

Поиск циклических зависимостей

Вызывается `bool CheckForCyclicity(const Impl& impl) const`

Далее проверка наличия зависимых ячеек: `_impl.GetReferencedCells().empty()`

Если список пуст то возвращается false так как нет зависимостей которые могут вызвать циклику

return false

Если список НЕ пуст

создаем хранилище для указателей на соответствующие ячейки
`std::unordered_set<const Cell*> refs`
и заполняем:
`for (const auto& pos : impl.GetReferencedCells())
refs.insert(sheet_.GetCellPtr(pos))`

Далее реализация обхода в глубину

Создается набор `visited` для отслеживания посещенных ячеек и стек `to_visit` для хранения ячеек, которые нужно посетить.
`std::unordered_set<const Cell*> visited`
`std::stack<const Cell*> to_visit`

Начинаем с текущей ячейки (`this`), помещая ее в стек `to_visit`.
`to_visit.push(this)`

1) В цикле `while`, пока стек `to_visit` не пуст, извлекаем текущую ячейку из стека
2) Добавляем в `visited`.

проверяем есть ли она в `refs`

если да то возвращаем true так как найдена циклическая зависимость

Если циклических зависимостей не обнаружено, то для каждой ячейки, на которую ссылается текущая, и которая еще не была посещена, добавляем ее в стек `to_visit` для дальнейшего исследования

когда все ячейки посещены и обработаны и не было найдено циклических зависимостей возвращаем false