Поиск циклических зависимостей с оптимизацией, обеспечивающей проход по каждой ячейке только один раз

В методе **Cell::Set(std::string text)**, если text содержит формулу (text[0] == FORMULA_SIGN && text.size() > 1), то владеть созданным формульным объектом (FormulaImpl) сначала будет временный умный указатель: **std::unique_ptr<Impl> impl_temp = std::make_unique<FormulaImpl>(text.substr(1))**;

С ним произойдет вызов вспомогательного метода **bool Cell::WillCauseCircularDependency(const Impl& impl_temp)**: если вернёт **true**, то выбросится исключение **CircularDependencyException**, **в ином же случае** создание данной ячейки не приведёт к возникновению циклической зависимости, соответственно, **ячейка будет успешно создана**: поле **impl_** ячейки (юник птр) будет владеть данным формульным объектом: **impl_ = std::move(impl_temp)**;

Для поиска циклических зависимостей в однонаправленном графе используем алгоритм на основе **поиска в глубину (DFS)** с дополнительной проверкой на наличие циклов. Для обхода графа необходимо знать ребра, поэтому в классе FormulaImpl требуется добавить публичный метод, который будет возвращать позиции участвующих в формуле ячеек - иначе говоря, позиции ячеек, от которых зависит данная ячейка: назовём его **GetDependencies()** (результат его вызова - это результат вызова метода GetReferencedCells() с объектом типа Formula, указатель на который хранит FormulaImpl).

Обход графа и поиск циклических зависимостей будет происходить в методе **Cell::WillCauseCircularDependency**.

Чтобы избежать повторного прохождения через одну и ту же ячейку, используем два контейнера:
std::unordered_set<const Cell*> visited - для хранения указателей на ячейки, которые уже были посещены.
std::stack<const Cell*> path - для хранения указателей на ячейки, которые составляют путь обхода.
Метод **Cell::WillCauseCircularDependency** имеет доступ к таблице (в классе Cell есть поле sheet&), и получает в качестве аргумента формульный объект (FormulaImpl), таким образом, у него есть всё необходимое, чтобы осуществить поиск в глубину (DFS) и определить, можно ли создать ячейку (не приведёт ли это к циклической зависимости), указывающую на такой формульный объект.
Алгоритм поиска циклических зависимостей (**вызывается, когда GetDependencies() возвращает НЕ пустой контейнер**) будет выглядеть следующим образом: