Разработать плагин для Eclipse, который позволит открывать/сохранять PVS-Studio отчет , осуществлять навигацию по предупреждениям и запускать анализ текущего проекта.

## Вводная часть к плагинам

Разработать статический анализатор кода — это ещё не все. Анализатор, как правило, представляет собой инструмент командной строки, которым в таком виде не все захотят пользоваться. Этот факт наталкивает на мысль, что для более удобной и приятной работы с анализатором нужны различные плагины, которые будут позволять запускать анализ, просматривать отчеты анализатора, сохранять отчеты и так далее, не вдаваясь в подробности консольного запуска анализатора.

На сегодняшний день у нас реализованы плагины:

* для C++: Visual Studio
* для C#: Visual Studio, JetBrains Rider
* для Java: IntelliJ IDEA, gradle, maven.

Но много плагинов не бывает! Самую предпочитаемую java разработчикам IDE (IntelliJ IDEA) мы уже поддержали. Но что на счет остальных? Несмотря на то, что Eclipse IDE в последнее время теряет свою популярность, она по-прежнему остается в топе используемых IDE. Eclipse представляет собой основанную на Java расширяемую платформу разработки с открытым исходным кодом. По сути - это просто среда разработки и набор сервисов для построения приложений на основе встраиваемых компонентов (плагинов). В составе Eclipse имеется стандартный набор плагинов, в том числе хорошо известный инструментарий - Java Development Tools (JDT). Поэтому многие ассоциируют Eclipse как Java IDE, но многие могут не знать, что Eclipse так же C/C++ IDE, JavaScript/TypeScript IDE, PHP IDE и так далее, благодаря множеству разнообразным плагинам. Поскольку Eclipse полностью состоит из плагинов, пора взяться за дело и написать свой плагины.

Официальная [страничка](https://www.eclipse.org/articles/Article-Your%20First%20Plug-in/YourFirstPlugin.html) Eclipse по разработке плагина поможет вам сформировать почву для дальнейших действий.

Полезные ссылки для дальнейшего изучения:

* [Eclipse Plug-in Development FAQ](https://wiki.eclipse.org/Eclipse_Plug-in_Development_FAQ)
* [PDE Does Plug-ins](https://www.eclipse.org/articles/Article-PDE-does-plugins/PDE-intro.html)
* [Creating your first eclipse plugin](https://medium.com/@ravi_theja/creating-your-first-eclipse-plugin-9b1b5ba33b58)

## О задании

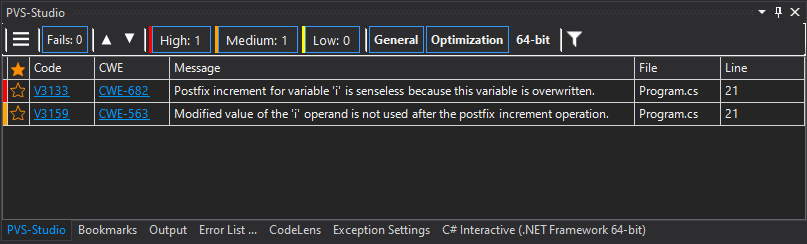
Нужно сделать java плагин для Eclipse IDE, который будет соответствовать минимальным требованиям:

* Меню выбора для открытия/сохранения отчета анализатора
* Окно с загруженными предупреждениями
* Двойным щелчком на предупреждение открывать файл на строке с ошибкой
* Запускать анализ текущего проекта

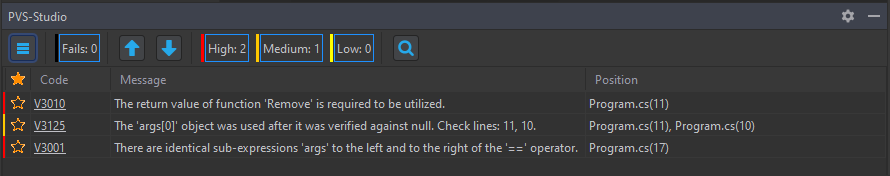
Чтобы иметь представление о том, что из себя представляет отчет анализатора, загляните [сюда](https://github.com/PVSStudioHacks/TulaHack2021/blob/main/sources/report.json)

В качестве визуального представления о плагине вы можете посмотреть на уже реализованные нами: Visual Studio/JetBrains Rider/InteliJ IDEA. Чтобы вживую пощелкать плагины, их можно скачать [здесь](https://www.viva64.com/ru/pvs-studio-download/).

Вот пример того, как это выглядит окно с предупреждениями в Visual Studio 2019:



А вот пример как это выглядит в Rider / IntelliJ IDEA:



Для реализации запуска анализа понадобится сам анализатор PVS-Studio for Java. Его можно скачать вот [тут](https://www.viva64.com/ru/m/0044/#ID3330A9E049).

После того как скачаете, распакуйте архив по адресу:

* Windows: “%APPDATA%/PVS-Studio-Java/”
* \*nix: “~./config/PVS-Studio-Java/”

Для запуска анализа необходимы лицензионные данные. Их вы можете найти [**здесь**](https://github.com/PVSStudioHacks/TulaHack2021/blob/main/sources/PVS-Studio.lic) .

Чтобы анализатор подхватил лицензию, необходимо создать файл “%APPDATA%/PVS-Studio-Java /PVS-Studio.lic” (по умолчанию анализатор ищет лицензионные данные здесь) с двумя строчками:

userName

serialNumber

После того, как вы все сделаете анализатор можно будет найти по пути файл “%APPDATA%/PVS-Studio-Java/${VERSION}/pvs-studio.jar". Вот пример простого запуска анализа:

java -jar pvs-studio.jar -s A.java B.java C.java -e Lib1.jar Lib2.jar -o report.txt -O text

Важные опции:

* ‘-s’ - набор \*.java файлов или директорий для анализа. При необходимости перечисления нескольких файлов/директорий в качестве разделителя используется ' '. Пример: --src "path/to/file1" "path/to/file2" "path/to/dir"
* ‘-e’ - определение classpath (.jar/.class файлы, директории). При необходимости перечисления нескольких сущностей classpath в качестве разделителя используется ' '. Пример: --ext "path/to/file.jar" "path/to/dirJars"

С остальными опциями можно ознакомиться [здесь](https://www.viva64.com/ru/m/0044/#ID3330A9E049).

Примечание. Сообщество написало огромное количество плагинов для Eclipse. P.S. возможно, можно подсмотреть кое-что.