

Расчет формулы через дерево вычислений практически не меняется, с той лишь разницей, что в узлах (листьях) появляется новый класс `CellExpr : Expr`, хранящий позицию ячейки участвующей в формуле и перегруженного метода `Evaluate`, принимающего указатель (ссылку на лист), в котором находится ячейка.

Метод `CellExpr :: Evaluate(const SheetInterface& sheet)` будет иметь следующий алгоритм {

Внутри объекта сохранена позиция ячейки `Position cell_`

Проверяем ячейку на вхождение в диапазон листа:

```
if (!cell_.IsValid()) throw InvalidPositionException("#REF!");
```

Если вошла в диапазон, проверяем заполнена ли она (есть ли эта позиция в множестве заполненных позиций листа и )

```
if (cells_posit.count(cell_) == 0) return 0.0;
```

если заполнена

```
else {
```

находим ячейку

```
auto cell_ptr = sheet.GetCell(Position pos); // Вот для этого и
```

передавалась ссылка на лист

извлекаем значение

```
Cell::Value value = cell_ptr->GetValue();
```

если значение строка , содержит ли она что-то и можно ли её

преобразовать в число

```
if (std::holds_alternative<std::string>(value)) {
```

```
std::string val_str = std::get<std::string>
```

если пустая строка - возвращаем 0

```
if (val_str.empty()) return 0.0;
```

```
try{
```

если можно - возвращаем число

```
(value);
```

```
double val_doub = stod(val_str);
```

```
return val_doub;
```

```
}
```

```
catch(...) {
```

если нет - кидаем исключение

```
throw FormulaError("#VALUE!");
```

```
}
```

```
}
```

если при подсчете содержимого ячейки произошла ошибка, извлекаем эту ошибку и кидаем её дальше

```
else if (std::holds_alternative<FormulaError>(value)) throw
```

```
std::get<FormulaError>(value);
```

если все нормально, возвращаем извлеченное число

```
else return std::get<double>(value);
```

```
}
```

Так же, возможно в методе `Evaluate` класса `BinaryOpExpr` `final : public Expr` и `class UnaryOpExpr` `final : public Expr` внести обработку и переброс выкидываемых исключений в `operand_->Evaluate()`, `lhs_->Evaluate()` , `rhs_->Evaluate()` .

Внутри метода `Evaluate` класса `BinaryOpExpr` `final : public Expr` может выброситься исключение `FormulaError {"ARITHM"}`.