<<interface>> SheetInterface

- + SetCell(Position pos, string text)
- + GetCell(Position pos) const : const CellInterface*
- + GetCell(Position pos) : CellInterface*
- + ClearCell(Position pos)
- + GetPrintableSize() const : Size
- + PrintValues(ostream& output) const
- + PrintTexts(ostream& output) const

Диаграмма классов для реализации таблицы

PositionHasher

- + operator()(Position pos) const: size t
- + N const: uint64_t = 37

A

Sheet

- + SetCell(Position pos, string text)
- + GetCell(Position pos) const : const CellInterface*
- + GetCell(Position pos) : CellInterface*
- + ClearCell(Position pos)
- + GetPrintableSize() const : Size
- + PrintValues(ostream& output) const
- + PrintTexts(ostream& output) const
- + GetConcreteCell(Position pos) const : const Cell*
- + GetConcreteCell(Position pos): Cell*
- HasCell(Position pos) const : bool
- HasConcreteCell(Position pos) const : bool
- PrintCells(ostream& output, const function<void(const CellInterface&)>& printCell) const
- cells : unordered map<Position, unique ptr<Cell>, PositionHasher>
- row to cell count : map<int, int>
- column to cell count : map<int, int>

Алгоритм добавления новой ячейки/изменения существующей ячейки

Если в метод "Sheet::SetCell(Position pos, std::string text)" будет подан pos некорректный (pos.IsValid() = false), то приложение бросает исключение InvalidPositionException.

Далее будем считать, что в метод "Sheet::SetCell(Position pos, std::string text)" подается корректная позиция (pos.IsValid() = true).

- 1. Если в таблице нет ячейки по позиции pos:
 - 1.2. Создаем ячейку
 - 1.3. Обновляем количество установленных ячеек в строке pos.row и колонке pos.col используя row_to_cell_count_, column_to_cell_count_ соответственно (для быстрого расчета печатной области).
- 2. Устанавливаем текст в созданную ячейку cell, используя метод "Cell::Set(string text)"
 - 2.1. Проверяем, совпадает ли текущий текст в ячейке с text (для того, чтобы при вызове метода "Cell::Set(string text)" с одним и тем же text ничего не делать)
 - 2.2. Проверяем, является ли text формулой
 - 2.2.1. Если текст формула:
 - 2.2.1.1. Получаем формулу по text используя "std::unique_ptr<FormulaInterface> ParseFormula(string expression)"
 - 2.2.1.2. Проверяем формулу на наличие циклических ссылок используя методы
 - "Cell::CheckCircularDependency", в которых выполняется проход DFS по ячейкам, от которых зависит формула (внутри используются методы "vector<Position> Cell::GetReferencedCells()", "Cell* Sheet::GetConcreteCell(Position pos)").
 - 2.2.1.2. Если есть циклическая зависимость в formula приложение бросает исключение CircularDependencyException
 - 2.3. Сбрасываем кэш текущей ячейки, а также всех ячеек, которые зависят от текущей контейнер "unordered set<Cell*> Cell::cells from ".
 - 2.4. Получаем список ячеек, от которых зависит текущая ячейка и убираем связь (от тех ячеек к тек. ячейке)
 - 2.5. Если текст формула:
 - 2.5.1. Определяем "impl " объектом unique ptr<Cell::FormulaImpl>
 - 2.5.2. Проходим по всем ячейкам, от которых зависит формула (результат метода "vector<Position> Cell::GetReferencedCells()")
 - 2.5.2.1. Если в таблице еще нет ячейки (не был вызван метод "Sheet::SetCell(Position pos, std::string text)"), которая необходима формуле создаем пустую ячейку (метод "Sheet::CreateEmptyCell(Position pos)") и устанавливаем у неё связь на тек. ячейку. ВАЖНО! Если позиция ячейки, от которой зависит формула не валидна ячейка не будет создана в таблице (не будет вызова "Sheet::CreateEmptyCell(Position pos)"). Это сделано для того, чтобы потом при получении значения тек. ячейки было значение "FormulaError" с "Category::Ref".
 - 2.6. Если текст не формула: определяем "impl_" объектом unique_ptr<Cell::TextImpl>

Диаграмма классов для реализации ячейки таблицы <<interface>> CellInterface + GetValue() const : variant<string, double, FormulaError> + GetText() const : string + GetReferencedCells() const : vector<Position> Cell + GetValue() const : Value + GetText() const : string + GetReferencedCells() const : vector<Position> + Set(string text) + Clear() + IsEmpty() const : bool - ClearValueCache() - CheckCircularDependency(const vector<Position>& referenced cells) const : bool - CheckCircularDependency(Position cell_position, vector<Position>& checked_cells) const : bool - impl_: unique_ptr<Impl> - value_cache_: mutable optional<Value> - cells_from_: unordered_set<Cell*> - sheet_: Sheet&

