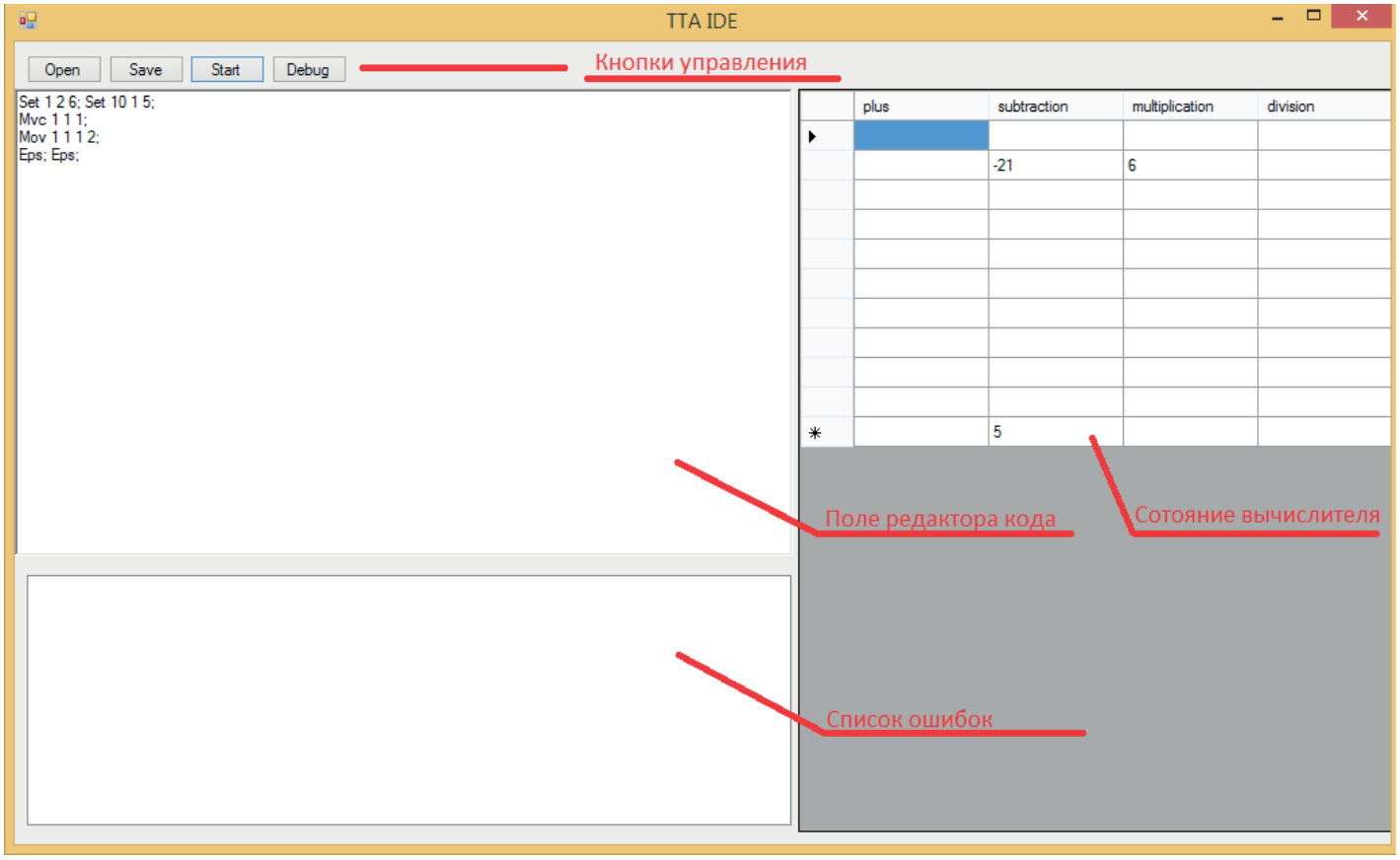
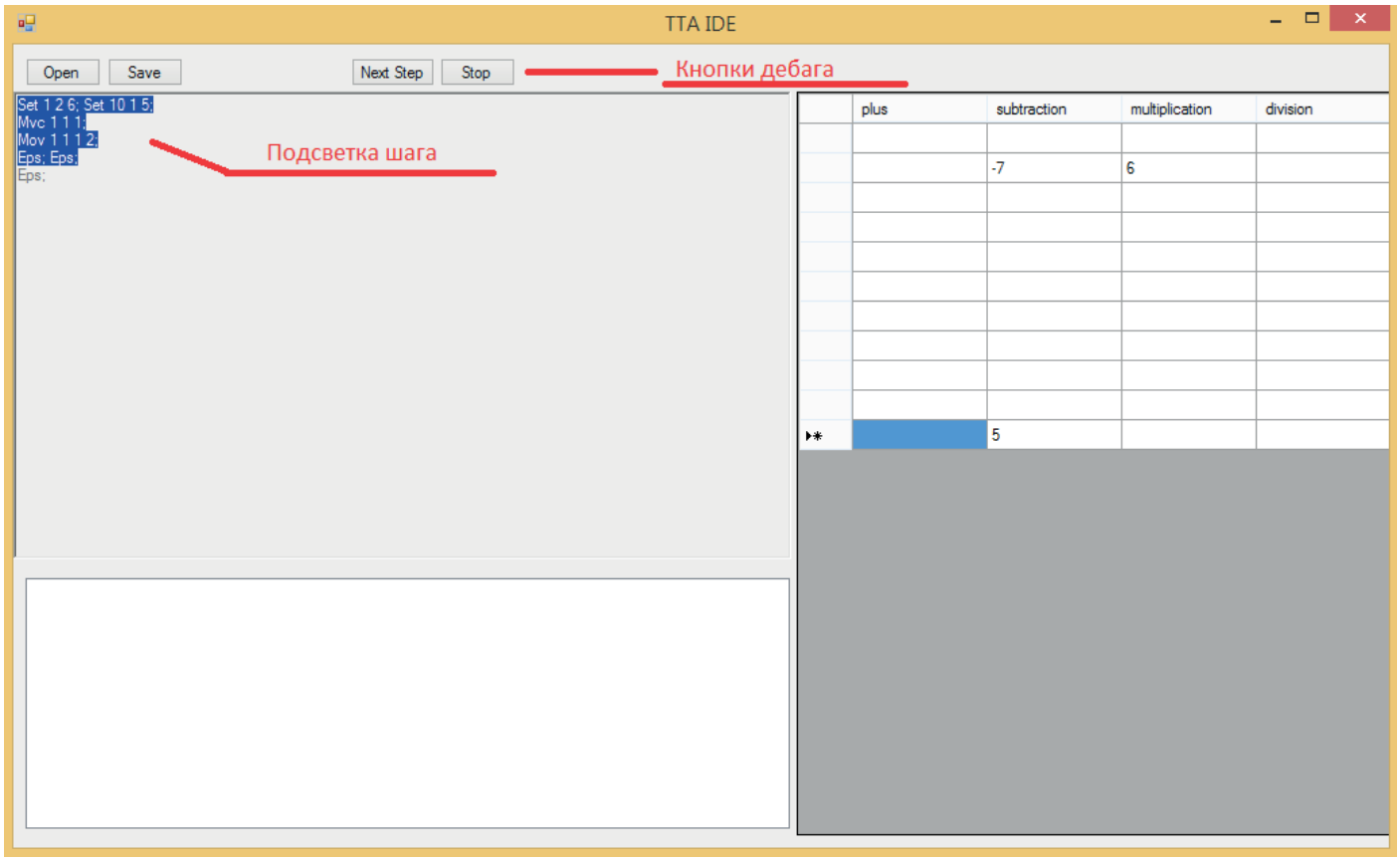
Докуметация к IDE и внутренней реализации вычислителя, компилятора, парсера.

Автор: Булгаков А.В.

Главное окно IDE



Главное окно IDE при выполнении Debug



Описание модулей проекта

module Processor

Модуль содержит в себе описание вычислителя реализующего эмуляцию архитектуры TTA. Processor представляет собой двумерную матрицу, колонки которой реализуют заданные на этапе создания лямбда функции и строки которой не ограничены.

type Processor

Конструкторы

Имя	Описание
Processor((f : array<'a -> 'a -> 'a>))	Инициализирует новый экземляр класс Processor с заданными в столбцах функциях

Методы

Имя	Описание
ValueAt row col	Возвращает значение содержащееся в row строке, col столбце
Check(arr : Asm<'a>[])	Проверяет подаваемы массив команд на наличие RunTime исключений
countCells	Возвращает количество непустых ячеек в матрице
executeLine(arr : Asm<'a>[])	Выполняет подаваемый на вход массив команд
executeProgram(arr : Program<'a>)	Выполняет заданную программу
getGrid	Возвращает матрицу ячеек
Dispose	Очищает ячейки матрицы

module Cell

Модуль содержит в себе описание ячейки вычислителя.

type Cell

Конструкторы

Имя	Описание
Cell<'a> (operation : 'a -> 'a -> 'a)	Инициализирует новый экземляр класса Cell с лямбда фунцией в ней

Поля

Имя	Описание
Value	Получает или задает значение в даной ячейке

Методы

Имя	Описание
Execute arg	Выполняет функцию в ячейки от значения ячейки и arg

module TTA.ASM

Модуль содержит в себе тип Asm<'a> описания комманд интерпритатора и тип Program<'a>.

type Program<'a> представляет собой Asm<'a>[][] массив массиво команд.

type Asm

Конструкторы

Имя	Описание
Set ((line : int<ln>, col : int<col>)*, x :'a)	Записать в ячейку (line, col) значение x
Mov((l1 : int<ln>, c1: int<col>), (l2 : int<ln>, c2 : int<col>))	Взять значение из ячейки (l2, c2) выполнить с ним операцию из ячейки (l1, c1) и записать результат в (l2, c2)
Mvc ((line : int<ln>, col : int<col>)*, x :'a)	Выполнить операцию из (line, col) с аргуметом x
Eps	Ничего не делать

module Compiler

Модуль содержит в себе описание компилятора для вычислителя. Является связующим звеном между IDE и остальными частями программы. Выполняет роль инкапсуляции в солюшене. Таким образом можно выстраивать IDE оперируя тоолько этим классом.

type Compiler

Конструкторы

Имя	Описание
Compiler()	Инициализирует новый экземляр класса Compiler.

Методы

Имя	Описание
getErrorsList()	Возвращает HashSet<(string*int*int)> содержащую Теккст ошибки и номера строки и операции в которой произошла ошибка.
Compile(str : string)	Компилирует строку в Program<'a> для дальнейших операций.
Run()	Выполняет скомпилированный Program<'a>
Step(i :int)	Выполняет i-ый массив в скопилирванной программе
Stop()	Очищает скомпилированную программу и выполняет Dispos
getGrid()	Возвращает матрицу ячеек.
getStringGrid(num:int)	Возвращает словарь ячеек вида Dictionary<int, string> из i-ого столбца таблицы.
CountRows()	Количество ненулевых строк.
CountCols()	Количество столбцов.

module MyParser.MyParser
Модуль содержит в себе описание парсера из String в Program<'a>

type MyParser
Конструкторы

Имя	Описание
MyParser()	Инициализирует новый экземляр класса MyParser.

Методы

Имя	Описание
Parse (input: string)	Преобразует входную строку в Program<'a> для дальнейшей работы с вычислителем.

Синтаксис языка

Команды вводятся через « ; »

Список комманд можно посмотреть в описании модуля Asm

Примеры команд:

```
//Положить в ячейку (0,0) значение 1
Set 0 0 1;
//Положить в ячейку (1,1) результат операции от значения в ячейки и 2
Mvc 1 1 2;
//Положить в (0,0) результат выполнения операции ячеек (0,0) (1,1)
Mov 0 0 1 1;
//Ничего не делать
Eps;
```