A logo with a design on it

Description automatically generated

**โครงงาน**

Numerical Method

Calculator Website

**จัดทำโดย**

6504062620175 นายอธิชา เล็กสรรเสริญ

**เสนอ**

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

**วิชา 040613204 Object-Oriented Programming**

ภาคเรียนที่ 1 ปีการศึกษา 2566

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

บทที่ 1 **ที่มาและความสำคัญของโครงการ**

โครงงานนี้จัดขึ้นเพื่อวัดผลความสามารถในการเรียนวิชา Object Oriented Programming โดยการนําเรื่องที่เรียนมาสร้างเป็นชิ้นงานในรูปแบบเว็บ โดยใช้แนวคิดการเขียนโปรแกรมแบบเชิงวัดถุ และยังช่วยให้ผู้จัดทำเรียนรู้อุปกรณ์และเครื่องมือ ผู้จัดทําได้สร้างเว็บนี้ขึ้นมา

**ประเภทของโครงการ**

โปรแกรมเว็บแอปพลิเคชัน Full-Stack

**ประโยชน์**

1. เพื่อให้สามารถคำนวนปัญหาทาง Numerical ได้อย่างสะดวกสบาย
2. เพื่อนำความรู้จากวิชา Database, Numerical Methods, Object-Oriented Programming มาประยุกต์ใช้
3. เพื่อนำแนวคิดการเขียนโปรแกรมแบบเชิงวัดถุมาประยุกต์ใช้

**ขอบเขตของโครงการ**

1. ความต้องการของระบบ (Functional Requirements)

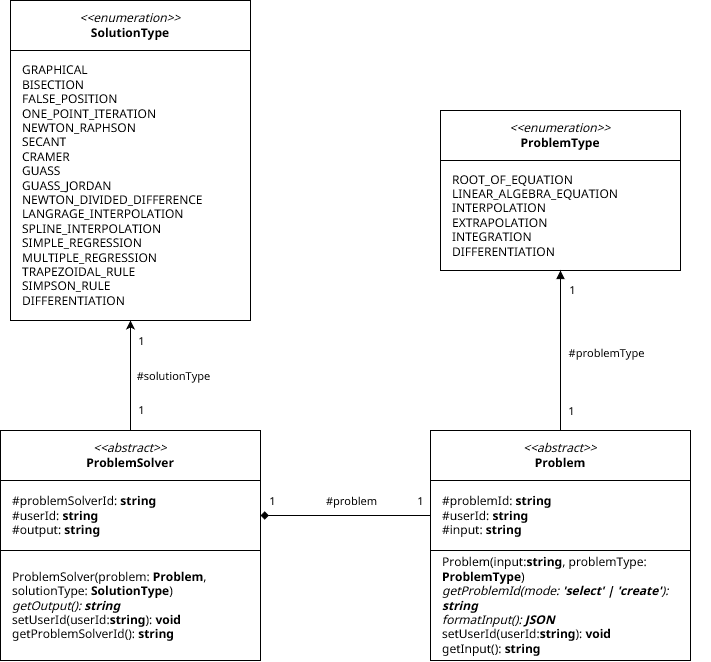
* สามารถคำนวนปัญหาทาง Numerical ดังนี้ได้
* รากของสมการ (Root of Equation)
* สมการพีชคณิตเชิงเส้น (Linear Algebra Equation)
* การประมาณค่าในช่วง (Interpolation)
* การประมาณค่านอกช่วง (Extrapolation)
* ปริพันธ์ (Integration)
* อนุพันธ์ (Differentiation)
  + สามารถแสดงผลลัพธ์ออกเป็นกราฟแบบโต้ตอบ (Interactive graph) โดยใช้ ployly.js
  + สามารถเก็บข้อมูลของปัญหาที่เคยคำนวณ โดยใช้ PlanetScale Database
  + สามารถเข้าสู่ระบบด้วย Google Account
  + สามารถแสดงปัญหาที่เคยคำนวนแล้ว และแสดงจำนวนครั้งที่ถูกคำนวณได้

1. ตารางเวลาการดำเนินโครงการ (Project Schedule)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **งาน** | **สัปดาห์ 1**  **(ก.ย.)** | | | **สัปดาห์ 2**  **(ต.ค.)** | | | **สัปดาห์ 3**  **(ต.ค.)** | | | **สัปดาห์ 4**  **(ต.ค.)** | | | **สัปดาห์ 5**  **(ต.ค.)** | | |
| **25** | **27** | **29** | **2** | **4** | **6** | **9** | **11** | **13** | **16** | **18** | **20** | **23** | **25** | **27** |
| 1. ติดตั้งและออกแบบฐานข้อมูล | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1. ระบบเข้าสู่ระบบด้วย Google |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1. ปัญหา Root of Equation |  |  | x | x | x | x |  |  |  |  |  |  |  |  |  |
| 1. กราฟแบบโต้ตอบ (plotly.js) |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |
| 1. ปัญหา Linear Algebra Equation |  |  |  |  |  |  |  | x | x | x |  |  |  |  |  |
| 1. ปัญหา Interpolation |  |  |  |  |  |  |  |  |  |  | x | x |  |  |  |
| 1. ปัญหา Extrapolation |  |  |  |  |  |  |  |  |  |  |  |  | x |  |  |
| 1. ปัญหา Integration |  |  |  |  |  |  |  |  |  |  |  |  |  | x |  |
| 1. ปัญหา Differentiation |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x |
|  | 0% | 6% | 13% | 22% | 28% | 34% | 44% | 50% | 56% | 66% | 72% | 78% | 88% | 94% | 100% |

บทที่ 2 **การพัฒนา**

**แผนภาพ Class Diagram**



โครงการนี้จะมีคราสหลักอยู่สองคราสคือ

1. คราส Problem จะเก็บข้อมูลของปัญหาต่างๆ เช่น
   * Id ของปัญหา
   * Id ของผู้ใช้งาน (ที่ถามปัญหา)
   * รายละเอียดของปัญหาซึ่งจะเก็บในรูปของ **JSON** ที่เป็น **string**
   * ประเภทของปัญหา เช่น Root of equation, …

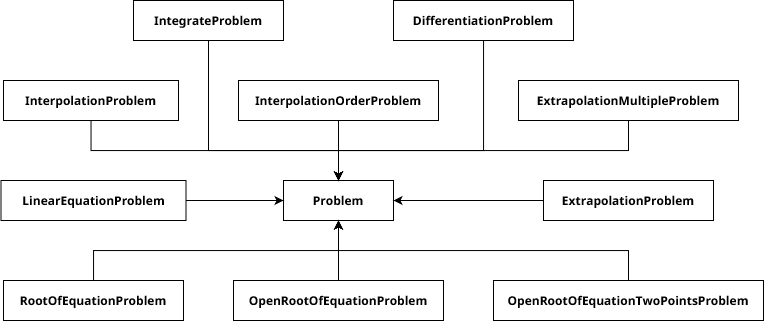
และจะมี Methods ต่างๆ ดังนี้

* getProblemId(mode: ‘select’ | ‘create’) ซึ่งเป็น Abstract method โดย methods นี้จะสามารถเลือกได้ว่าจะสร้าง Problem ใหม่หรือดึงจากที่มีอยู่บนฐานข้อมูล ซึ่งแต่ละปัญหาจะมีการดึงข้อมูลจากฐานข้อมูลที่แตกต่างกัน จึงสามารถ Inherit คราสนี้เพื่อไปสร้างเป็นปัญหาต่างๆ ได้
* setUserId(userId: string) ตั้ง Id ของผู้ใช้งาน (ที่ถามปัญหา)
* getInput() ส่งรายละเอียดของปัญหา
* formatInput() เป็น Abstract class ที่จะเช็คว่าข้อมูลที่ส่งมาจากผู้ใช้งานให้ตรงกับปัญหาที่ถาม หากไม่ตรงจะไม่สามารถแก้ปัญหานั้นได้ หรือข้อมูลเยอะเกินที่ระบบรองรับ

1. คราส ProblemSolver จะเก็บวิธีการแก้ปัญหารูปแบบต่างๆ ของปัญหานั้นๆ เช่น
   * Id ของ Solution
   * Id ของผู้ใช้งาน (ที่ถามปัญหาด้วย Solution นี้)
   * รายละเอียดของวิธีการแก้ปัญหาซึ่งจะเก็บอยู่ในรูปของ **JSON** ที่เป็น **string**
   * ประเภทของวิธีแก้ปัญหา เช่น Graphical Method, …

และจะมี Methods ต่างๆ ดังนี้

* getOutput() เป็น Abstract Method ที่จะดึง input ของคราส Problem มาแก้ปัญหาต่างๆ ตามวิธีการแก้ปัญหาของ Object นั้นๆ โดย Methods จะดึงวิธีแก้ปัญหามาจากฐานข้อมูลและเพิ่มจำนวนการแก้ ถ้าหากเคยมีคนถามคำถามนี้แล้ว หากไม่มีจะทำการเพิ่มเข้าไปในฐานข้อมูล
* setUserId(userId: string) ตั้ง Id ของผู้ใช้งาน (ที่ถามปัญหาด้วย Solution นี้)
* getProblemSolverId() โดย Method นี้จะดึง Id ของ Solution จากฐานข้อมูล จาก Id ของปัญหาและวิธีการแก้ปัญหา หากมีจะตั้งให้ output เป็นคำตอบจากฐานข้อมูลเลย



แผนภาพการสืบทอดคราส ของคราส Problem

(แบบง่าย)

A diagram of a diagram

Description automatically generated A screen shot of a computer code

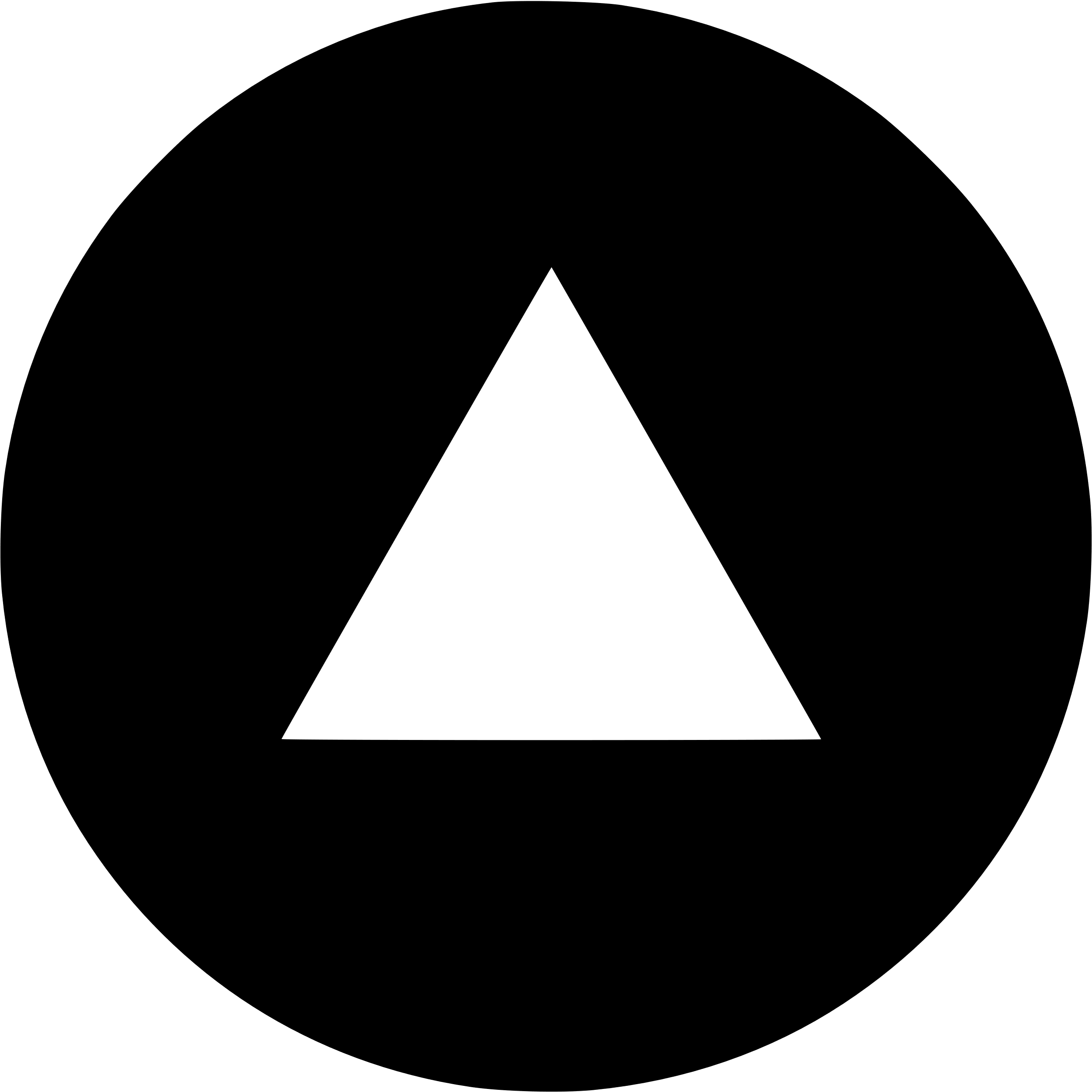
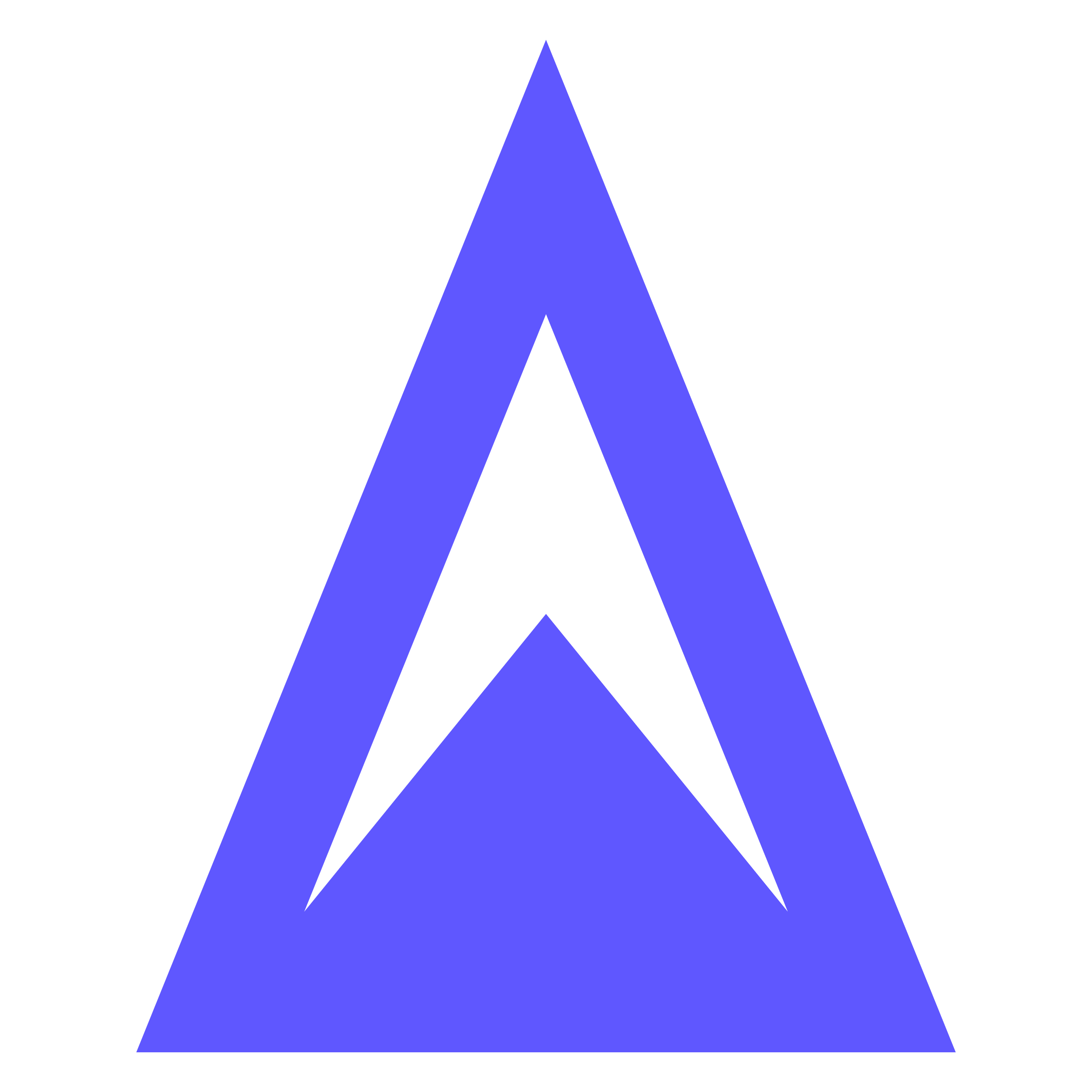
Description automatically generated

ตัวอย่างของการ Implement คราส Graphical Method และ

ปัญหาของ Root Of Equation

**รูปแบบการพัฒนาโครงการ**

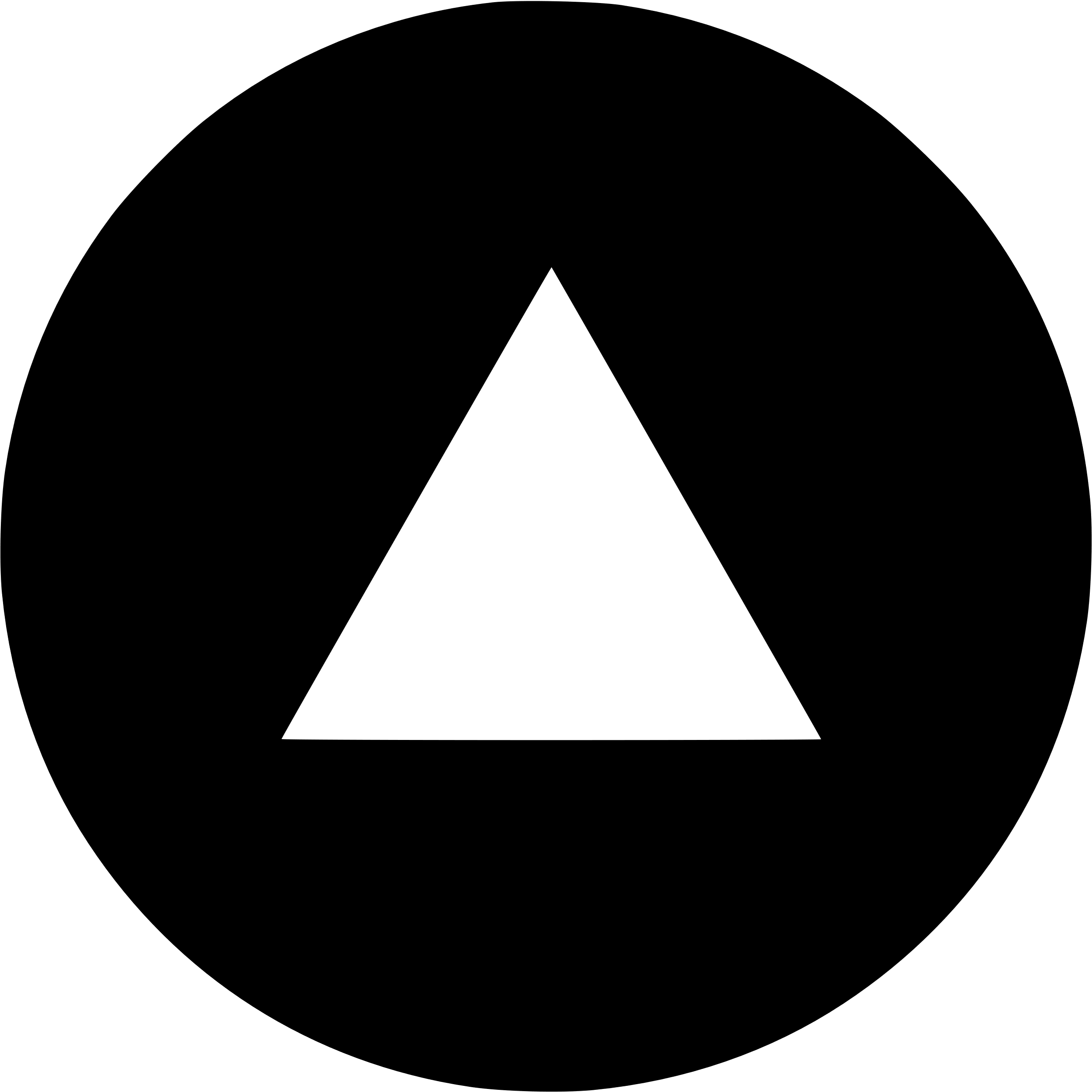
A white circle with black lines in it

Description automatically generated **** A white letter on a red background

Description automatically generatedA black background with a black square

Description automatically generated with medium confidenceA red lines on a black background

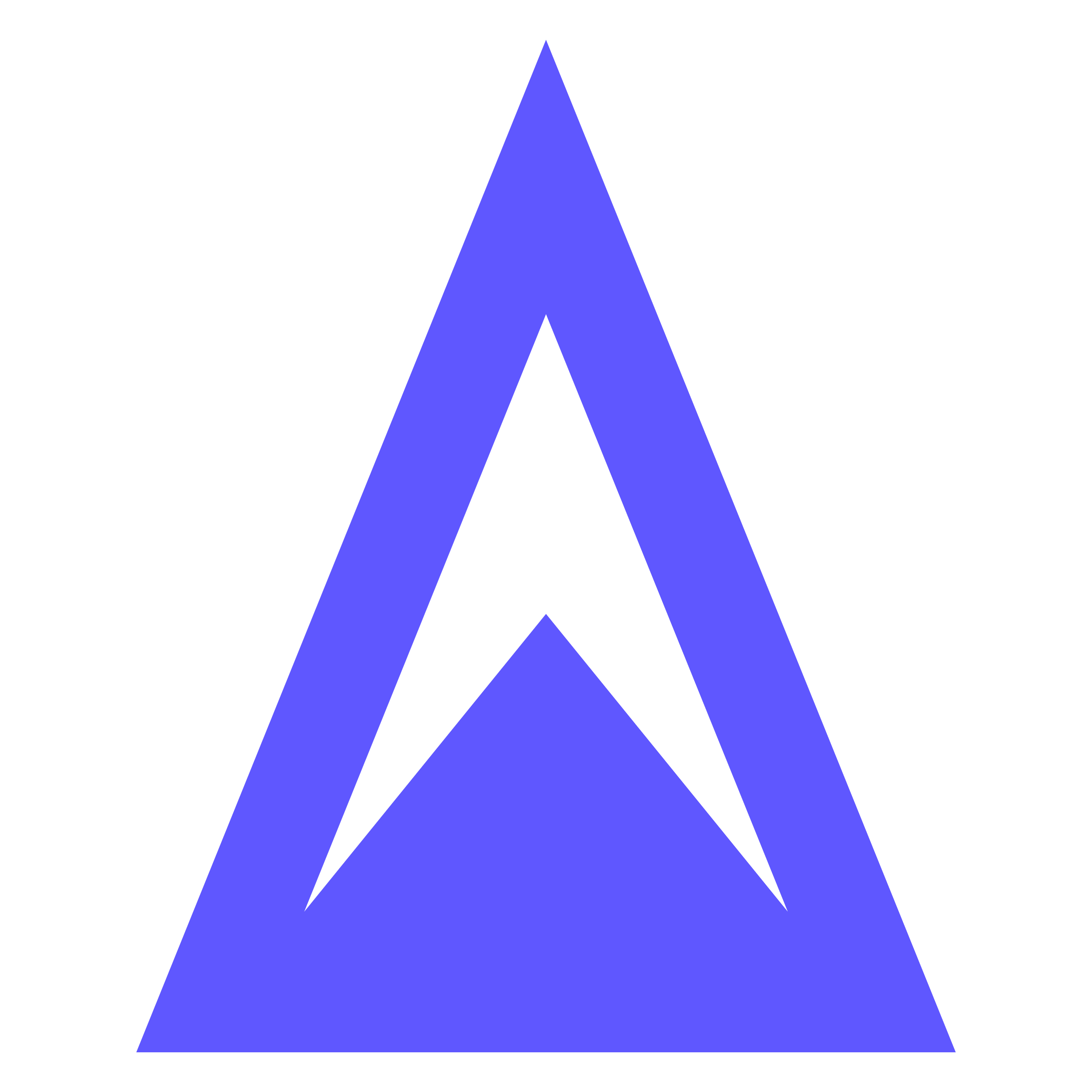
Description automatically generated

* ภาษา: undefined Svelte, undefined TypeScript
* Framework: SvelteKit (SvelteKit Framework Svelte + Vite Framework Vite)
* ผู้ให้บริหารโฮสติ้ง: **** Vercel
* ฐานข้อมูล
  + Prisma ORM Prisma เป็นไลบรารี่จับคู่เชื่อมโยงระหว่างโมเดลเชิงวัตถุและเชิงสัมพันธ์ หรือ ORM
  + A white circle with black lines in it

    Description automatically generated PlanetScale ผู้ให้บริหารฐานข้อมูลแบบ MySQL
* CSS และ Components:
  + TailwindCSS TailwindCSS เป็น CSS Utility Framework
  + ![A red lines on a black background

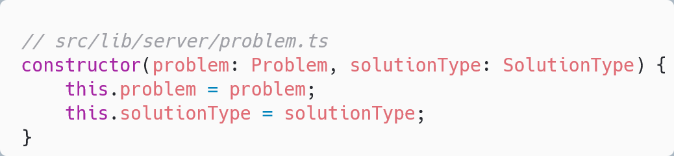
    Description automatically generated](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABoAAAAaCAMAAACelLz8AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAnUExURQAAAOtPJyYMBkoYDAEAAFEbDTwUCUQWC1EbDEQWCiUMBkkYDAAAANeSuk4AAAANdFJOU////////////////wA96CKGAAAACXBIWXMAACHVAAAh1QEEnLSdAAAAQElEQVQoU2MYgoCRC8rAAIyMUAYGoLcMJ4RCAjAZRg4IjQBwGTYIjQBwGQzjiDAN0wnQcMO0BwY4MUwbzICBAQCPGwB62If+8gAAAABJRU5ErkJggg==) shadcn-svelte เป็นไลบรารี่ที่รวม Components ต่างๆ
  + A black background with a black square

    Description automatically generated with medium confidence KaTeX เป็นไลบรารี่แสดงสมการทางคณิตศาสตร์
* กราฟแบบโต้ตอบ: Plotly.js Plotly.js
* ไลบรารี่คำนวนทางคณิตศาสตร์: ![A white letter on a red background

  Description automatically generated](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABoAAAAaCAMAAACelLz8AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAqUExURdw5Euh7Yf///+uPeeFZOPXLwf76+frn4+BWNd9OK/nk39w6FOZ2WwAAAMW5auwAAAAOdFJOU/////////////////8ARcDcyAAAAAlwSFlzAAAh1QAAIdUBBJy0nQAAAEhJREFUKFPdjNsNACAIA7HGV+L+84pYjW6g3geluYD8gYPvkRGsDqDYwNiRKFgPKKgYWYrOdRbN8PEKY+4adaud7Ypxm3oUkQb9bAFptddeZgAAAABJRU5ErkJggg==) Math.js
* ไลบรารี่ระบบล็อคอิน:  Lucia

**แนวคิดการเขียนโปรแกรมเชิงวัตถุ**

* **Constructor**

Constructor ของคราส Problem และคราส ProblemSolver เพื่อรับ Parameters ต่างๆ มาตั้ง Attribute



เมื่อคราส RootOfEquationProblem inherit คราส Problem มาจึงไม่จำเป็นต้องใส่ solutionType แต่สามารถเรียก super class Constructor ได้เลย

* **Encapsulation**

A screen shot of a computer code

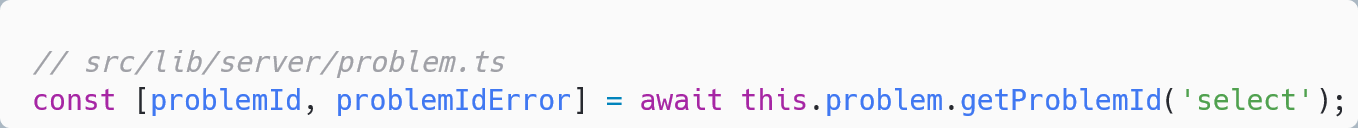
Description automatically generated

Attribute ของคราสหลักๆ เช่นคราส Problem และคราส ProblemSolver จะเป็นแบบ Protected เพื่อให้สามารถมองเห็นได้เฉพาะคราสที่ Inherit ไปเช่น คราส CramerSolver, คราส GuassEliminationSolver, … โดยหากต้องการค่าของ Attribute สามารถใช้ได้จาก getter methods เช่น getInput(), getOutput()

* **Composition**

****

ในคราสของ ProblemSolver จะมี Attribute เป็นวัตถุ (Object) ของคราส Problem ซึ่งไว้เก็บปัญหาที่คราส ProblemSolver นั้นๆ พยายามแก้ ตัวอย่างเช่นคราส TrapezoidalRuleSolver จะมี Class ปัญหาคือคราส IntegrateProblem

****

การเรียกใช้งานคราส Problem

ในคราส ProblemSolver

* **Polymorphism**
* **Abstract**
* **Inheritance**