## CYCLIC DEPENDENCIES Начапо bool Cell::WouldIntroduceCircularDependency(const Impl& new\_impl) const Проверка, будет ли внесение новой реализации ячейки new impl приводить к циклической зависимости. True new\_impl.GetReferencedCells().empty() Проверка является ли new\_impl независимым std::unordered\_set<const Cell\*> referenced Создаем контейнер for (const auto& pos : new impl.GetReferencedCells()) referenced.insert(sheet .GetCellPtr(pos)) вставка указателя на ячейку в множество referenced, чтобы отслеживать ссылки на другие ячейки, на которые ссылается new\_impl. конец цикла std::unordered set<const Cell\*> visited используется для отслеживания посещенных ячеек std::stack<const Cell\*> to\_visit стек используется для хранения ячеек, которые нужно посетить или обработать to\_visit.push(this) текущая ячейка будет помещена в стек для последующей обработки или посещения while (!to\_visit.empty()) обрабатываем все элементы стека, пока стек не будет пустым const Cell\* current = to visit.top() текущий элемент, который нужно обработать или посетить to\_visit.pop() обработан или посещен, и его можно удалить из стека visited.insert(current) отслеживание посещенных элементов return false True referenced.find(current) != referenced.end() return true for (const Cell\* incoming : current->l\_nodes\_) visited.find(incoming) == visited.end() содержит ли множество visited элемент, указанный в переменной incoming to\_visit.push(incoming) добавляет элемент, указанный в переменной incoming конец цикла for конец цикла while