



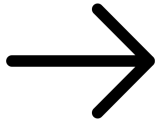
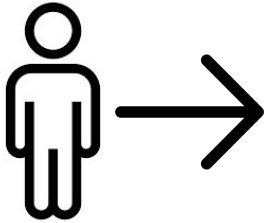
Development of a Hint Recommendation Module for an Automated Programming Assignment Assessment System

Author: Krivonosov Egor Dmitrievich

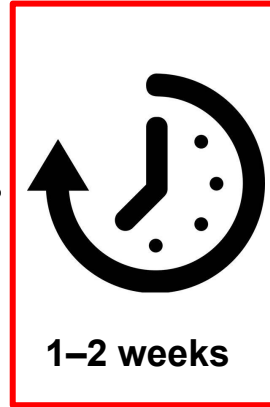
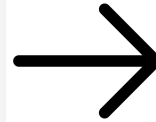
Saint-Petersburg, 2024

Problem

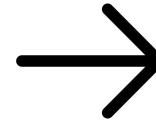
Student



```
code-n-test Tech Support TestCodePage Teacher Logout EN
Java
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         double a = scanner.nextDouble();
7         double b = scanner.nextDouble();
8         double c = scanner.nextDouble();
9         double discriminant = b * b - 4 * a * c;
10
11         if (discriminant > 0) {
12             double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
13             double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
14             System.out.println(root1 + " " + root2);
15         } else if (discriminant == 0) {
16             double root = -b / (2 * a);
17             System.out.println(root);
18         }
19     }
20 }
```



1-2 weeks



Teacher



Purpose and the Solution

- **Purpose:**

- Improve the quality of code written by students through real-time advice and feedback.

- **Solution:**

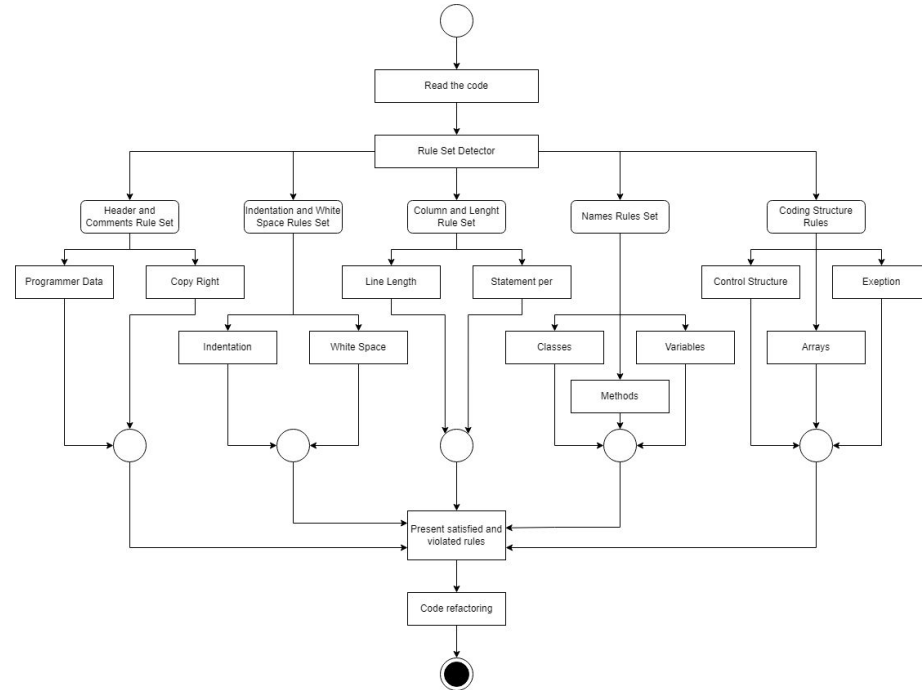
- My project aims to develop a **hints module** to provide immediate, **automated feedback** on code submissions within the Code&Test platform. This tool will facilitate a real-time learning environment, allowing students to quickly identify and correct coding errors, thus accelerating their learning curve and enhancing code quality.



- Using static code analysers and dynamic ones together allows you to get the best result.
- Many solutions support only a certain programming language, without the possibility of extending it.
- There are very few solutions that give feedback on code in an understandable form for any student with any level of programming knowledge.
- Mostly all the solutions are paid or do not have open access.



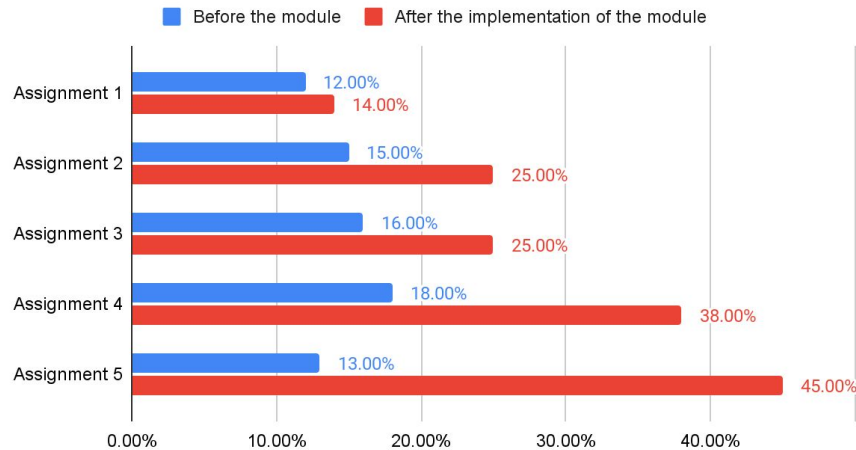
Architecture of the Solution



Picture 1 - The part responsible for Static code analysis

Results to Date

Code quality % (Java)



On average, student
code quality
improved by 25%
over the semester.

Conclusions and Future

- **The next phase:**
 - a. Adding support for other programming languages like Python, JavaScript, C, C++ and other...
 - b. Integrating additional levels of educational feedback.
 - c. Adding dynamic analysis based on test case generation for the programme.
 - d. Improving the UI.



Thanks for your attention!

it's **MO** *re than a*
UNIVERSITY