Як свій фінальний проект з курсу «Основи програмування» я вирішила займатись покращенням середовища програмування Pharo. Зокрема, переважна частина моєї роботи ґрунтувалась на перетворенні правил. Правила у Pharo відповідають за виведення помилок на екран, якщо в методі чи класі є щось, що порушує ці правила. Вони були написані використовуючи застарілий SmallLint формат, а також за рахунок часто громіздкої імплементації було складно зрозуміти, для чого ці правила, що вони роблять, і як саме вони це виконують. Моєю роботою було створити нову, більш зрозумілу і компактнішу імплементацію правил, а також зробити краще пояснення у виді невеликих підказок, які вибиваються на екран поряд з помилкою, а також у вигляді підсвічення місця коду, яке збуджує цю помилку. Крім цього всього, я також працювала над створенням авто фіксів, зокрема, реалізовуючи авто фікс, який створює методи, які ще не існують.

Крім переписування правил в інший формат, мені також часто доводилось повністю перероблювати спосіб їх імплементації. Наприклад, для декількох я робила їхні node-версії, що часто не тільки швидше, а й правильніше знаходило методи, що порушували правила. Зокрема, детальніше про процес створення node-версії правила є описано на моєму блозі на прикладі RBBadMessageRule. В [цій частині](https://medium.com/@myroslavarm/node-version-of-the-rbbadmessagerule-60e27c38a4bd) я з’ясувала, що node-версія мого правила працює швидше, а в [цій](https://medium.com/@myroslavarm/node-version-of-the-rbbadmessagerule-testing-accuracy-810a1e8b3d0d), що вона й з більшою точністю знаходить методи з цими так званими «поганими» повідомленням. Щоправда, такий підхід працює не для всіх правил, адже для деяких обхід дерева займає більшість часу, і така версія може працювати повільніше оригінальної.

Також в останній період роботи над курсовою окрім перетворення правил, я ще додатково екпериментувала з налаштуваннями середовища і створенням власних налаштувань, які могли б вмикати/вимикати окреме правило. Невдовзі планую зробити також налаштування, яке уможливлює банення правила для всього імеджу в загальному, адже іноді в процесі розробки девелоперам необхідно ігнорувати якесь правило, а наразі є тільки спосіб відключити його на стороні певного класу, методу чи пекеджу, а не усього імеджу.