Interpreter::parse	112	140	8	29	1
Interpreter::isOperator	142	144	4	3	1
Interpreter::isVariable	146	148	1	3	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
InterpreterTest::testBasicMathOperations	10	24	1	11	0
InterpreterTest::testDivisionByZero	27	31	1	5	0
InterpreterTest::testVariableHandling	34	41	1	7	0
InterpreterTest::testUndefinedVariable	44	48	1	5	0
InterpreterTest::testComplexExpression	51	58	1	7	0

AI: Gemini 1.5 Flash

Design pattern: Flyweight

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
TextRenderingSystem::TextRenderingSystem	24	32	1	9	1
TextRenderingSystem::setFont	34	40	2	7	1
TextRenderingSystem::setFontSize	42	45	1	4	1
TextRenderingSystem::setColor	47	49	1	3	1
TextRenderingSystem::setAlignment	51	53	1	3	1
TextRenderingSystem::setLineSpacing	55	57	1	3	1
TextRenderingSystem::setLetterSpacing	59	61	1	3	1
TextRenderingSystem::render	63	90	5	21	3

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
TextRenderingSystemTest::testSetFont	10	17	1	7	0
TextRenderingSystemTest::testSetFontSize	20	25	1	6	0
TextRenderingSystemTest::testSetColor	28	32	1	5	0
TextRenderingSystemTest::testSetAlignment	35	39	1	5	0
TextRenderingSystemTest::testSetLineSpacing	42	46	1	5	0
TextRenderingSystemTest::testSetLetterSpacing	49	53	1	5	0
TextRenderingSystemTest::testRender	56	63	1	5	0
TextRenderingSystemTest::MockGraphics2D::drawString	69	72	1	2	3

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
TextRenderer::renderText	10	72	7	37	▲ 7
TextRenderer::getFont	74	80	2	7	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
TextRendererTest::renderText_FontManagement	11	16	1	5	0
TextRendererTest::renderText_TextColor	19	24	1	5	0
TextRendererTest::renderText_TextAlignment	27	36	1	5	0
TextRendererTest::renderText_LineSpacing	39	44	1	4	0
TextRendererTest::renderText_Kerning	47	52	1	4	0
TextRendererTest::renderText_WrapWidth	55	61	1	4	0
TextRendererTest::getFont_FontCache	64	71	1	7	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
TextRenderer::renderText	11	52	▲ 11	23	▲ 8

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
TextRendererTest::testRenderText_ValidInput	11	22	1	10	0
TextRendererTest::testRenderText_InvalidAlignment	25	36	1	10	0
TextRendererTest::testRenderText_InvalidLineSpacing	39	50	1	10	0
TextRendererTest::testRenderText_InvalidLetterSpacing	53	64	1	10	0
TextRendererTest::testRenderText_InvalidLanguage	67	78	1	10	0

AI: Gemini 1.5 Flash

Design pattern: Interpreter

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
NumberExpression::NumberExpression	12	14	1	3	1
NumberExpression::getValue	16	18	1	3	0
NumberExpression::interpret	21	23	1	3	1
VariableExpression::VariableExpression	29	31	1	3	1
VariableExpression::interpret	34	40	2	7	1
BinaryExpression::BinaryExpression	48	52	1	5	3
BinaryExpression::interpret	55	74	6	19	1
Parser::Parser	81	84	1	4	1
Parser::parse	86	88	1	3	0
Parser::_parseExpression	90	105	4	16	0
Parser::_parseTerm	107	121	4	15	0
Parser::_parseFactor	123	147	▲ 11	25	0
Parser::validateCharacter	149	156	9	8	1
Expression_Interpreter_R1_code_GeminiFlash::main	160	184	3	23	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Expression_Interpreter_R1_test_GeminiFlash::testNumberExpression	9	13	1	5	0
Expression_Interpreter_R1_test_GeminiFlash::testVariableExpression	16	21	1	6	0
Expression_Interpreter_R1_test_GeminiFlash::testVariableExpressionNotFound	24	28	1	5	0
Expression_Interpreter_R1_test_GeminiFlash::testBinaryExpressionAddition	31	37	1	7	0
Expression_Interpreter_R1_test_GeminiFlash::testBinaryExpressionSubtraction	40	46	1	7	0
Expression_Interpreter_R1_test_GeminiFlash::testBinaryExpressionMultiplication	49	55	1	7	0
Expression_Interpreter_R1_test_GeminiFlash::testBinaryExpressionDivision	58	64	1	7	0
Expression_Interpreter_R1_test_GeminiFlash::testBinaryExpressionDivisionByZero	67	73	1	7	0
Expression_Interpreter_R1_test_GeminiFlash::testParseSimpleExpression	76	81	1	6	0
Expression_Interpreter_R1_test_GeminiFlash::testParseVariableExpression	84	89	1	6	0
Expression_Interpreter_R1_test_GeminiFlash::testParseAdditionExpression	92	101	1	10	0
Expression_Interpreter_R1_test_GeminiFlash::testParseComplexExpression	104	137	1	21	0
Expression_Interpreter_R1_test_GeminiFlash::testInterpretComplexExpression	140	148	1	9	0
Expression_Interpreter_R1_test_GeminiFlash::testParseInvalidExpression	151	154	1	4	0
Expression_Interpreter_R1_test_GeminiFlash::testParseInvalidOperator	157	160	1	4	0
Expression_Interpreter_R1_test_GeminiFlash::testParseMissingClosingParenthesis	163	166	1	4	0
Expression_Interpreter_R1_test_GeminiFlash::testParseInvalidCharacter	169	172	1	4	0
Expression_Interpreter_R1_test_GeminiFlash::testParseEmptyExpression	175	178	1	4	0
Expression_Interpreter_R1_test_GeminiFlash::testParseNestedParentheses	181	196	1	16	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
NumberExpression::NumberExpression	11	13	1	3	1
NumberExpression::getValue	15	17	1	3	0
NumberExpression::interpret	20	22	1	3	1
VariableExpression::VariableExpression	28	30	1	3	1
VariableExpression::interpret	33	39	2	7	1
BinaryExpression::BinaryExpression	47	51	1	5	3
BinaryExpression::interpret	54	73	6	19	1
Parser::Parser	80	83	1	4	1
Parser::parse	85	87	1	3	0
Parser::_parseExpression	89	104	4	16	0
Parser::_parseTerm	106	120	4	15	0
Parser::_parseFactor	122	146	▲ 11	25	0
Parser::validateCharacter	148	155	9	8	1
Expression_Interpreter_R2_code_GeminiFlash::main	159	183	3	23	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Expression_Interpreter_R2_test_GeminiFlash::testNumberExpression	9	13	1	5	0
Expression_Interpreter_R2_test_GeminiFlash::testVariableExpression	16	21	1	6	0
Expression_Interpreter_R2_test_GeminiFlash::testVariableExpressionNotFound	24	28	1	5	0
Expression_Interpreter_R2_test_GeminiFlash::testBinaryExpressionAddition	31	37	1	7	0
Expression_Interpreter_R2_test_GeminiFlash::testBinaryExpressionSubtraction	40	46	1	7	0
Expression_Interpreter_R2_test_GeminiFlash::testBinaryExpressionMultiplication	49	55	1	7	0
Expression_Interpreter_R2_test_GeminiFlash::testBinaryExpressionDivision	58	64	1	7	0
Expression_Interpreter_R2_test_GeminiFlash::testBinaryExpressionDivisionByZero	67	73	1	7	0
Expression_Interpreter_R2_test_GeminiFlash::testParseSimpleExpression	76	81	1	6	0
Expression_Interpreter_R2_test_GeminiFlash::testParseVariableExpression	84	89	1	6	0
Expression_Interpreter_R2_test_GeminiFlash::testParseAdditionExpression	92	101	1	10	0
Expression_Interpreter_R2_test_GeminiFlash::testParseComplexExpression	104	137	1	21	0
Expression_Interpreter_R2_test_GeminiFlash::testInterpretComplexExpression	140	148	1	9	0
Expression_Interpreter_R2_test_GeminiFlash::testParseInvalidExpression	151	154	1	4	0
Expression_Interpreter_R2_test_GeminiFlash::testParseInvalidOperator	157	160	1	4	0
Expression_Interpreter_R2_test_GeminiFlash::testParseMissingClosingParenthesis	163	166	1	4	0
Expression_Interpreter_R2_test_GeminiFlash::testParseInvalidCharacter	169	172	1	4	0
Expression_Interpreter_R2_test_GeminiFlash::testParseEmptyExpression	175	178	1	4	0
Expression_Interpreter_R2_test_GeminiFlash::testParseNestedParentheses	181	196	1	16	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
NumberExpression::NumberExpression	11	13	1	3	1
NumberExpression::getValue	15	17	1	3	0
NumberExpression::interpret	20	22	1	3	1
VariableExpression::VariableExpression	28	30	1	3	1
VariableExpression::interpret	33	39	2	7	1
BinaryExpression::BinaryExpression	47	51	1	5	3
BinaryExpression::interpret	54	73	6	19	1
Parser::Parser	80	83	1	4	1
Parser::parse	85	87	1	3	0
Parser::_parseExpression	89	104	4	16	0
Parser::_parseTerm	106	120	4	15	0
Parser::_parseFactor	122	146	▲ 11	25	0
Parser::validateCharacter	148	155	9	8	1
Expression_Interpreter_R3_code_GeminiFlash::main	159	183	3	23	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Expression_Interpreter_R3_test_GeminiFlash::testNumberExpression	9	13	1	5	0
Expression_Interpreter_R3_test_GeminiFlash::testVariableExpression	16	21	1	6	0
Expression_Interpreter_R3_test_GeminiFlash::testVariableExpressionNotFound	24	28	1	5	0
Expression_Interpreter_R3_test_GeminiFlash::testBinaryExpressionAddition	31	37	1	7	0
Expression_Interpreter_R3_test_GeminiFlash::testBinaryExpressionSubtraction	40	46	1	7	0
Expression_Interpreter_R3_test_GeminiFlash::testBinaryExpressionMultiplication	49	55	1	7	0
Expression_Interpreter_R3_test_GeminiFlash::testBinaryExpressionDivision	58	64	1	7	0
Expression_Interpreter_R3_test_GeminiFlash::testBinaryExpressionDivisionByZero	67	73	1	7	0
Expression_Interpreter_R3_test_GeminiFlash::testParseSimpleExpression	76	81	1	6	0
Expression_Interpreter_R3_test_GeminiFlash::testParseVariableExpression	84	89	1	6	0
Expression_Interpreter_R3_test_GeminiFlash::testParseAdditionExpression	92	101	1	10	0
Expression_Interpreter_R3_test_GeminiFlash::testParseComplexExpression	104	137	1	21	0
Expression_Interpreter_R3_test_GeminiFlash::testInterpretComplexExpression	140	148	1	9	0
Expression_Interpreter_R3_test_GeminiFlash::testParseInvalidExpression	151	154	1	4	0
Expression_Interpreter_R3_test_GeminiFlash::testParseInvalidOperator	157	160	1	4	0
Expression_Interpreter_R3_test_GeminiFlash::testParseMissingClosingParenthesis	163	166	1	4	0
Expression_Interpreter_R3_test_GeminiFlash::testParseInvalidCharacter	169	172	1	4	0
Expression_Interpreter_R3_test_GeminiFlash::testParseEmptyExpression	175	178	1	4	0
Expression_Interpreter_R3_test_GeminiFlash::testParseNestedParentheses	181	196	1	16	0

AI: Gemini 1.5 Pro

Design pattern: Flyweight

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
ConcreteTextFormat::ConcreteTextFormat	20	27	1	8	▲ 6
ConcreteTextFormat::display	30	34	1	5	4
ConcreteTextFormat::getFontStyle	36	45	3	10	0
TextFormatFactory::getTextFormat	52	58	2	7	▲ 6
TextRenderer::setTextFormat	65	67	1	3	▲ 6
TextRenderer::renderText	69	71	1	3	4

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
FlyweightPatternTest::testGetTextFormat	13	17	1	5	0
FlyweightPatternTest::testRenderText	20	209	1	▲ 130	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
FontFactory::getFont	12	25	4	14	4
TextRendererImpl::render	36	41	1	5	▲ 5
TextRendererFactory::getRenderer	48	50	1	3	0
TextFormat::TextFormat	62	68	1	7	▲ 5
TextFormat::getFont	70	72	1	3	0
TextFormat::toString	77	79	1	3	0

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
FlyweightTextRenderingTest::testFontFactory	11	18	1	7	0
FlyweightTextRenderingTest::testTextRenderer	21	25	1	5	0
FlyweightTextRenderingTest::testTextRendering	28	33	1	5	0
FlyweightTextRenderingTest::testTextFormatToString	38	41	1	4	0
FlyweightTextRenderingTest::testFontFactoryWithDifferentStyles	44	51	1	8	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ConcreteTextFormat::ConcreteTextFormat	19	26	1	8	▲ 6
ConcreteTextFormat::drawText	29	33	1	5	4
ConcreteTextFormat::getFontStyle	35	44	3	10	0
TextFormatFactory::getTextFormat	51	57	2	7	▲ 6

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
FlyweightTextRenderingTest::testTextRendering	11	33	1	13	0

AI: Gemini 1.5 Pro

Design pattern: Interpreter

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Number::Number	11	13	1	3	1
Number::interpret	16	18	1	3	1
Variable::Variable	24	26	1	3	1
Variable::interpret	29	35	2	7	1
Add::Add	42	45	1	4	2
Add::interpret	48	50	1	3	1
Subtract::Subtract	57	60	1	4	2
Subtract::interpret	63	65	1	3	1
Multiply::Multiply	72	75	1	4	2
Multiply::interpret	78	80	1	3	1
Divide::Divide	87	90	1	4	2
Divide::interpret	93	99	2	7	1
Context::setVariable	105	107	1	3	2
Context::getVariable	109	111	1	3	1
Expression_Interpreter_R1_code_GeminiPro::main	114	128	1	11	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Expression_Interpreter_R1_test_GeminiPro::testNumber	10	14	1	5	0
Expression_Interpreter_R1_test_GeminiPro::testAdd	17	21	1	5	0
Expression_Interpreter_R1_test_GeminiPro::testSubtract	24	28	1	5	0
Expression_Interpreter_R1_test_GeminiPro::testMultiply	31	35	1	5	0
Expression_Interpreter_R1_test_GeminiPro::testDivide	38	42	1	5	0
Expression_Interpreter_R1_test_GeminiPro::testDivideByZero	45	49	1	5	0
Expression_Interpreter_R1_test_GeminiPro::testVariable	52	57	1	6	0
Expression_Interpreter_R1_test_GeminiPro::testUndefinedVariable	60	64	1	5	0
Expression_Interpreter_R1_test_GeminiPro::testComplexExpression	67	77	1	10	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Number::Number	12	14	1	3	1
Number::interpret	17	19	1	3	1
Variable::Variable	25	27	1	3	1
Variable::interpret	30	36	2	7	1
Add::Add	43	46	1	4	2
Add::interpret	49	51	1	3	1
Subtract::Subtract	58	61	1	4	2
Subtract::interpret	64	66	1	3	1
Multiply::Multiply	73	76	1	4	2
Multiply::interpret	79	81	1	3	1
Divide::Divide	88	91	1	4	2
Divide::interpret	94	100	2	7	1
Context::setVariable	106	108	1	3	2
Context::getVariable	110	112	1	3	1
Expression_Interpreter_R2_code_GeminiPro::main	115	129	1	11	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Expression_Interpreter_R2_test_GeminiPro::testNumber	11	15	1	5	0
Expression_Interpreter_R2_test_GeminiPro::testAdd	18	22	1	5	0
Expression_Interpreter_R2_test_GeminiPro::testSubtract	25	29	1	5	0
Expression_Interpreter_R2_test_GeminiPro::testMultiply	32	36	1	5	0
Expression_Interpreter_R2_test_GeminiPro::testDivide	39	43	1	5	0
Expression_Interpreter_R2_test_GeminiPro::testDivideByZero	46	50	1	5	0
Expression_Interpreter_R2_test_GeminiPro::testVariable	53	58	1	6	0
Expression_Interpreter_R2_test_GeminiPro::testUndefinedVariable	61	65	1	5	0
Expression_Interpreter_R2_test_GeminiPro::testComplexExpression	68	78	1	10	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Number::Number	12	14	1	3	1
Number::interpret	17	19	1	3	1
Variable::Variable	25	27	1	3	1
Variable::interpret	30	36	2	7	1
Add::Add	43	46	1	4	2
Add::interpret	49	51	1	3	1
Subtract::Subtract	58	61	1	4	2
Subtract::interpret	64	66	1	3	1
Multiply::Multiply	73	76	1	4	2
Multiply::interpret	79	81	1	3	1
Divide::Divide	88	91	1	4	2
Divide::interpret	94	100	2	7	1
Context::setVariable	106	108	1	3	2
Context::getVariable	110	112	1	3	1
Expression_Interpreter_R3_code_GeminiPro::main	115	129	1	11	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Expression_Interpreter_R3_test_GeminiPro::testNumber	10	14	1	5	0
Expression_Interpreter_R3_test_GeminiPro::testAdd	17	21	1	5	0
Expression_Interpreter_R3_test_GeminiPro::testSubtract	24	28	1	5	0
Expression_Interpreter_R3_test_GeminiPro::testMultiply	31	35	1	5	0
Expression_Interpreter_R3_test_GeminiPro::testDivide	38	42	1	5	0
Expression_Interpreter_R3_test_GeminiPro::testDivideByZero	45	49	1	5	0
Expression_Interpreter_R3_test_GeminiPro::testVariable	52	57	1	6	0
Expression_Interpreter_R3_test_GeminiPro::testUndefinedVariable	60	64	1	5	0
Expression_Interpreter_R3_test_GeminiPro::testComplexExpression	67	77	1	10	0

AI: Github Copilot

Design pattern: Flyweight

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ConcreteTextStyle::setFont	35	41	1	7	▲ 5
ConcreteTextStyle::setColor	44	46	1	3	1
ConcreteTextStyle::setAlignment	49	51	1	3	1
ConcreteTextStyle::setLineSpacing	54	56	1	3	1
ConcreteTextStyle::setKerning	59	61	1	3	1
ConcreteTextStyle::setMultilingualSupport	64	66	1	3	1
ConcreteTextStyle::setRichTextSupport	69	71	1	3	1
ConcreteTextStyle::setTextWrapping	74	76	1	3	1
ConcreteTextStyle::setSpecialCharactersSupport	79	81	1	3	1
ConcreteTextStyle::getFontType	84	86	1	3	0
ConcreteTextStyle::getFontSize	88	90	1	3	0
ConcreteTextStyle::isBold	92	94	1	3	0
ConcreteTextStyle::isItalic	96	98	1	3	0
ConcreteTextStyle::isUnderline	100	102	1	3	0
ConcreteTextStyle::getColor	104	106	1	3	0
ConcreteTextStyle::getAlignment	108	110	1	3	0
ConcreteTextStyle::getLineSpacing	112	114	1	3	0
ConcreteTextStyle::getKerning	116	118	1	3	0
ConcreteTextStyle::isMultilingualSupport	120	122	1	3	0
ConcreteTextStyle::isRichTextSupport	124	126	1	3	0
ConcreteTextStyle::isTextWrapping	128	130	1	3	0
ConcreteTextStyle::isSpecialCharactersSupport	132	134	1	3	0
TextStyleFactory::getTextStyle	141	148	2	8	1
Text::Text	156	159	1	4	2
Text::render	161	164	1	3	0
TextRenderingSystem::main	169	185	1	14	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
TextStyleFactoryTest::testGetTextStyle	8	18	1	9	0
ConcreteTextStyleTest::testSetFont	23	32	1	9	0
ConcreteTextStyleTest::testSetColor	35	40	1	5	0
ConcreteTextStyleTest::testSetAlignment	43	48	1	5	0
ConcreteTextStyleTest::testSetLineSpacing	51	56	1	5	0
ConcreteTextStyleTest::testSetKerning	59	64	1	5	0
ConcreteTextStyleTest::testSetMultilingualSupport	67	72	1	5	0
ConcreteTextStyleTest::testSetRichTextSupport	75	80	1	5	0
ConcreteTextStyleTest::testSetTextWrapping	83	88	1	5	0
ConcreteTextStyleTest::testSetSpecialCharactersSupport	91	96	1	5	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Font::Font	19	25	1	7	▲ 5
Font::applyStyle	28	31	1	3	1
Color::Color	37	39	1	3	1
Color::applyStyle	42	45	1	3	1
Alignment::Alignment	51	53	1	3	1
Alignment::applyStyle	56	59	1	3	1
TextStyleFactory::getFont	66	72	2	7	▲ 5
TextStyleFactory::getColor	74	79	2	6	1
TextStyleFactory::getAlignment	81	86	2	6	1
TextRenderer::renderText	91	96	2	6	2
TextRenderer::main	98	105	1	7	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
TextRendererTest::testRenderText	10	17	1	7	0
TextRendererTest::testFontReuse	20	25	1	5	0
TextRendererTest::testColorReuse	28	33	1	5	0
TextRendererTest::testAlignmentReuse	36	41	1	5	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
Font::Font	18	24	1	7	▲ 5
Font::applyStyle	27	30	1	3	1
Color::Color	36	38	1	3	1
Color::applyStyle	41	44	1	3	1
Alignment::Alignment	50	52	1	3	1
Alignment::applyStyle	55	58	1	3	1
LineSpacing::LineSpacing	64	66	1	3	1
LineSpacing::applyStyle	69	72	1	3	1
Kerning::Kerning	78	80	1	3	1
Kerning::applyStyle	83	86	1	3	1
TextStyleFactory::getFont	93	99	2	7	▲ 5
TextStyleFactory::getColor	101	106	2	6	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity <small>(Threshold: 10)</small>	Lines of Code <small>(Threshold: 50)</small>	Parameter Count <small>(Threshold: 4)</small>
TextRendererTest::setUpStreams	16	18	1	3	0
TextRendererTest::restoreStreams	21	23	1	3	0
TextRendererTest::testTextRendering	26	44	1	16	0
TextRendererTest::testTextStyleFactory	47	67	1	17	0

AI: Github Copilot

Design pattern: Interpreter

รอบที่ 1

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
NumberExpression::NumberExpression	12	14	1	3	1
NumberExpression::interpret	17	19	1	3	1
VariableExpression::VariableExpression	25	27	1	3	1
VariableExpression::interpret	30	36	2	7	1
BinaryExpression::BinaryExpression	44	48	1	5	3
BinaryExpression::interpret	51	70	6	19	1
Parser::Parser	77	80	1	4	1
Parser::parse	82	84	1	3	0
Parser::_parseExpression	86	101	4	16	0
Parser::_parseTerm	103	117	4	15	0
Parser::_parseFactor	119	143	▲ 11	25	0
Parser::validateCharacter	146	153	9	8	1
Expression_Interpreter_R1_code_Copilot::main	157	181	3	23	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Expression_Interpreter_R1_test_Copilot::testNumberExpression	10	14	1	5	0
Expression_Interpreter_R1_test_Copilot::testVariableExpression	17	22	1	6	0
Expression_Interpreter_R1_test_Copilot::testAdditionExpression	25	29	1	5	0
Expression_Interpreter_R1_test_Copilot::testSubtractionExpression	32	36	1	5	0
Expression_Interpreter_R1_test_Copilot::testMultiplicationExpression	39	43	1	5	0
Expression_Interpreter_R1_test_Copilot::testDivisionExpression	46	50	1	5	0
Expression_Interpreter_R1_test_Copilot::testComplexExpression	53	65	1	11	0
Expression_Interpreter_R1_test_Copilot::testDivisionByZero	68	72	1	5	0
Expression_Interpreter_R1_test_Copilot::testInvalidOperator	75	79	1	5	0
Expression_Interpreter_R1_test_Copilot::testInvalidCharacter	82	85	1	4	0
Expression_Interpreter_R1_test_Copilot::testMissingClosingParenthesis	88	91	1	4	0
Expression_Interpreter_R1_test_Copilot::testNestedParentheses	94	99	1	6	0
Expression_Interpreter_R1_test_Copilot::testEmptyExpression	102	105	1	4	0
Expression_Interpreter_R1_test_Copilot::testVariableNotFound	108	112	1	5	0

รอบที่ 2

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
NumberExpression::NumberExpression	12	14	1	3	1
NumberExpression::interpret	17	19	1	3	1
VariableExpression::VariableExpression	25	27	1	3	1
VariableExpression::interpret	30	36	2	7	1
BinaryExpression::BinaryExpression	44	48	1	5	3
BinaryExpression::interpret	51	70	6	19	1
Parser::Parser	77	80	1	4	1
Parser::parse	82	84	1	3	0
Parser::_parseExpression	86	101	4	16	0
Parser::_parseTerm	103	117	4	15	0
Parser::_parseFactor	119	143	▲ 11	25	0
Parser::validateCharacter	146	153	9	8	1
Expression_Interpreter_R2_code_Copilot::main	157	181	3	23	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Expression_Interpreter_R2_test_Copilot::testNumberExpression	10	14	1	5	0
Expression_Interpreter_R2_test_Copilot::testVariableExpression	17	22	1	6	0
Expression_Interpreter_R2_test_Copilot::testAdditionExpression	25	29	1	5	0
Expression_Interpreter_R2_test_Copilot::testSubtractionExpression	32	36	1	5	0
Expression_Interpreter_R2_test_Copilot::testMultiplicationExpression	39	43	1	5	0
Expression_Interpreter_R2_test_Copilot::testDivisionExpression	46	50	1	5	0
Expression_Interpreter_R2_test_Copilot::testComplexExpression	53	65	1	11	0
Expression_Interpreter_R2_test_Copilot::testDivisionByZero	68	72	1	5	0
Expression_Interpreter_R2_test_Copilot::testInvalidOperator	75	79	1	5	0
Expression_Interpreter_R2_test_Copilot::testInvalidCharacter	82	85	1	4	0
Expression_Interpreter_R2_test_Copilot::testMissingClosingParenthesis	88	91	1	4	0
Expression_Interpreter_R2_test_Copilot::testNestedParentheses	94	99	1	6	0
Expression_Interpreter_R2_test_Copilot::testEmptyExpression	102	105	1	4	0
Expression_Interpreter_R2_test_Copilot::testVariableNotFound	108	112	1	5	0

รอบที่ 3

Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
NumberExpression::NumberExpression	12	14	1	3	1
NumberExpression::interpret	17	19	1	3	1
VariableExpression::VariableExpression	25	27	1	3	1
VariableExpression::interpret	30	36	2	7	1
BinaryExpression::BinaryExpression	44	48	1	5	3
BinaryExpression::interpret	51	70	6	19	1
Parser::Parser	77	80	1	4	1
Parser::parse	82	84	1	3	0
Parser::parseExpression	86	101	4	16	0
Parser::parseTerm	103	117	4	15	0
Parser::parseFactor	119	143	▲ 11	25	0
Parser::validateCharacter	146	153	9	8	1
Expression_Interpreter_R3_code_Copilot::main	157	181	3	23	1

Test Code

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Expression_Interpreter_R3_test_Copilot::testNumberExpression	10	14	1	5	0
Expression_Interpreter_R3_test_Copilot::testVariableExpression	17	22	1	6	0
Expression_Interpreter_R3_test_Copilot::testAdditionExpression	25	29	1	5	0
Expression_Interpreter_R3_test_Copilot::testSubtractionExpression	32	36	1	5	0
Expression_Interpreter_R3_test_Copilot::testMultiplicationExpression	39	43	1	5	0
Expression_Interpreter_R3_test_Copilot::testDivisionExpression	46	50	1	5	0
Expression_Interpreter_R3_test_Copilot::testComplexExpression	53	65	1	11	0
Expression_Interpreter_R3_test_Copilot::testDivisionByZero	68	72	1	5	0
Expression_Interpreter_R3_test_Copilot::testInvalidOperator	75	79	1	5	0
Expression_Interpreter_R3_test_Copilot::testInvalidCharacter	82	85	1	4	0
Expression_Interpreter_R3_test_Copilot::testMissingClosingParenthesis	88	91	1	4	0
Expression_Interpreter_R3_test_Copilot::testNestedParentheses	94	99	1	6	0
Expression_Interpreter_R3_test_Copilot::testEmptyExpression	102	105	1	4	0
Expression_Interpreter_R3_test_Copilot::testVariableNotFound	108	112	1	5	0

สรุปผลการทดลอง

ในการทดลองครั้งนี้ พบว่า Generative AI ทุกตัวที่ศึกษา (ChatGPT-4, Gemini 1.5 Pro, Gemini 1.5 Flash, และ GitHub Copilot) สามารถสร้างโค้ดได้อย่างน่าพอใจ โดยเฉพาะในเรื่องของ Cyclomatic Complexity ที่ถือว่าต่ำ ซึ่งหมายความว่าโค้ดที่สร้างขึ้นมีความซับซ้อนน้อย และง่ายต่อการเข้าใจและบำรุงรักษา อย่างไรก็ตาม พบว่าการสร้างโค้ดโดย ChatGPT มีมาตรฐานที่ไม่สม่ำเสมอในบางครั้ง แม้จะใช้ Prompt ที่คล้ายกัน ส่งผลให้โค้ดบางส่วนที่สร้างขึ้นไม่ได้มาตรฐานที่คาดหวัง

นอกจากนี้ ยังพบว่า Generative AI มีปัญหาในด้านการมีค่า Parameter Count ที่สูง โดยเฉพาะในกรณีของการสร้าง Interpreter Pattern ซึ่งอาจส่งผลให้เกิดความซับซ้อนในโค้ดที่ถูกสร้างขึ้นได้ ในด้านของการสร้าง Test Code พบว่า GitHub Copilot มีประสิทธิภาพที่ดีที่สุด โดยสามารถสร้างทั้งโค้ดและテストโค้ดได้อย่างมีคุณภาพ ซึ่งส่งผลให้เปอร์เซ็นต์ Statement Coverage และ Branch Coverage สูงที่สุดในการทดสอบโค้ดที่สร้างขึ้น นอกจากนี้ ยังมีความสามารถในการทำงานตาม Requirements ที่กำหนดไว้ได้ในบางครั้ง

การทดลองนี้แสดงให้เห็นว่าแต่ละ Generative AI มีความสามารถในการสร้างโค้ดที่มีคุณภาพ แต่มีความแตกต่างกันในเรื่องของความสม่ำเสมอและประสิทธิภาพในการสร้างโค้ดและการทดสอบ โดยสรุปได้ดังนี้:

1. **ChatGPT-4o:** แม้จะสร้างโค้ดที่มี Cyclomatic Complexity ต่ำ แต่บางครั้งก็แสดงให้เห็นถึงความไม่สม่ำเสมอในการตอบสนองต่อ Prompt ที่คล้ายกัน ส่งผลให้โค้ดบางส่วนอาจไม่ได้มาตรฐานที่ต้องการ
2. **GitHub Copilot:** มีความโดยเด่นในการสร้างทั้งโค้ดและ test code ที่มีคุณภาพสูง ซึ่งสามารถทำให้ได้ทั้ง Statement Coverage และ Branch Coverage ที่สูงที่สุด และยังทำได้ตาม Requirements ที่กำหนดไว้ในบางครั้ง
3. **Gemini 1.5 Pro และ Gemini 1.5 Flash:** แม้จะสร้างโค้ดที่มีคุณภาพ แต่ไม่สามารถทำได้ในระดับสูงเหมือน GitHub Copilot และมีความซับซ้อนในการสร้างโค้ดที่สูงขึ้นในบางกรณี

การทดลองนี้ชี้ให้เห็นถึงข้อดีและข้อจำกัดของการใช้ Generative AI ในการพัฒนาและทดสอบซอฟต์แวร์ ทำให้ผู้พัฒนาและผู้จัดการซอฟต์แวร์สามารถเลือกใช้ Generative AI ได้อย่างเหมาะสมและมีประสิทธิภาพมากขึ้นในกระบวนการพัฒนาซอฟต์แวร์ ในอนาคต