Написать диагностику, которая будет искать ошибку в исходном коде на языке C++

## Вводная часть

Компания PVS-Studio занимается разработкой одноименного статического анализатора. Одним из основных направлений развития анализатора является реализация новых интересных и нетривиальных диагностик. Новые диагностики очень важны, так как они позволяют находить новые ошибки. Поэтому наше задание связано именно с написанием диагностики =)

Чтобы вам было с чего начать, мы подготовили небольшой [проект](https://github.com/PVSStudioHacks/TulaHack2021/blob/main/sources/examples/C%2B%2B.7z), который содержит минималистичный пример диагностического правила на основе clang-tidy. Так же в нем будет приведен пример, для поиска следующей ошибки. В файле project.cpp имеется вот такой код:

auto GetValue()

{

return 42;

}

void Example()

{

auto i = GetValue();

i = i++;

}

В результате работы диагностического правила будет выдано предупреждение на строку 'i = i++;', указывающее на то, что выполнение постфиксного инкремента данной переменной бессмысленно, потому что эта переменная сразу же перезаписывается.

Инструкция по сборке clang-tidy и внедрению диагностики-образца находится в файле *ClangTidyExample/readme.txt*, рядом с исходным кодом диагностики. Инструкция по проверке тестового проекта находится папке с его исходным кодом, в файле *CppProjectForAnalysis/readme.txt*.

## О задании

Нужно реализовать диагностику для поиска ошибки в исходном C++ коде. В результате ее работы должно выводиться сообщение об ошибке с указанием её позиции (файл, строка).

Для выполнения задания допускается использовать другие фреймворки для статического анализа, в том числе основанные на других языках программирования.

Варианты диагностических правил для реализации (**на выбор**):

**1.** Вместо переменной используется строковый литерал, с именем этой переменной.

std::string GetStrValue()

{

return ....;

}

void Error1(const std::string &userName)

{

auto name = GetStrValue();

if (userName == "name") // <=

{

// Do something

}

}

**2.** Cчётчик цикла запустили не в ту сторону.

void Error2(std::vector<std::string> &content)

{

for (int i = 0; i < content.size(); i--) // <=

{

// Do something

}

}

**3.** Сopy-paste ошибка. Сравнивают не те поля структуры, что подразумевались.

struct Data

{

int x;

int y;

int z;

};

void Error3(const Data &a, const Data &b)

{

if (a.x == b.x && a.y == b.y && a.z == b.y) // <=

{

// Do something

}

}

**Приветствуется:**

* Реализация как можно большего количества заданных диагностических правил
* Допустима реализация ваших идей для других диагностических правил, после выполнения вышеперечисленных заданий
* Развитие удобной инфраструктуры (многопоточный анализ, походы ООП и прочее)