**Общее описание вычислителя**

Вычислитель представляет собой эмулятор архитектуры ТТА. Он представлен в виде матрицы с заданными операциями в её ячейках. Для работы с матрицей существуют основные операции: Set, Mov, Mvc, Eps. Для создания матрицы нужно указать тип, с которым она будет оперировать и функции, которые будут в ячейках. Функции распределяются по столбцам матрицы, т. е. в матрице столько столбцов, сколько функций было задано. Подсчет выражения выполняется в несколько потоков, количество потоков определяется количеством функций, записанных в строке. Если в каждой строке разное количество функций, то код не будет выполняться. Также нельзя в разных потоках использовать одну и ту же ячейку.

**Операция Set**

Set x y argument – операция присваивания ячейке х – строки, у – столбца аргумента argument.

**Операция Mov**

Mov x y a b – операция перемещения значения, с выполнением операции из a – строки, a – столбца в x – строку, y - столбец

**Операция Mvc**

Mvс x y argument – операция выполнения функции х – строки, у – стоблца с аргументом argument

**Операция Eps**

Eps – ничего не выполняется. Используется для поддержания параллельности кода.

**Возможные ошибки**

**In line <i> сan’t find Cell <x> <y>** В матрице нет такого значения.

**not known lexeme** ошибка ввода кода.

**code content empty line** в коде присутствует пустая строка.

**can't parse this code, mismatch some elements** отсутствуют некоторые обязательные символы.

**in line <i> can’t create cell <x> <y>** данной ячейки не существуют в текущем вычислителе.

**in line <i> can’t write and read together in equal cells** во время одного такта производится одновременная запись или чтение из одной ячейки.

**In line <i> workflows aren't compatible** – Нарушение многопоточности. Потоки несовместимы.

**сan't do operation** – Невозможно выполнить операцию (деление на ноль).

**Ввод операций в вычислитель**

**Общее описание**

Каждая операция отделяется символом <**;**> Для отделения тактов используется перенос строк. Следующий рисунок отображает определение тактов и правила написания операций (Для наглядности все операции разделены чертами).



**Общее описание модулей проекта**

**module Cell**

Модуль содержит в себе описание ячейки вычислителя.

**type Cell<'a>**

**Конструктор**

|  |  |
| --- | --- |
| Имя | Описание |
| Cell<’a> (operation : ‘a -> ‘a -> ‘a) | Инициализирует новый экземляр класса Cell, параметризованный типом ‘a с операцией в ней. |

**Поля**

|  |  |
| --- | --- |
| Имя | Описание |
| Value | Переменная, хранящая значение типа ‘a, по умолчанию стандартное значение. |
| Operation | Переменная, хранящее переданную конструктором операцию |

**Методы**

|  |  |
| --- | --- |
| Имя | Описание |
| RunOp(operand) | Выполняет функцию ячейки, где первый аргумент – значение ячейки, второй аргумент – operand. |

module Processor

Модуль содержит в себе описание матрицы и операций над ней.

**type Matrix<'a>**

**Конструктор**

|  |  |
| --- | --- |
| Имя | Описание |
| Matrix<'a>(functions: ('a -> 'a -> 'a) array) | Инациализирует экземпляр класса Matrix, параметризованный типом ‘a количество колоной и их содержимое задается массивом функций. В каждой ячейке колонки соответствующая ей операция. |

**Методы**

|  |  |
| --- | --- |
| Имя | Метод |
| ValueInCell row col | Возвращает значение в ячейке row строки, col столбца |
| Dispose | Очищает все ячейки |
| getMatrix | Возвращает матрицу |
| NumCols | Возвращает количество столбцов матрицы |
| NumRows | Возвращает количество строк матрицы |
| CreateSetCells col | Возвращает позиции и значение созданных в столбце ячеек |
| WorkFlows (program: Program<'a>) | Проверяет совместимость потоков в массиве оперций |
| Check (line: array<Asm<'a>>) | Проверяет обращение к одной и той же ячейке в одном такте операций |
| RunLine (line: array<Asm<'a>>) | Выполняет один такт |
| RunOp (program: Program<'a>) | Выполняет все вычисления |

**module TTA.ASM**

Модуль содержит в себе тип Asm<’a> описания команд интерпретатора и тип Program<’a>. type Program<’a> представляет собой Asm<’a>[][] массив массивов команд.

**type Asm<'a>**

**Поля**

|  |  |
| --- | --- |
| Имя | Описание |
| Set of (int<ln>\*int<col>)\*'a | Операция присваивания ячейки |
| Mov of (int<ln>\*int<col>)\*(int<ln>\*int<col>) | Операция переноса значения с выполнением операции |
| Mvc of (int<ln>\*int<col>)\*'a | Операция присваивания с выполнением операции |
| Eps | Операция ничего не делает |

**Namespace IDE**

**Form1.cs**

Главное окно среды.

**Объекты**

|  |  |
| --- | --- |
| Имя | Описание |
| richTextBox1 | Поле для ввода операций над матрицей |
| Button1 | Кнопка открытия сохраненных операций |
| Button2 | Кнопка сохранения операций |
| Button3 | Кнопка для выполнения операций |
| Button4 | Кнопка деббагинга (построчное выполнение операций) |
| Button5 | Кнопка остановки деббагинга |
| Data | Структура отображающая состояние ячеек матрицы, к которым было обращение |
| errorBox | Структура, выводящая список ошибок |

**События**

|  |  |
| --- | --- |
| Имя | Описание |
| Load | Открытие файла кода |
| Save | Сохранение файла кода |
| Start | Запуск кода |
| Debug | Запуск деббагинга |
| Stop | Остановка деббагинга |
| Close | Предложение сохранение файла при закрытии |