

Для инвалидации кеша в классе Cell появляются дополнительные приватные поля

```
bool is_valid_ //валидность вычисленного значения
```

```
Value temp_ //кеш вычисленного значения
```

В Классе Sheet появляется дополнительное приватное поле

```
std::unordered_map<Position,vector<Position>> cell_dependencies
```

```
//словарь, отображающий при расчете каких ячеек участвует ключевая ячейка.
```

Когда мы пытаемся инициализировать формулой, содержащей ссылки на другие ячейки, какую-то ячейку (пока временную, которую в случае отсутствия циклических зависимостей будет заменена с требуемой), мы проверяем её на наличие в качестве ключа cell_dependencies. Если она там есть, то для соответствующих значений позиций ячеек, сохраненных под ключом, переводим is_valid_ в значение false с помощью метода Validate(bool valid). Затем, берем позиции ячеек, участвующих в вычислении формулы (они есть в списке FormulaAST, к которому мы получаем доступ через метод std::vector<Position> GetReferencedCells()), добавляем их в качестве ключей cell_dependencies, а в качестве значения, добавляем позицию ячейки, в которой происходит расчет формулы (т.е мы таким образом показываем, какая ячейка влияет на другие). Значение is_valid_ является триггером, для получения значения из ячейки. Если оно true, то мы берем temp_ значение, сохраненное при первом запуске Cell::GetValue(). Если false, то повторно вычисляем через evaluate формулу.

Если ячейку удаляем, опять проверяем её наличие в качестве ключа в cell_dependencies. Переводим в переводим is_valid_ в значение false для всех значений позиций ячеек, сохраненных под ключом. Удаляем ключ из cell_dependencies. Удаляем ячейку. Если эта ячейка участвует при расчете формул других ячеек - ничего страшного, она будет интерпретироваться как 0.