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

## Вводная часть

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! Тем более что все поддержанные IDE довольно тяжеловесны, а Visual Studio еще и не кроссплатформенная. Исходя из этого хочется иметь плагин в [VSCode](https://code.visualstudio.com/). Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Поскольку VSCode полностью состоит из плагинов\расширений, пора взяться за дело и написать свой плагин.

Официальная [страничка](https://code.visualstudio.com/api) VSCode по разработке плагина поможет вам сформировать почву для дальнейших действий. Так же вот [тут](https://github.com/microsoft/vscode-extension-samples/tree/master/helloworld-sample/) доступен простой пример плагина.

## О задании

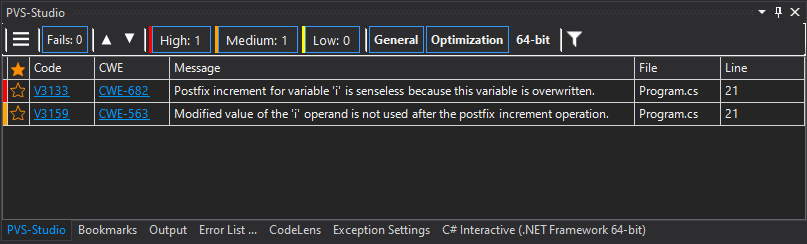
Нужно сделать плагин для VSCode, который будет соответствовать минимальным требованиям:

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

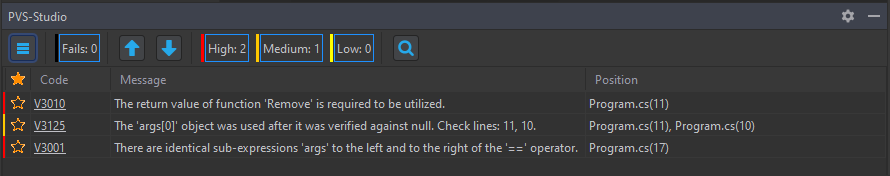
Чтобы иметь представление о том, что из себя представляет отчет анализатора, загляните [сюда](https://github.com/PVSStudioHacks/TulaHack2021/blob/main/sources/report.json).

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 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. Его можно скачать вот [тут](https://www.viva64.com/ru/pvs-studio-download/) (Windows, exe). Для запуска анализа необходимы лицензионные данные. Их вы можете найти [здесь](https://github.com/PVSStudioHacks/TulaHack2021/blob/main/sources/PVS-Studio.lic).

Для того, чтобы ввести полученные данные, выполните:

PVS-Studio\_Cmd.exe credentials --username NAME --serialNumber XXXX-XXXX-XXXX-XXXX

Чтобы реализовать запуск анализа с помощью PVS-Studio нужно воспользоваться утилитой PVS-Studio\_Cmd.exe. После установки PVS-Studio ее можно будет найти по пути "C:\Program Files (x86)\PVS-Studio". Вот пример простого запуска анализа:

PVS-Studio\_Cmd.exe -t "D:\BestProject\BestProjectCpp.sln" \

-o "D:\BestProject\results.plog"

Где в '-t' нужно передать путь до решения или проекта. Во флаг '-o' передается путь до выходного файла, который и будет содержать в себе результаты работы. Подробнее о данной утилите Вы можете почитать [здесь](https://www.viva64.com/ru/m/0035/).