

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Пензенский государственный университет» (ПГУ)

В. В. Рееда, О. Н. Рееда

Программирование в SimInTech

Методические указания
к лабораторным работам

Пенза
Издательство ПГУ
2023

УДК 001.8 (083.95)

P13

Р е ц е н з е н т

доктор технических наук, профессор,
академик Российской метрологической академии,
заместитель директора Государственного регионального центра
стандартизации, метрологии и испытаний в Пензенской области

А. А. Данилов

Регада, Владимир Викторович.

P13 Программирование в SimInTech : метод. указания к лаб.
работам / В. В. Регада, О. Н. Регада. – Пенза : Изд-во ПГУ,
2023. – 76 с.

Рассмотрены основы программирования на языке, встроенном в систему модельно-ориентированного проектирования SimInTech, с помощью которого можно написать программу (скрипт) для выполнения в процессе расчета определенных действий с объектами схемы при наступлении некоторого события, а также для задания глобальных или локальных констант и переменных для текущей страницы проекта. Описания лабораторных работ снабжены методическими указаниями, содержащими необходимый теоретический материал по соответствующей тематике, и списком контрольных вопросов.

Издание подготовлено на кафедрах «Электроэнергетика и электротехника» и «Информационно-измерительная техника и метрология» Пензенского государственного университета и предназначено для лабораторных и индивидуальных занятий по курсу «Компьютерные технологии» для обучающихся по направлениям подготовки 13.03.02 «Электроэнергетика и электротехника», 12.03.01 «Приборостроение», 27.03.01 «Стандартизация и метрология» и специальности 27.05.02 «Метрологическое обеспечение вооружения и военной техники».

УДК 001.8 (083.95)

© Пензенский государственный
университет, 2023

СОДЕРЖАНИЕ

Введение.....	4
Лабораторная работа № 1. Знакомство со средой и языком программирования SimInTech.....	5
Лабораторная работа № 2. Разветвляющиеся и циклические программы в SimInTech.....	29
Лабораторная работа № 3. Работа с массивами в среде SimInTech.....	40
Лабораторная работа № 4. Визуализация данных в среде SimInTech.....	57
Список литературы	75

ВВЕДЕНИЕ

В настоящее время для автоматизации математических расчетов и моделирования в инженерной практике используются различные системы, такие как Mathcad, MATLAB и др. Среди них можно выделить российскую систему модельно-ориентированного проектирования SimInTech. Эта система представляет собой среду, позволяющую создавать математические модели, алгоритмы управления, интерфейсы управления и автоматической генерации кода для программируемых контроллеров и графических дисплеев [1].

SimInTech обладает широкими возможностями по организации вычислений, связанных с решением алгебраических и обыкновенных дифференциальных уравнений. Расчетные математические модели в SimInTech создаются посредством функционально блочного программирования при помощи блоков, которые содержатся в различных библиотеках [2].

При функциональном и динамическом моделировании различных систем SimInTech может использовать встроенный специализированный язык программирования высокого уровня SimInTech. Он позволяет написать программу (скрипт) для выполнения в процессе расчета определенных действий с объектами схемы при наступлении некоторого события, а также для задания локальных или глобальных переменных и констант проекта. На языке SimInTech описывается функционирование типового блока «Язык программирование» из палитры Динамические.

Язык программирования содержит встроенные ключевые слова, константы, декларации, операторы и функции/процедуры. Основной отличительной особенностью языка программирования SimInTech является то, что текст программы предназначен для исполнения на каждом расчетном шаге при моделировании схемы [2].

SimInTech по скорости расчета сложных или математически жестких моделей превосходит зарубежные программы для моделирования на 20 %. Набор методов решения дифференциальных уравнений содержит как классические методы, используемые в конкурирующих программных продуктах, так и авторские, обеспечивающие преимущества при расчете сложных систем [3].

Лабораторная работа № 1

ЗНАКОМСТВО СО СРЕДОЙ И ЯЗЫКОМ ПРОГРАММИРОВАНИЯ SimInTech

Цель работы: знакомство с интерфейсом системы SimInTech и общими сведениями по использованию блока «Язык программирования» для написания программ на языке программирования SimInTech.

Рабочее задание

1. Запустить программу SimInTech, на экране появится главное окно с интерфейсом SimInTech, представленное на рис. 1 [2].

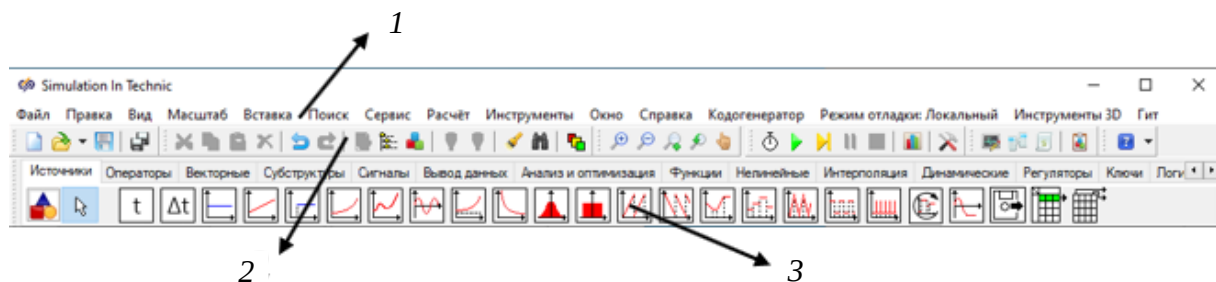


Рис. 1

В первой строке главного окна располагается область **Меню главного окна 1**, откуда, в частности, можно управлять созданием, редактированием и режимами моделирования созданных моделей, которые называются в SimInTech расчетными схемами.

Во второй строке окна располагается область **Панель кнопок 2** с разбитыми по группам наборами кнопок, вызывающих наиболее часто используемые команды.

В третьей строке окна располагается область **Палитра блоков 3**, на закладках которой появляются блоки, соответствующие выбранной расчетной схеме.

2. Для решения конкретной инженерной задачи должна выбираться соответствующая расчетная схема с решателем и своими настройками.

Выбрать в первой строке меню главного окна команду **Файл** → **Новый проект** → **Схема модели общего вида** (рис. 2).

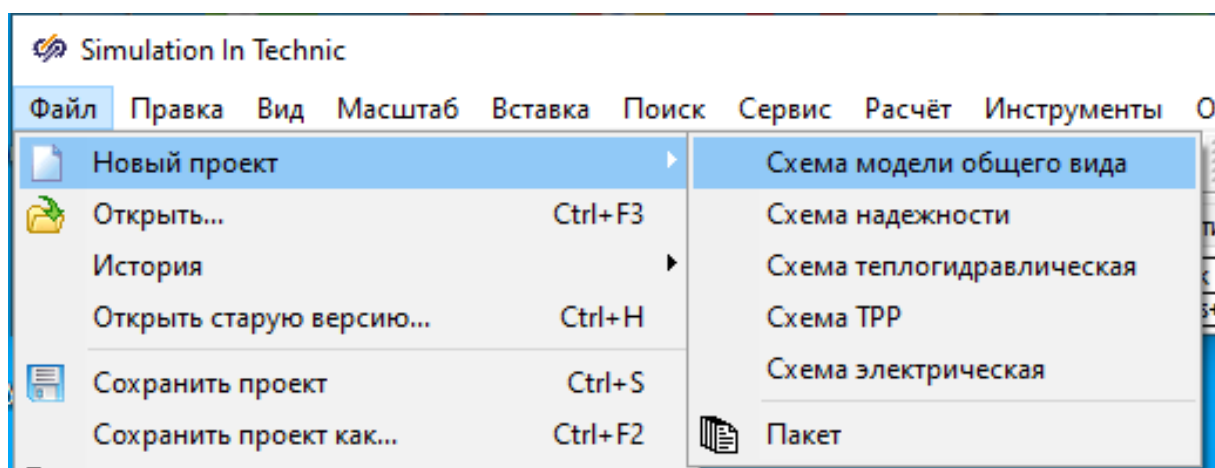


Рис. 2

В результате откроется окно проекта (рис. 3), в заголовке которого указано имя файла **Схема модели общего вида** и расширение файла по умолчанию **prt**.

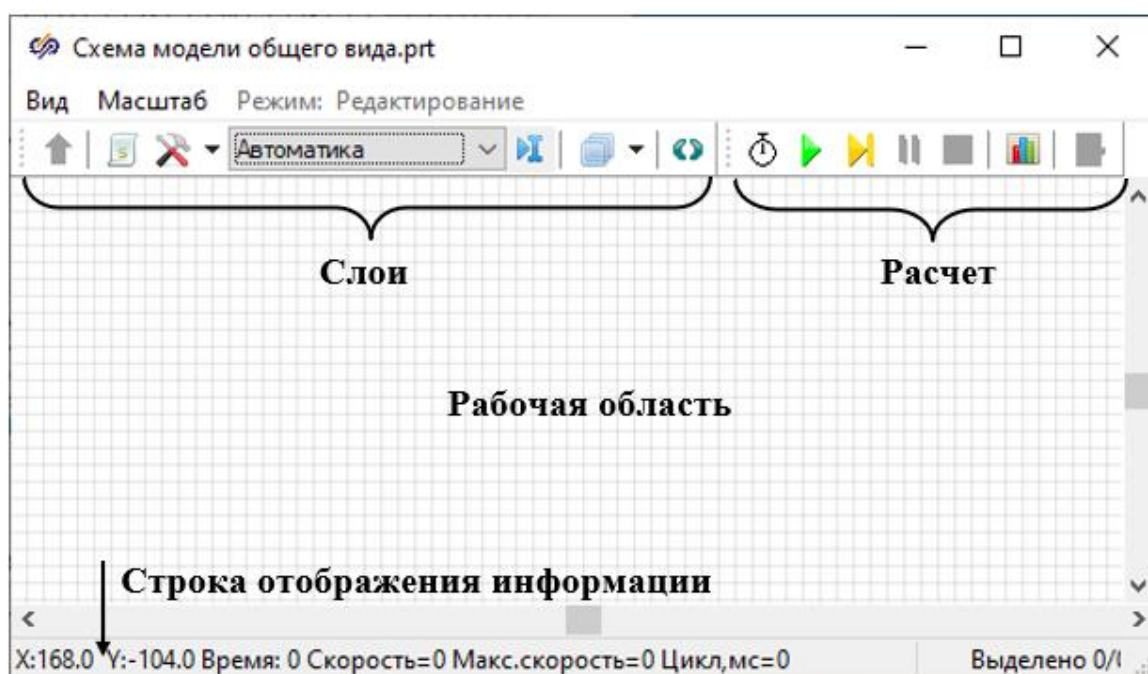


Рис. 3

Панель кнопок сверху окна проекта содержит инструменты настройки проекта «Слои» и управления расчетом «Расчет». Внизу окна проектов расположена строка отображения расчетной информации. В центре окна проектов располагается рабочая область, в которой при помощи блоков, выбираемых из палитры блоков, можно создавать и редактировать ранее созданные расчетные модели, создавать собственные блоки и т.д.

Для выполнения заданных действий с блоками модели, а также для задания локальных или глобальных переменных и констант проекта можно написать программу (скрипт) на языке SimInTech. Для этого необходимо нажать кнопку «Скрипт»  из раздела «Слои» на панели окна редактирования проекта.

В результате откроется окно **Скрипт страницы** (рис. 4), содержащее панель кнопок, поле ввода текста скрипта и панель сообщений внизу окна.

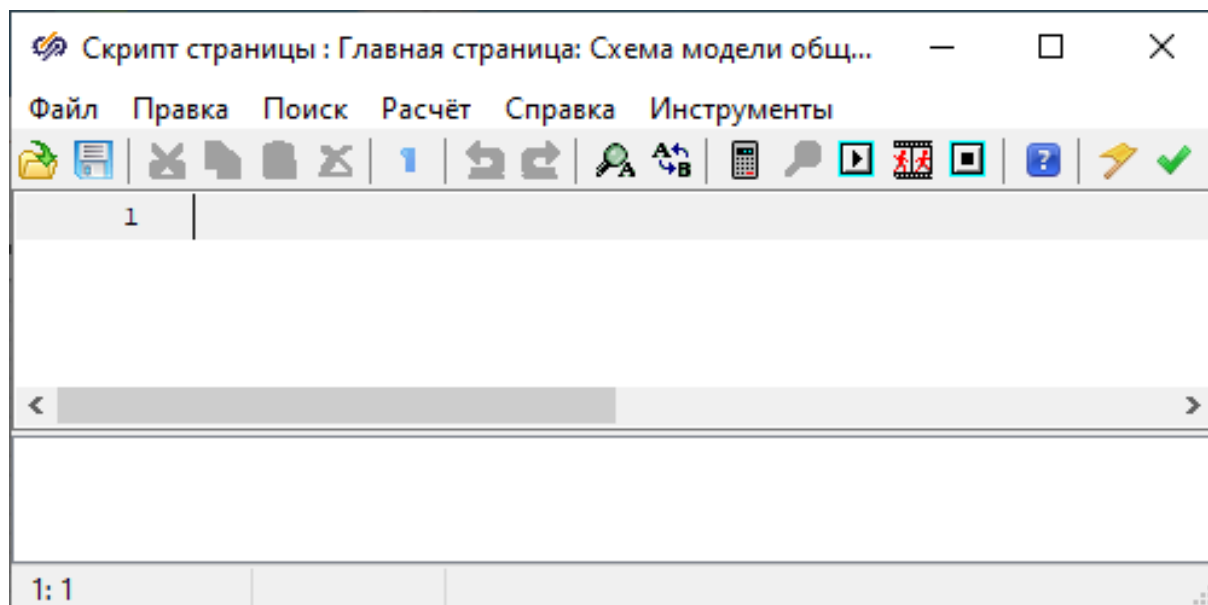


Рис. 4

Набрать в поле ввода текст скрипта, приведенный на рис. 5. Нажать кнопку «Просмотр переменных скрипта» 1 либо кнопку «Просмотреть значение предварительно выделенной переменной» 2, либо кнопку «Выполнить скрипт» 3.

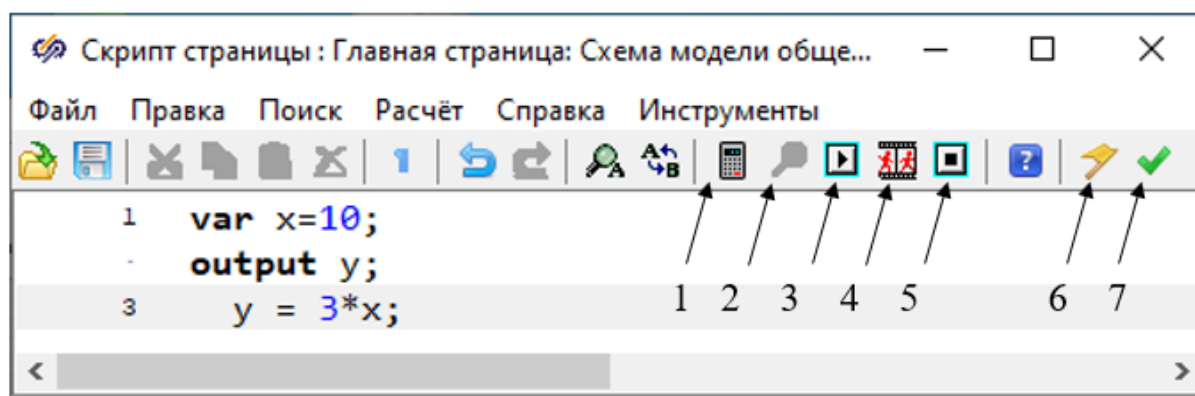


Рис. 5

При этом справа от скрипта программы откроется окно **Просмотр значений переменных** (рис. 6).

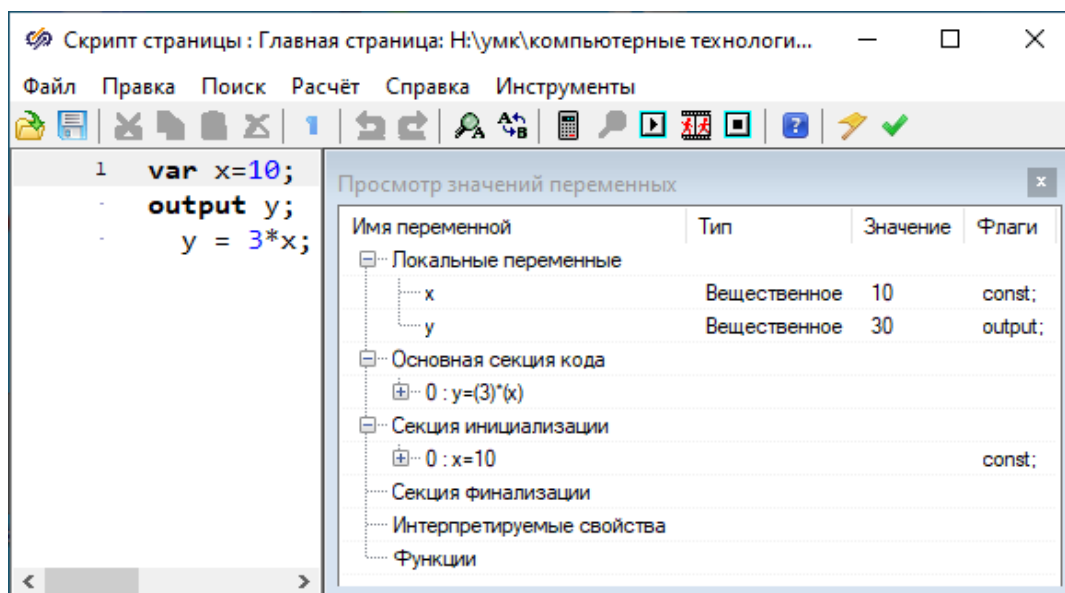


Рис. 6

При необходимости можно запустить пошаговую отладку скрипта, нажав кнопку 4 , и завершить ее отладку, нажав кнопку 5 .

После внесения изменений в программу нужно нажать кнопку «Применить изменения» 6  или на кнопку «Применить изменения и закрыть окно редактирования» 7 .

3. Программу на языке программирования SimInTech можно привязать к конкретному блоку схемного окна проектов, выбрав левой клавишей мыши (ЛКМ) в области **Палитра блоков** на закладке **Динамические** блок **Язык программирования**. Затем переместить курсор в окно проекта и щелчком ЛКМ указать место вставки блока (рис. 7).

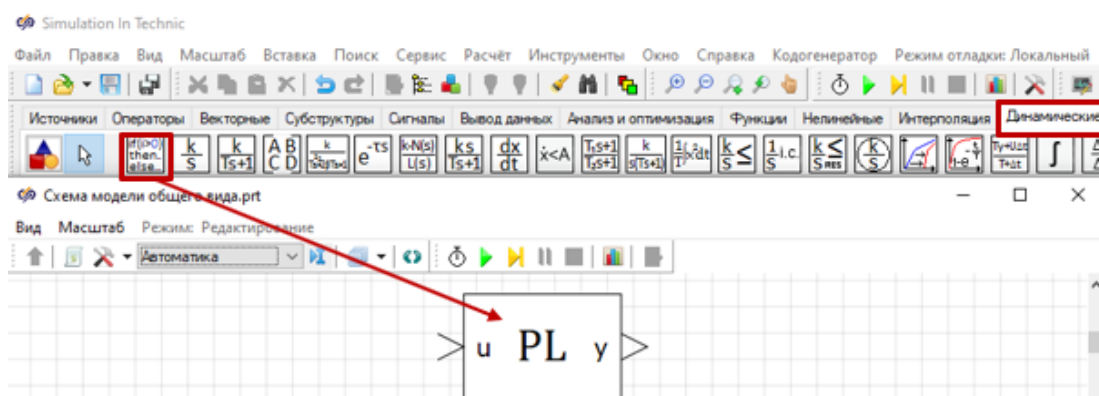


Рис. 7

Блок **Язык программирования** предназначен для создания блоков, выполняющих сложные операции, включая операции над матрицами, векторами, комплексными числами. Внутри диалогового окна блока **Язык программирования**, которое представляет собой окно текстового редактора алгоритма, может быть написана любая программа на языке программирования SimInTech.

Двойной щелчок ЛКМ по Блок **Язык программирования** открывает диалоговое окно блока «**Язык программирования**» (рис. 8) с программным кодом по умолчанию, который определяет вид блока в окне проектов.

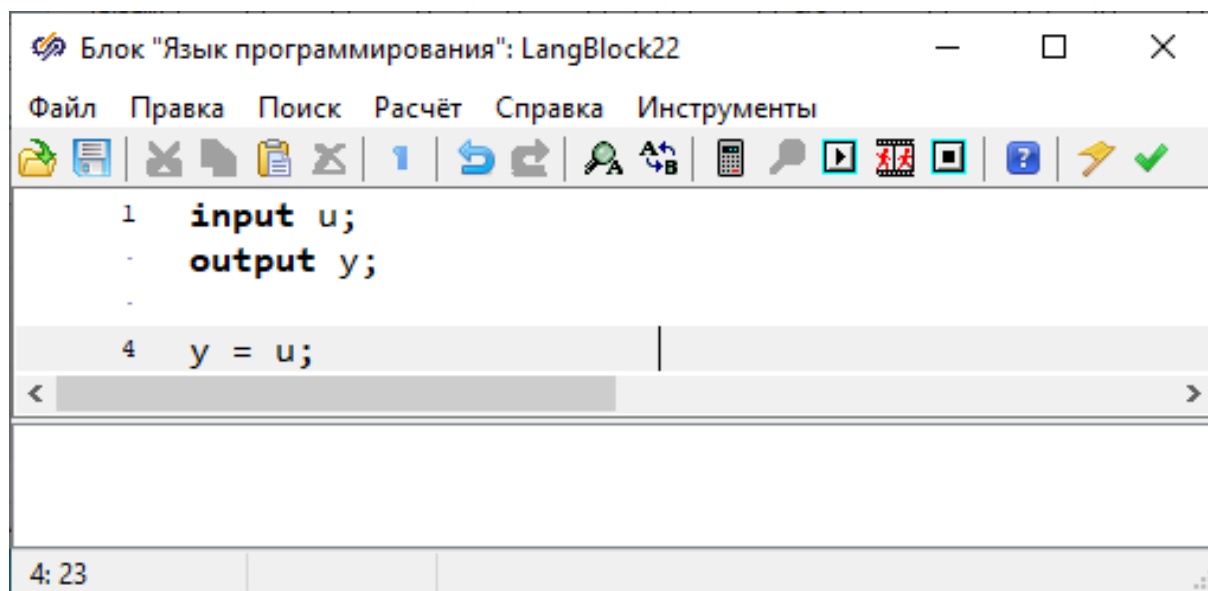


Рис. 8

В программе, приведенной на рис. 8, с помощью ключевых слов **input** и **output** задаются входные **u** и выходные **y** переменные блока соответственно. В результате у блока **Язык программирования** формируются входной **u** выходной **y** порты (см. рис. 7). Сигнал, поступающий на вход блока, передается без изменений на его выход.

Сохранить текст программы в виде текстового файла под именем **лр_1_1** с расширением **ТХТ**, выбрав в диалоговом окне блока **Язык программирования** команду **Файл → Сохранить**.

Сохранить проект в виде файла в папке по указанию преподавателя под именем **лр_1_1** с расширением по умолчанию **prt**, используя в схемном окне проекта команду **Файл → Сохранить проект как...**

4. Ознакомиться с общими сведениями о языке программирования SimInTech при необходимости можно непосредственно из программы,

выбрав в главном меню главного окна команду **Справка → Содержание → Язык программирования** или нажав кнопку $\langle F1 \rangle$ [1].

Программа (скрипт) на языке программирования SimInTech состоит из *деклараций* и *операторов*, которые разделяются между собой точкой с запятой «;». Программа может содержать комментарии, заключенные в фигурные скобки { }, либо начинающиеся с символов «//» и до конца строки.

Входные, выходные и динамические переменные или константы можно использовать в программе только после их описания (инициирования) в декларации или задания им значения с помощью оператора присваивания. Значения остальных переменных, используемых в программе, могут быть рассчитаны в соответствии с выражением, которое присваивается переменной. Декларации рекомендуется помещать в начало программы.

Идентификаторы языка программирования SimInTech (имена констант, переменных, меток, функций и процедур) могут содержать буквы латинского и русского алфавитов, знак подчеркивания и цифры. Идентификатор может иметь произвольную длину и должен начинаться с буквы или знака подчеркивания. Прописные и строчные буквы в идентификаторах по умолчанию не различаются. Не допускается использовать в качестве идентификаторов ключевые слова (эти слова автоматически выделяются при вводе полужирным шрифтом).

Декларации должны начинаться с *ключевых слов*:

const – константы;

var – алгебраические переменные;

init – динамические (дифференциальные) переменные;

input – входные переменные блока;

output – выходные переменные блока.

После ключевого слова перечисляются через запятую имена переменных либо констант, которым с помощью оператора присвоения значения «=» задается начальное значение. В случае нулевого начального значения переменных либо констант указывается только их имя.

Константы отличаются от переменных тем, что они задаются только один раз в декларации. Если константа потом переопределена, то она будет воспринята как переменная.

Открыть диалоговое окно блока **Язык программирования** и ввести в раскрывшемся окне текстового редактора следующий программный код:

```

1  const e=4; //инициализация константы вещественного типа
-  var i=[], //инициализация массива i нулевой длины
-    r=[1,2,3,4]; //инициализация массива r
-  output e,r,i; //инициализация выходной переменной e,r,i
-    i = e/(r);    //основная секция кода
,

```

В программе после ключевого слова **const** иницируется константа **e=4**.

Затем после ключевого слова **var** через запятую иницируются две переменные: одномерный массив **i** переменной размерности и одномерный массив **r** длиной 4, в котором заданы четыре значения его элементов.

Далее в программе после ключевого слова **output** декларируются выходные переменные, значения которых будут выводиться в соответствующие порты блока **Язык программирования**.

В основной секции кода вычисляются элементы одномерного массива **i**.

5. Вернуться в окно проекта, выделить блок **Язык программирования**, нажать правую клавишу мыши и выбрать в появившемся контекстном меню команду **Свойства объекта**.

В появившемся окне (рис. 9) нужно открыть закладку **Общие** и нажать на пиктограмму  в строке **Графическое изображение**.

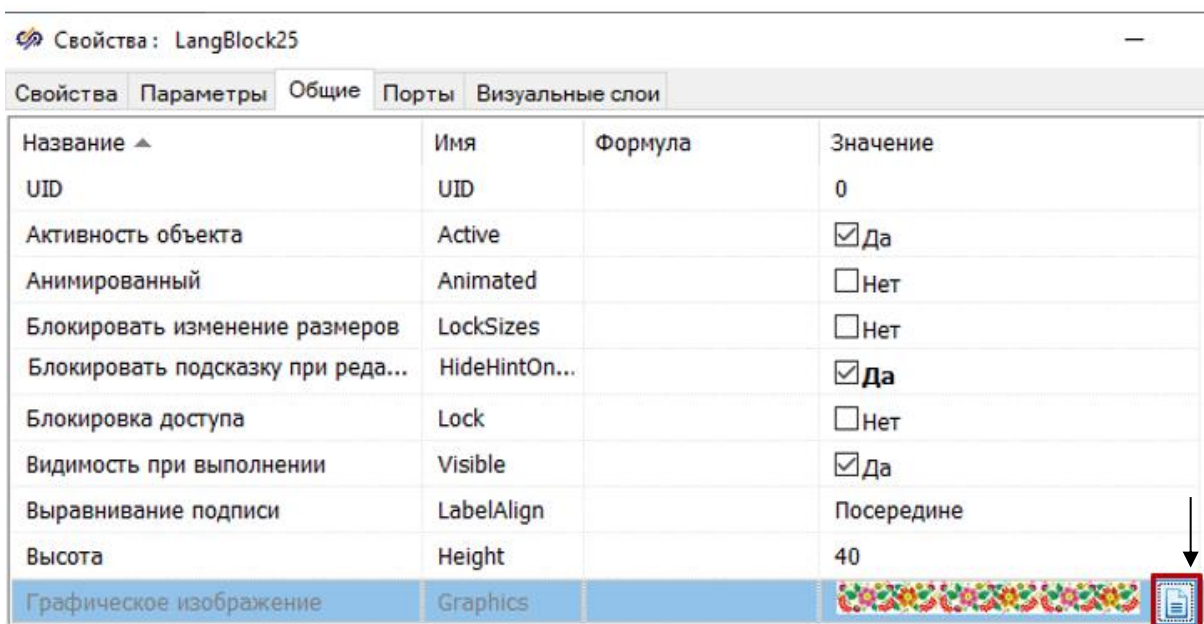


Рис. 9

В результате откроется окно **Графический редактор**, представленное на рис. 10.

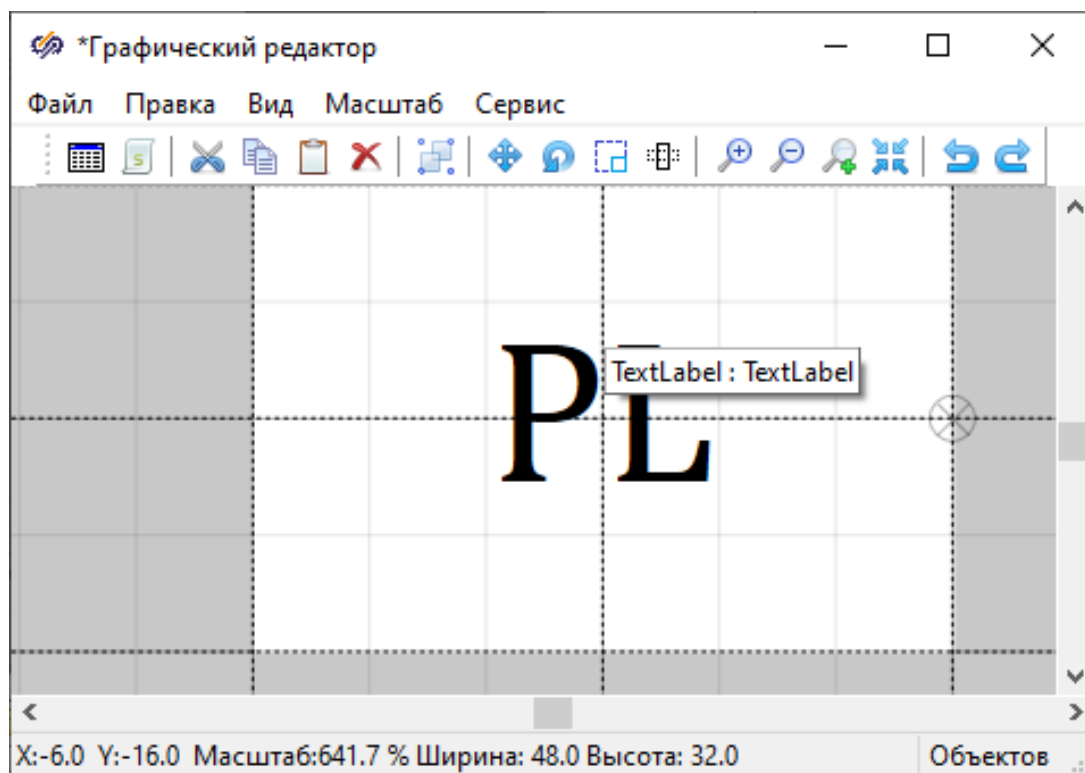


Рис. 10

Выполнить двойной щелчок ЛКМ по тексту и изменить с помощью клавиатуры текст в появившемся окне **Редактирование текста** в соответствии с рис. 11.

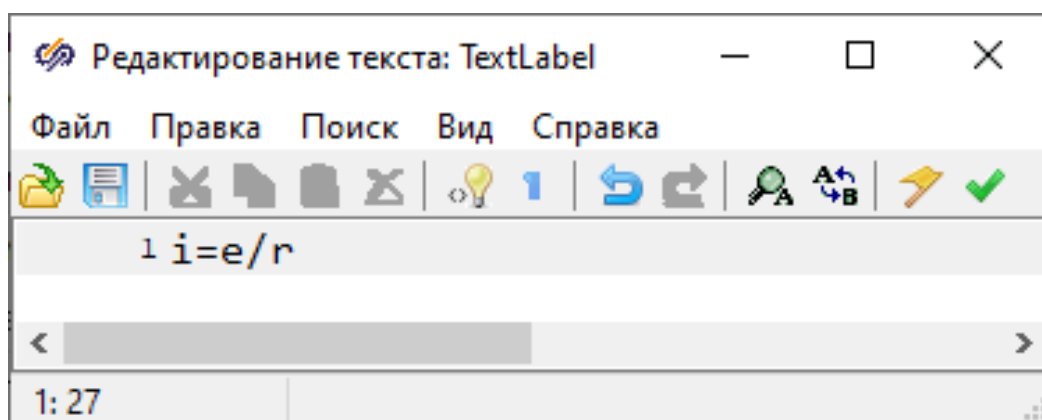


Рис. 11

Для сохранения изменений и закрытия окна редактирования текста нажать кнопку 

6. Вернуться в окно проектов и запустить программу на выполнение, нажав на **Пуск**  на панели инструментов окна проектов или на клавишу F9. Подтвердить запрос на сохранение текста программы. После завершения программы в строке отображения информации появится сообщение **Конечное время достигнуто (time=1)**.

Из рис. 12 видно, как изменится вид блока **Язык программирования** в окне проектов по сравнению с видом этого блока на рис. 7. Помимо того, что изменился текст внутри блока, у него появились три выводных порта, на которые может выводиться информация о значениях выходных переменных блока **e, r, i**.

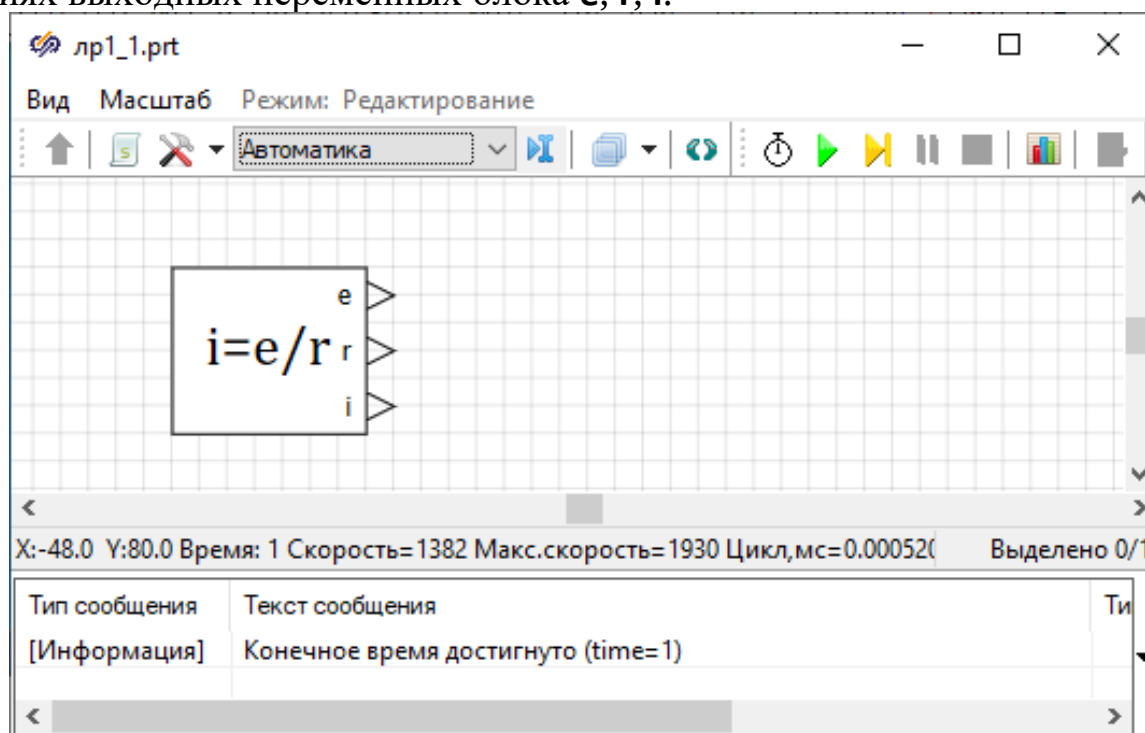


Рис. 12

Для вывода результатов работы программы в окне проектов необходимо поочередно подвести курсор к каждому выходному порту блока, дождаться, когда курсор превратится в стрелку вида  и сделать однократный щелчок ЛКМ.

В результате на выходе порта появится линия связи, которую нужно протянуть вправо от порта, а затем сделать однократный щелчок правой клавишей мыши в конце линии связи.

Если повторно запустить программу на выполнение и нажать клавишу «Показать значения на линиях связи»  в окне проектов, то под линиями связи выведутся соответствующие значения выходных переменных блока **e, r, i** (рис. 13).

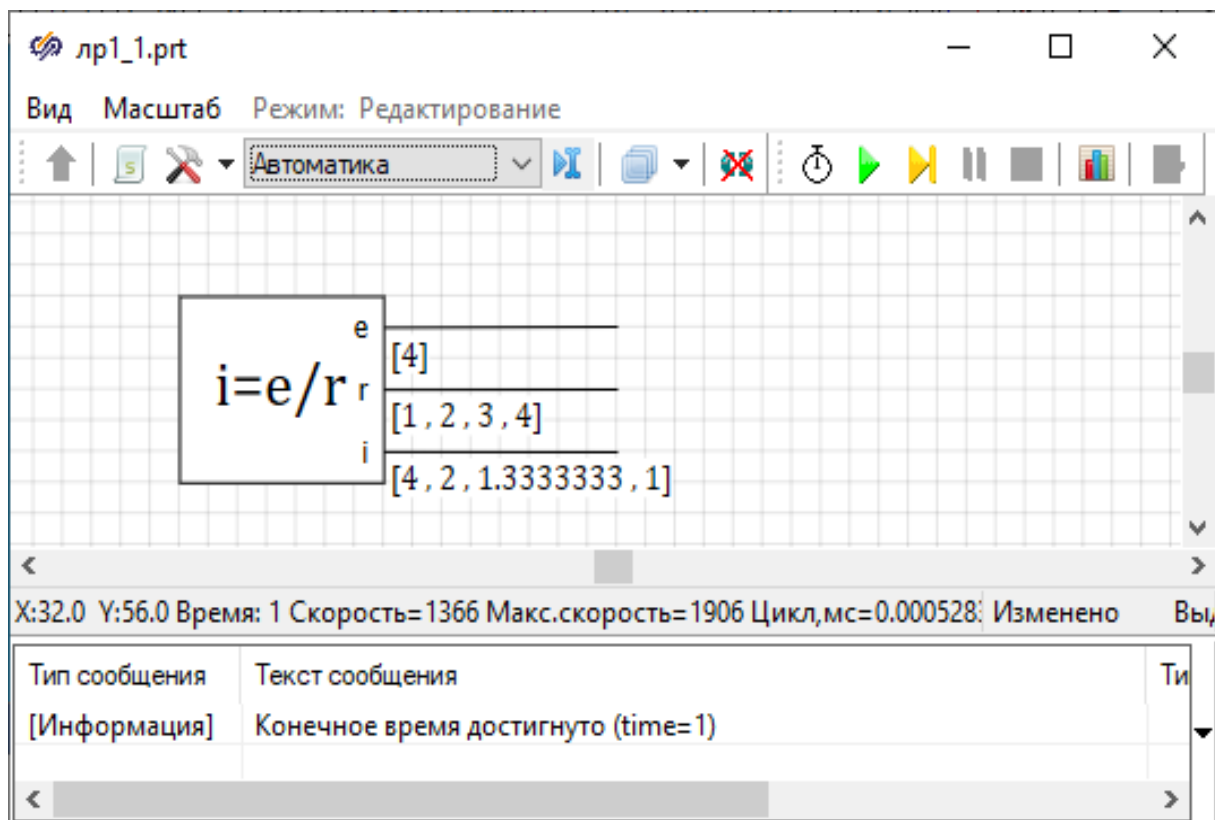


Рис. 13

7. Открыть текстовое окно редактора блока **Язык программирования** и снова запустить программу на выполнение.

В результате справа от текста программы появится окно **Просмотр значений переменных** (рис. 14), в котором будут приведены в виде таблицы имена переменных, их типы, значения и флаги.

Задание для отчета по лабораторной работе



Отредактировать в диалоговом окне блока **Язык программирования** программный код из п. 4, задав значение константы **е**, равным номеру варианта *N* по указанию преподавателя.

Вставить в отчет по лабораторной работе текст полученной программы, а также результаты ее выполнения в схемном окне проекта и в диалоговом окне блока **Язык программирования**.

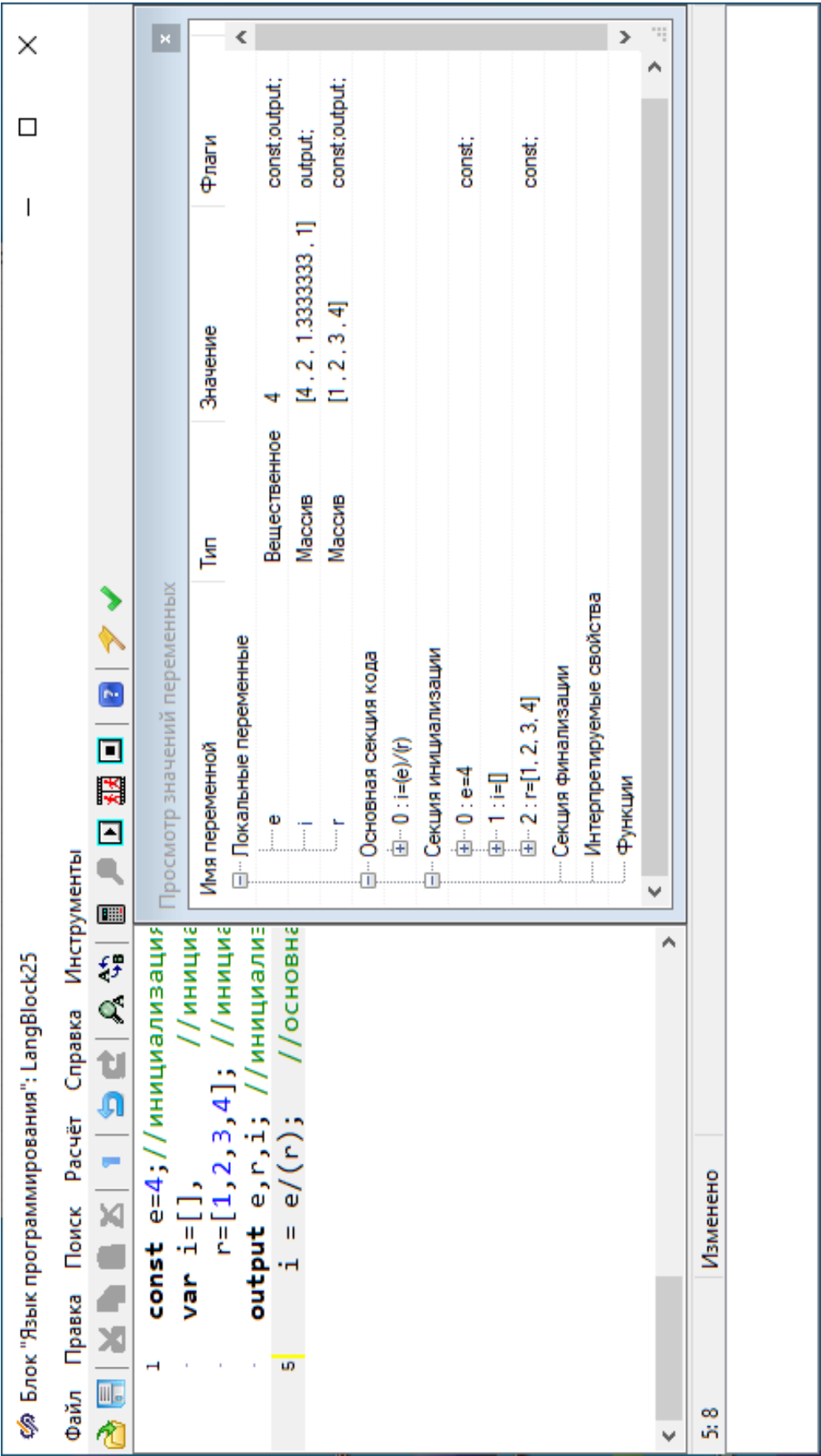


Рис. 14

8. Основным понятием большинства языков программирования является *математическое выражение*, которое строится на основе *чисел, констант, переменных, операторов, функций* и различных *спец-знаков*.

Константы в языке программирования SimInTech делятся на числовые и строковые (символьные). Строковые константы – это любой набор символов в кавычках: "**строка символов**".

Далее в описании форматов команд символами <...> обозначены обязательные идентификаторы, а символами {...} – необязательные дополнительные идентификаторы [2]. Формат задания числовых констант:

{<знак>}<целая часть>{<десятичная часть>} {E<показатель степени>}{i}.

Числовые константы в языке программирования SimInTech, аналогично другим языкам программирования высокого уровня, могут быть целыми, вещественными (дробным с фиксированной и плавающей запятой), а также комплексными.

В качестве десятичного разделителя в числах используется точка, например: **10.5**.

Большие и малые вещественные числа N выводятся в формате с плавающей запятой, который представляется следующим образом:

$$N = M \cdot 10^p,$$

где M – мантисса числа (в нормализованной форме записи мантисса десятичных чисел принимает значение от единицы до девяти); p – порядок числа с учетом знака.

В формате записи с плавающей запятой мантисса числа отделяется от его порядка символом **E** без пробелов. При этом для положительных порядков знак не указывается, а для отрицательных указывается символ «-». Например, число **0,0000123** в формате с плавающей запятой записывается как **1.23E-5**.

В SimInTech символы **i, j, l, J** непосредственно после числа означают, что число является мнимым, например мнимая единица обозначается **1i**. Например, если ввести число **5i**, то SimInTech принимает его как комплексное число **0+5i**, у которого вещественная часть равна 0, а мнимая 5.

Константы, как и переменные, могут быть векторными и матричными. Вектор-строку задают в квадратных скобках как набор констант, разделенных между собой запятыми; например вектор дли-

ной 4: **[1,2,3,4]**. Матрица задается как набор векторных констант (строк), заключенных в квадратные скобки и разделенных между собой точкой с запятой; например, матрица размером 2×3: **[[1,2,3];[4,5,6]]**.

Именованные константы задаются с помощью ключевого слова **const**, после которого через запятую перечисляются имена констант с присвоенными им значениями, например: **const a=10.5, p=[1,2,3,4]**.

В программах SimInTech часто используются следующие встроенные числовые константы:

- **pi** – число π ;
- **expbase** – число e .

9. В SimInTech можно использовать *переменные* – это объекты, имеющие имена, предназначенные для хранения данных в памяти персонального компьютера (ПК). В зависимости от этих данных переменные могут быть числовыми или символьными, векторными или матричными.

Для присвоения значения переменным используется *операция присваивания*, которая имеет следующий формат:

<Имя переменной>=<выражение> {;}

Типы переменных определяются выражением, значение которого присваивается переменной. Выражение справа от знака присваивания может быть просто числом, вектором, матрицей, арифметическим выражением, строкой символов в кавычках или символьным выражением.

Формат задания типа и значения алгебраических переменных:

**var <имя переменной>{:<тип переменной>}
{=<инициирующее выражение>;}**

Рассмотрим наиболее часто используемые типы переменных.

Тип **integer** – целое 32-битное число. В программе, приведенной ниже, трем переменным присваивается значение 31 в десятичном, шестнадцатеричном (префикс **0x**) и двоичном (префикс **0b**) представлении:

```
var dec_x:integer=31, //в десятичном представлении
    hex_x:integer= 0x1F, //в шестнадцатеричном представлении
    binary_x:integer= 0b11111; //в двоичном представлении
```

Тип **double** – целое 64-разрядное число двойной точности стандарта IEEE-754 с плавающей запятой и знаком. В программе,

приведенной ниже, переменной `real_x` присваивается значение, равное -0,0001005 в формате с плавающей запятой:

```
1 | var real_x: double=-100.5e-6;
```

Тип **complex** – комплексное 128-битное число. Имеет формат вывода $a \pm bi$, где a – действительная часть, b – мнимая часть. Можно задать комплексное число также в виде (a, b) :

```
1 | var Z:complex=-3+4i,Z1:complex=(-3,4);
```

Тип **Array** – массив (вектор-строка) действительных чисел. В программе, приведенной ниже, переменной `p` размерностью 4 присваиваются значения элементам массива, равные 1, 2, 3, 4:

```
1 | var p:Array=[1,2,3,4];
```

Тип **boolean** – двоичное однобитное значение. В программе, приведенной ниже, переменной `bool_value` присваивается значение 0 или может присваиваться значение 1:

```
1 | var bool_value: boolean=TRUE; //или FALSE
```

Тип **string** – строка символов. В программе, приведенной ниже, переменной `str_value` присваивается значение строки символов «Лабораторная работа № 1»:

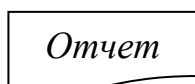
```
1 | var str_value: string="Лабораторная работа №1";
```

В программах SimInTech доступна системная переменная **time**, равная модельному времени.

Задание для отчета по лабораторной работе



Самостоятельно рассмотреть другие типы переменных, используемых в SimInTech.



Включить в отчет по лабораторной работе описание рассмотренных типов переменных и примеры их использования.

10. Вычисления математических выражений, содержащих операторы и функции, составляют главную цель любой системы, предназначенной для численных расчетов.

Оператор представляет собой специальное обозначение для определенной операции над данными (операндами).

Функции – это объекты с уникальными именами, которые выполняют определенные преобразования своих аргументов и при этом обязательно *возвращают* результаты на место вызова.

Если операции имеют равный приоритет, то они будут выполняться последовательно, слева направо, но сначала вычисляются выражения, заключенные в круглые скобки «()», после чего они рассматриваются как операнды. Операции с более высоким приоритетом будут выполняться в первую очередь, независимо от их расположения

В табл. 1 приведены обозначения арифметических, целочисленных и логических операторов SimInTech [1], их назначение и примеры применения.

Таблица 1

Обозначение	Назначение	Пример применения
1	2	3
+	Сложение выражения 1 (15) и выражения 2 (10)	$15 + 10 = 25$
	Поэлементное сложение двух векторов	$[3,4] + [2,2] = [5,6]$
	Поэлементное сложение двух матриц	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$
	Для строк сложение соответствует конкатенации	"Sim"+"InTech"="SimInTech"
–	Вычитание из выражения 1 (15) выражения 2 (10)	$15 - 10 = 5$
	Поэлементное вычитание двух векторов	$[3,4] - [2,2] = [1,2]$
	Поэлементное вычитание двух матриц	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} - \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} -4 & -4 \\ -4 & -4 \end{bmatrix}$
*	Умножение множителя 1 (15) на множитель 2 (10)	$15 * 10 = 150$
	Векторы умножаются поэлементно	$[3,4]*[2,2] = [6,8]$

1	2	3
	Перемножение матриц	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$
/	Деление делимого (15) на делитель (10)	15/10
	Поэлементное деление векторов	$[3,4] / [2,2] = [1.5, 2]$
	Для матриц производится умножение на обратную матрицу	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}^{-1} = \begin{bmatrix} 3 & -2 \\ 2 & -1 \end{bmatrix}$
.*	Поэлементное перемножение матриц	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .* \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 5 & 12 \\ 21 & 32 \end{bmatrix}$
./	Поэлементное деление матриц	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} ./ \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1/5 & 2/6 \\ 3/7 & 4/8 \end{bmatrix}$
^	Возведение первого выражения в степень второго. Вектора обрабатываются поэлементно	$5^2 = 25$
.^	Поэлементное возведение матрицы в степень	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .^ \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 64 \\ 2187 & 65536 \end{bmatrix}$
!	Вычисление факториала целого числа	$5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$
..	Вычисление интервала целых чисел от a до b.	1..5 обозначает [1,2,3,4,5]
or	Побитовая логическая операция ИЛИ	$x1 = \text{true} \text{ or } x2 = \text{true}$
and	Побитовая логическая операция И	$x1 = \text{true} \text{ and } x2 = \text{true}$
div	Целочисленное деление	$5 \text{ div } 2 \rightarrow (\text{возвращает}) 2$
mod	Остаток от целочисленного деления	$5 \text{ mod } 2 \rightarrow 1$
>	Операция БОЛЬШЕ. Операция возвращает единицу, если операнд 1 больше, чем операнд 2	$5 > 3 \rightarrow 1$ $5 > 10 \rightarrow 0$
<	Операция МЕНЬШЕ. Операция возвращает единицу, если операнд 1 меньше, чем операнд 2	$5 < 15 \rightarrow 1$ $15 < 5 \rightarrow 0$
>=	Операция БОЛЬШЕ ЛИБО РАВНО. Операция возвращает единицу если операнд 1 больше либо равен операнду 2	$5 \geq 5 \rightarrow 1$ $15 \geq 5 \rightarrow 1$ $3 \geq 5 \rightarrow 0$

1	2	3
<=	Операция МЕНЬШЕ ЛИБО РАВНО. Операция возвращает единицу, если операнд 1 меньше либо равен операнду 2	$5 \leq 15 \rightarrow 1$ $5 \leq 5 \rightarrow 1$ $5 \leq 3 \rightarrow 0$
=	Операция РАВНО. Операция возвращает единицу, если операнд 1 равен операнду 2	$3+2=5 \rightarrow 1$ $3+2=6 \rightarrow 0$
<>	Операция НЕ РАВНО. Операция возвращает единицу, если операнд 1 не равен операнду 2	$3+2 \neq 6 \rightarrow 1$ $3+2 \neq 5 \rightarrow 0$

Задание для отчета по лабораторной работе



Создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Записать в окне **Скрипт страницы** программу для расчета значения X в соответствии с заданным числовым выражением для своего номера варианта N :

$$X = \left\{ \frac{8,8077}{20 - [28,2 : (13,333 \cdot 0,3 + 0,0001)] \cdot 2,004} + 4,9 \right\} \cdot \frac{N}{32}.$$



Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы в окне **Скрипт страницы**, а также результаты ее выполнения (значение X) в окне **Просмотр значений переменных**.

11. В среде SimInTech можно использовать *функции* или *процедуры*, которые в зависимости от значений параметров *возвращают* различные результаты на место их вызова.

Формат задания функций (процедур):

“имя функции или процедуры” { (“параметр 1” { “параметр 2” }) }

В качестве разделителя параметров функции может использоваться запятая, точка с запятой или пробел. Функция может не иметь параметров. При обращении к функции в скобках указывается значение ее параметра (*аргумента*).

В табл. 2 приведены обозначения некоторых стандартных функций SimInTech [1], их назначение и примеры применения.

Таблица 2

Обозначение	Назначение	Пример применения
abs	Функция получения модуля вещественного или комплексного числа или вектора	abs(-5) → (возвращает) 5 abs(3+4i) → 5 abs([-3,-4]) → [3, 4]
exp	Функция вычисления экспоненты вещественного или комплексного числа	exp(2) → 7.389051
lg	Функция вычисления десятичного логарифма вещественного или комплексного числа	lg(100) → 2
ln	Функция вычисления натурального логарифма вещественного или комплексного числа	ln(5) → 1.6094379
max	Функция вычисления максимального значения из двух значений или из значений вектора	max(1,2) → 2 max([1,2,3,4,5]) → 5
min	Функция вычисления минимального значения из двух значений или из значений вектора	min(1,2) → 1 min([1,2,3,4,5]) → 1
root	Функция вычисления корня произвольной степени вещественного или комплексного числа	root(2,9) → 3, где 2 – степень корня; 9 – входное значение
sign	Функция определяет знак числа	sign(-50) → -1 (знак отрицательного числа) sign(50) → 1 (знак положительного числа)
round	Функция округляет действительное число до целого	round(2.4) → 2
sqrt	Функция вычисления корня квадратного вещественного или комплексного числа	sqrt(25) → 5
trunc	Функция отделяет целую часть действительного числа от дробной	trunc(13.7) → 13
rand	Функция генерации равномерного шума в диапазоне значений от 0 до 1	var t:array = [0,0]; t[1]=rand; t[2]= rand; Значения элементов массива t будут заполняться случайными числами

В табл. 3 приведены обозначения тригонометрических функций SimInTech [1], их назначение и примеры применения.

Обратите внимание, что аргументы тригонометрических функций и возвращаемые обратными тригонометрическими функциями углы в SimInTech задаются в радианах.

Таблица 3

Обозначение	Назначение	Пример применения
arccos	Функция вычисления арккосинуса вещественного или комплексного числа. Значение вещественного аргумент x должно быть ограничено условием $-1 \leq x \leq 1$	$\text{arccos}(0.5) \rightarrow 1.0471976$
arcsin	Функция вычисления арксинуса вещественного или комплексного числа. Значение вещественного аргумент x должно быть ограничено условием $-1 \leq x \leq 1$	$\text{arcsin}(0.5) \rightarrow 0.52359878$
arctg	Функция вычисления арктангенса вещественного или комплексного числа	$\text{arctg}(1) \rightarrow 0.78539816$
atan2	Функция вычисления арктангенса точки с учетом квадранта. Формат: $\text{atan2}(y, x)$, где y – координата по оси ординат (действительное число или массив действительных чисел); x – координата по оси абсцисс (действительное число или массив действительных чисел)	$\text{atan2}(1,1) \rightarrow 0.78539816$ $\text{atan2}(1,-1) \rightarrow 2.3561945$ $\text{atan2}(-1,-1) \rightarrow -2.3561945$ $\text{atan2}(-1,1) \rightarrow -0.78539816$
sin	Функция вычисления синуса вещественного или комплексного числа	$\sin(\pi/6) \rightarrow 0.5$
cos	Функция вычисления косинуса вещественного или комплексного числа	$\cos(\pi/3) \rightarrow 0.5$
tg	Функция вычисления тангенса вещественного или комплексного числа	$\text{tg}(\pi/4) \rightarrow 1$
ctg	Функция вычисления котангенса вещественного или комплексного числа	$\text{ctg}(\pi/4) \rightarrow 1$

В табл. 4 приведены обозначения функций SimInTech для работы с комплексными числами [1] на примере комплексного числа $\underline{Z} = 3 + i4$.

Таблица 4

Обозначение	Назначение	Пример применения
arg	Функция получения аргумента комплексного числа z , которая возвращает угол (в радианах), образованный на комплексной плоскости вектором, определяемым комплексным числом, с вещественной осью	$\text{arg}(3+4i) \rightarrow 0.92729522$
complex	Функция получения комплексного числа	$\text{complex}(3,4) \rightarrow 3+4i$
conj	Функция получения комплексно-сопряженного числа	$\text{conj}(z) \rightarrow 3 - 4i$
imag	Функция получения мнимой части комплексного числа	$\text{imag}(z) \rightarrow 4$
real	Функция получения вещественной части комплексного числа	$\text{real}(z) \rightarrow 3$
sgr	Функция получения комплексного квадратного корня вещественного числа	$\text{sgr}(25) \rightarrow 5+0i$

Задание для отчета по лабораторной работе



Отчет

Написать в SimInTech программу, задающую комплексное число $\underline{Z} = N + i \cdot (N+5)$ для своего номера варианта N . Используя функции SimInTech вычислить действительную и мнимую части заданного комплексного числа, а также его модуль и аргумент.

Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы в окне **Скрипт страницы**, а также результаты ее выполнения в окне **Просмотр значений переменных**.



Даны $x = 1,5$; $y = 2$; $z = 3$. Вычислить a и b из табл. 5 для своего варианта задания в формате **double**.

Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы в окне **Скрипт страницы**, а также результаты ее выполнения (значения a и b) в окне **Просмотр значений переменных**.

Таблица 5

Номер варианта	a	b
1	2	3
1	$a = \frac{z + y / (x^2 + 4)}{e^{-x} / x^2 + 4}$	$b = x + y / (z + 1/6)$
2	$a = \frac{3,5 + e^{y-1}}{1 + x^2 y - \operatorname{tg} z }$	$b = \frac{(y-x)^2}{1-2y} + \frac{ y-x ^3}{3}$
3	$a = \frac{\sqrt{ x-1 } - \sqrt[3]{ z }}{1 + \frac{x^2}{2,5} + \frac{y^2}{4}}$	$b = \frac{1 + \cos \frac{y}{2}}{\frac{x^2}{2} + \sin^2 z}$
4	$a = \frac{z + x}{y^2 + y + x^3 / 1,3 }$	$b = 1 + \operatorname{tg} \frac{z}{2x + 2y}$
5	$a = \frac{\cos z + 1}{\sqrt{(1 + x \cdot y)}}$	$b = x^y + \frac{z^2}{3 + z^2 / 5}$
6	$a = \frac{1,5 + \sin^2 z}{ x - 2x / (1 + x^2 y^2) }$	$b = \cos^2 \frac{y}{z} + \sin^2 x$
7	$a = \left \frac{1,5y}{z + x^2 / 4} \right $	$b = x - \frac{x^2}{3!} + \frac{x^5}{5!}$
8	$a = \frac{\sin(x^2 - 2y + z)}{2,6 x^y + 2}$	$b = \cos^2 \left(x^2 + \frac{y}{z} \right)$
9	$a = \frac{\cos(x - \pi / 2)}{0,5 + \sin^2 x}$	$b = \frac{5y^2}{3z^2 + 7}$
10	$a = \frac{3,5 + \operatorname{tg}(x^2 + y)}{ x - 4x / (1 + xy^2) }$	$b = \sin^2 \left(\operatorname{arctg} \frac{1}{z + 3} \right)$

1	2	3
11	$a = \frac{2,6 + \operatorname{tg}(x - y)}{\left x - 2x / (x^2 + y^2) \right }$	$b = \operatorname{tg}^2 \frac{z + 2y}{2x}$
12	$a = \ln \left \frac{y + x^2 / 4}{5z} \right $	$b = 5,5 + \frac{x}{y(\operatorname{tg} z + 1)}$
13	$a = \frac{z + y / (x^2 + 4)}{e^{-x-2} / (x^2 + 4)}$	$b = \frac{(y - x)^2}{2} + \frac{ y - x ^3}{3}$
14	$a = \frac{3,5 + e^{y-1}}{1 + x^2 y - \operatorname{tg} z }$	$b = x / y(\operatorname{tg} z + 1/6)$
15	$a = \frac{3 + \sin^3(x^2 + y)}{2 + \left x - x / (1 + x^2 y^2) \right }$	$b = 1 + \frac{z^2}{3 + z^2 / 5}$
16	$a = \frac{\sqrt{ x - 1 } - \sqrt[3]{ y }}{1,5 + x^2 + y^2}$	$b = \cos^2 \left(\operatorname{arctg} \frac{1}{z + 1} \right)$
17	$a = \frac{\cos(x - \pi / 2)}{0,5 + \sin^2 x}$	$b = x + \frac{x^2}{3 + z} + \frac{x^2}{5 + z}$
18	$a = \frac{3,3 + y^2 + (x^2 + 2)}{e^{-0,5} + 1 / (x^2 + 4)}$	$b = \cos^2 \left(x^2 + \frac{y}{1 + z} \right)$
19	$a = y + \frac{3,5x}{y^2 - \sqrt{\frac{x^2}{2y + x^2}}}$	$b = 2 + \frac{y^2}{3 + \frac{z^2}{1 + x}} - 3x$
20	$a = \frac{2 \cos(z - 1)}{1 / 2 + \sin^2 z}$	$b = \sqrt{\sin^3 \frac{1}{x + z}}$
21	$a = \frac{z + y / (x^2 + 4)}{e^{-x} / x^2 + 4}$	$b = \sin^2 \left(\operatorname{arctg} \frac{1}{z + 3} \right)$
22	$a = \frac{2,6 + \operatorname{tg}(x - y)}{\left x - 2x / (x^2 + y^2) \right }$	$b = 5,5 + \frac{x}{y(\operatorname{tg} z + 1)}$
23	$a = \frac{\sqrt{ x - 1 } - \sqrt[3]{ z }}{1 + \frac{x^2}{2,5} + \frac{y^2}{4}}$	$b = 5,5 + \frac{x}{y(\operatorname{tg} z + 1)}$

1	2	3
24	$a = \frac{z + y / (x^2 + 4)}{e^{-x-2} / (x^2 + 4)}$	$b = x / y (\operatorname{tg} z + 1/6)$
25	$a = \frac{\cos z + 2}{\sqrt{(1 + x \cdot y)}}$	$b = \frac{(y - x)^2}{2} + \frac{ y - x ^3}{3}$
26	$a = \frac{\sqrt{ x-1 } - \sqrt[3]{ y }}{1,5 + x^2 + y^2}$	$b = \cos^2 \left(\operatorname{arctg} \frac{1}{z+1} \right)$
27	$a = \left \frac{1,5y}{z + x^2 / 4} \right $	$b = 1 + \operatorname{tg} \frac{z}{2x + 2y}$
28	$a = \frac{\cos(x - \pi/2)}{0,5 + \sin^2 x}$	$b = x + \frac{x^2}{3 + z} + \frac{x^2}{5 + z}$
29	$a = \frac{\cos(x - \pi/2)}{0,5 + \sin^2 x}$	$b = \cos^2 \frac{y}{z} + \sin^2 x$
30	$a = \frac{2 \cos(z-1)}{1/2 + \sin^2 z}$	$b = \frac{(y-x)^2}{2} + \frac{ y-x ^3}{3}$
31	$a = \frac{z + y / (x^2 + 4)}{e^{-x} / x^2 + 4}$	$b = \frac{(y+x)^2}{2} + \frac{ y+x ^3}{3}$
32	$a = \frac{\sqrt{ x+1 } - \sqrt[3]{ z }}{1 + \frac{x^2}{5} + \frac{y^2}{4}}$	$b = x + y / (z + 1/6)$
33	$a = \frac{3,5 + e^{y+1}}{5 + x^2 y + \operatorname{tg} z }$	$b = \frac{1 + \cos \frac{y}{2}}{1 + \sin^2 z}$
34	$a = \frac{z + 3x}{y^2 + y + x^3 / 1,3 }$	$b = 1 + \operatorname{tg} \frac{z}{x + y}$
35	$a = \frac{\cos(z + 3)}{\sqrt{(1 + x / y)}}$	$b = x^2 + \frac{z^2}{3 + z^2}$
36	$a = \frac{1,5 - \sin^2 z}{ x - 2x / (1 + y^2) }$	$b = \cos^2 z + \sin^2 x$
37	$a = \frac{\sin(x^2 + y)}{ x - 4x / (xz^2) }$	$b = \sin^2 \left(\frac{x+1}{z+3} \right)$

1	2	3
38	$a = \left \frac{x + 2y}{z + x^2 / 4} \right $	$b = x - \frac{x^2}{3!} + \frac{x^3}{4!}$
39	$a = \frac{\sin(x^2 - z)}{6y + 2}$	$b = \cos^2\left(x + \frac{y}{z}\right)$
40	$a = \frac{\cos(x + \pi / 3)}{5 - \sin^2 x}$	$b = \frac{5y^2}{3z - x}$

Контрольные вопросы

1. Каким образом можно написать скрипт в системе SimInTech?
2. Какова структура программ на языке SimInTech?
3. Каким образом можно вводить комментарии?
4. По каким правилам задаются идентификаторы?
5. Какие правила должны соблюдаться при написании деклараций на языке SimInTech?
6. Как изменяются свойства блока **Язык программирования**?
7. Как запустить написанную программу на выполнение?
8. Куда выводятся результаты выполнения программы?
9. Каков формат задания числовых констант на языке SimInTech?
10. Каков формат задания алгебраических переменных на языке SimInTech?
11. Какие форматы задания алгебраических переменных на языке SimInTech вы знаете?
12. Какие арифметические операторы используются в языке SimInTech?
13. Каков формат задания функции в языке SimInTech?
14. Какие стандартные функции используются в языке SimInTech?
15. Какие функции используются для работы с комплексными числами в языке SimInTech?

Лабораторная работа № 2

РАЗВЕТВЛЯЮЩИЕСЯ И ЦИКЛИЧЕСКИЕ ПРОГРАММЫ В SimInTech

Цель работы: знакомство с операторами, предназначенными для составления разветвляющихся и циклических программ в системе SimInTech.

Рабочее задание

1. Запустить программу SimInTech, создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования**, открыть его диалоговое окно и записать в нем программу, которая ищет максимальное значение из значений двух переменных:

```
output a1=11, a2=2;      //инициализированы 2 выходные
                        //переменные вещественного типа
if a1>a2 then am=a1 else am=a2;      //сохранение
//в переменной am максимального значения из a1 и a2
output am      //выходная переменная вещественного типа
```

Сохранить проект в своей папке в виде файла под именем **лр_2_1** с расширением **prt** и запустить программу на выполнение, нажав на кнопку **Пуск**  на панели инструментов окна проектов.

Добавить линии связи от появившихся выходных портов блока **Язык программирования**. Снова запустить программу на выполнение.

На рис. 1 показаны результаты работы этой программы. Значения двух переменных **a1**, **a2** и полученное максимальное значение **am** выводятся в выходные порты блока **Язык программирования**.

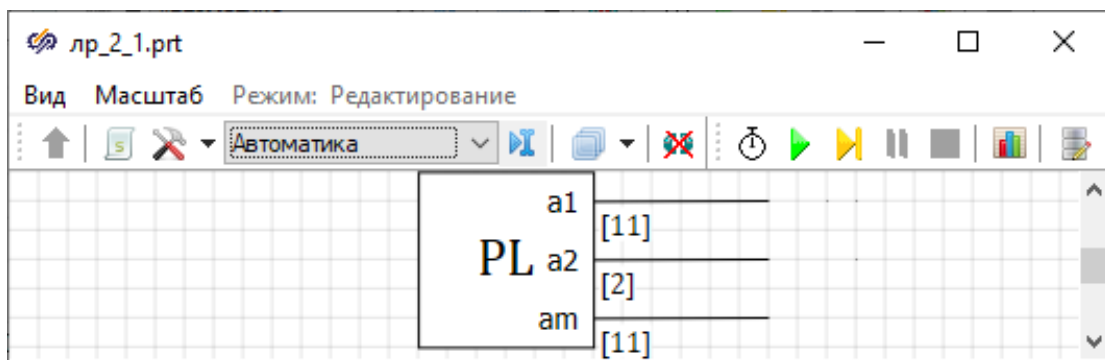


Рис. 1

В этой программе использован оператор условного перехода **if**, который используется для организации разветвляющихся алгоритмов и имеет следующий формат:

```
if <условие или логическое выражение>
then <оператор при выполнении условия>
{else <оператор при невыполнении условия>;}
```

Если условие верно, то будет выполнен оператор после ключевого слова **then**, после его выполнения программа переходит на следующую командную строку. Если же условие ложно, то будет выполнен оператор после ключевого слова **else** и переход на следующую командную строку. При этом можно задавать и сложные условия.

Отредактировать текст программы, задав значение $a_2=20$ и проверить правильность ее работы. Заккрыть окно проекта.

2. Создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования** и записать в его диалоговом окне программу, в которой условие после ключевого слова **if** задано в виде логического выражения и в каждой ветви используются операторные скобки:

```
var x1:boolean=1, x2:boolean=1;
// инициализированные переменные двоичного типа
var y, u; // переменные вещественного типа
if x1=true and x2=true then// условие в виде логического И
begin
  y=2; u=5;
end
else
begin
  y=0; u=0;
end;
output y,u;
```

После ключевых слов возможно использовать операторные скобки **begin ... end**, которые могут быть вложенными, объединяющими несколько операторов в один:

```
begin
  <операция 1>;
  ....
  <операция n>;
end;
```

Сохранить проект в своей папке в виде файла под именем **лр_2_2** и запустить программу на выполнение.

Добавить линии связи от появившихся выходных портов блока **Язык программирования**. Снова запустить программу на выполнение.

На рис. 2 показаны результаты работы этой программы при $x_1 = 1$ и $x_2 = 1$. Так как условие после оператора **if** истинно, то выполняются операторы между операторными скобками после ключевого слова **then**.

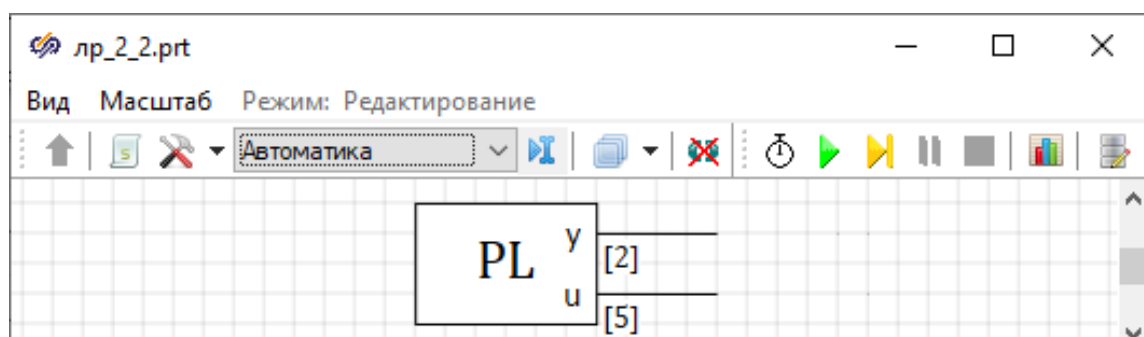


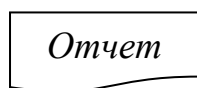
Рис. 2

Отредактировать текст программы, задав значение хотя бы одной из переменных x_1 или x_2 равным 0 и проверить правильность ее работы. Закрыть окно проекта.

Задание для отчета по лабораторной работе



Написать в SimInTech программу, вычисляющую значение ступенчатой функции Z из табл. 1 в формате **double** для своего варианта задания.



Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы, а также результаты ее выполнения в схемном окне проекта для всех ветвей.



Дано три числа. Упорядочить их в порядке возрастания для четных вариантов или в порядке убывания для нечетных вариантов.



Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы, а также результаты ее выполнения в схемном окне проекта.

Таблица 1

Номер варианта	Функция
1	2
1, 11, 21, 31	$Z = \begin{cases} 10,5x^2 + 1, & x \leq 0; \\ 5(x+1)^3, & 0 < x < 1; \\ \sqrt{x^2 + 1}, & x \geq 1 \end{cases}$
2, 12, 22, 32	$Z = \begin{cases} \cos(x-1), & -2\pi < x \leq 0; \\ \cos x + 1, & 0 < x < \pi; \\ \sin(x+0,5), & \pi \leq x \leq 2\pi \end{cases}$
3, 13, 23, 33	$Z = \begin{cases} x^2 + 5x, & x < 1; \\ (x^2 + 3x)^2, & x = 1; \\ (x - 2x^2), & x > 1 \end{cases}$
4, 14, 24, 34	$Z = \begin{cases} \sin 3x, & -\pi < x \leq 0; \\ \cos(x+1), & 0 < x < \pi/2; \\ \sin 2x, & \pi/2 \leq x \leq \pi \end{cases}$
5, 15, 25, 35	$Z = \begin{cases} 2 \sin(x), & x \leq 0 \\ -\sin(2x), & 0 < x \leq \pi \\ 3 \sin(x), & \pi < x \leq 2\pi \end{cases}$
6, 16, 26, 36	$Z = \begin{cases} \sin(x), & x \leq 0 \\ -\sin(x), & 0 < x \leq \pi \\ \sin(x), & \pi < x \leq 2\pi \end{cases}$
7, 17, 27, 37	$Z = \begin{cases} x^4 - 12, & x \leq -1; \\ (2x+1)^3, & -1 < x < 1; \\ \sqrt{\frac{x+11}{x}}, & x \geq 1 \end{cases}$
8, 18, 28, 38	$Z = \begin{cases} 15x+1, & x \leq 0; \\ 5(x+1)^2, & 0 < x < 1; \\ \sqrt{x^3-1}, & x \geq 1 \end{cases}$

появившихся выходных портов блока **Язык программирования**. Снова запустить программу на выполнение.

На рис. 3 показаны результаты работы этой программы.

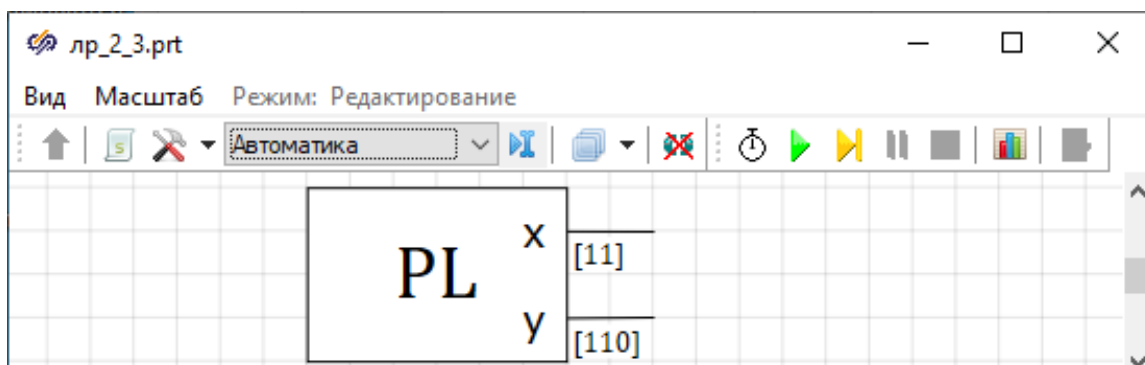


Рис. 3

4. В SimInTech для организации цикла с предусловием используется оператор **while**:

while <условие цикла> **do** <оператор цикла>;

Оператор цикла выполняется, пока выполняется условие цикла.

Создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования** и записать в его диалоговом окне программу с использованием оператора **while**, в которой вычисляется значение функции $y = \sum_{x=1}^{10} (x * 2)$ для x , меняются от 1 до 10 с шагом 1:

```
1  var y;
   y=0; x=1;
   while x<=10 do // условие цикла
   begin
       y=y+x*2; // операторы цикла
       x=x+1; // изменение переменной цикла
   end;
   output x,y;
```

Сохранить проект в своей папке в виде файла под именем **лр_2_4** и запустить программу на выполнение.

Добавить линии связи от появившихся выходных портов блока **Язык программирования**. Снова запустить программу на выполнение. Убедиться в том, что результаты работы этой программы совпадают с результатами работы предыдущей программы из п. 3.

5. В SimInTech для организации цикла с постусловием используется оператор **repeat**:

repeat

<оператор цикла>;

until <условие выхода>;

Оператор цикла выполняется, пока выполняется условие выхода из цикла.

Создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования** и записать в его диалоговом окне программу с использованием оператора **repeat**, в которой вычисляется значение функции $y = \sum_{x=1}^{10} (x * 2)$ для x , меняются от 1 до 10 с шагом 1:

```
1  var y;  
    y=0; x=1;  
    repeat  
    begin  
        y=y+x*2; // операторы цикла  
        x=x+1; // изменение переменной цикла  
    end;  
until x>10; // условие выхода  
output x,y;
```

Сохранить проект в своей папке в виде файла под именем **лр_2_5** и запустить программу на выполнение.

Добавить линии связи от появившихся выходных портов блока **Язык программирования**.

Снова запустить программу на выполнение. Убедиться в том, что результаты работы этой программы совпадают с результатами работы предыдущей программы из п. 3–4.

6. Циклы могут быть вложенными, когда внутри тела цикла необходимо повторять некоторую последовательность операторов.

Вложенные циклы не могут пересекаться: пока не закончился внутренний цикл, внешний не может завершиться.

С целью экономии машинного времени все операторы, которые не зависят от параметров внутреннего цикла, должны выноситься во внешний цикл.

Рассмотрим пример программы с использованием оператора **for**, в которой вычисляется значение функции $a = \sum_{n=1}^3 \left(\sqrt{10 + x^n} \right)$ при следующих значениях: $x = 1, 2, 3$.

Создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования** и записать в его диалоговом окне программу:

```

1  output a[3],x[3];
-  x=[1,2,3];
-  for (i=1,3,1) //условие цикла
-  begin
-      s=0;
-      for (n=1,3,1) //условие цикла
-      begin
-          a[i]=s+sqrt(10+x[i]^n);
-          s=a[i];
-      end;
10 end;

```

Сохранить проект в своей папке в виде файла под именем **лр_2_6** и запустить программу на выполнение.

Добавить линию связи от появившихся выходных портов блока **Язык программирования**. Снова запустить программу на выполнение.

На рис. 4 показаны результаты работы этой программы в виде значений векторов на соответствующих портах блока **Язык программирования**.

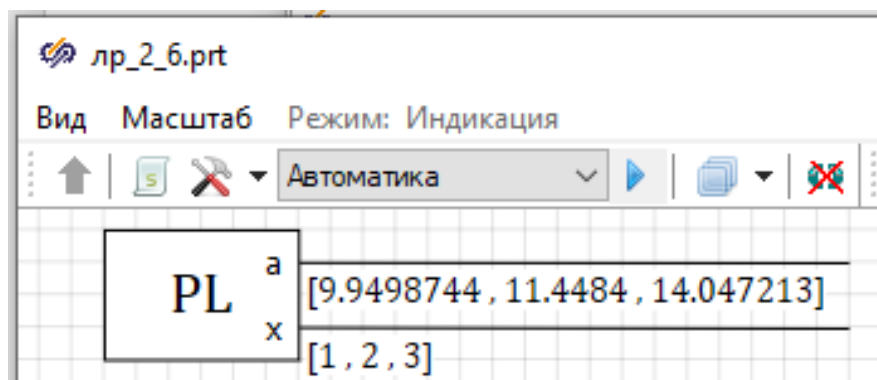


Рис. 4

7. Создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования** и записать в его диалоговом окне программу с использованием оператора **for**, в которой вычисляется значение функции

$$a = \prod_{n=1}^3 \sqrt{10 + x^n} \quad \text{при следующих значениях: } x = 1, 2, 3:$$

```

1  output a[3],x[3];
-  x=[1,2,3];
-  for (i=1,3,1) //условие  цикла
-  begin
-  |   s=1;
-  |   for (n=1,3,1) //условие  цикла
-  |   |   begin
-  |   |   |   a[i]=s*sqrt(10+x[i]^n);
-  |   |   |   s=a[i];
-  |   |   |   end;
10 |   end;
-  end;

```

Сохранить проект в своей папке в виде файла под именем **лр_2_7** и запустить программу на выполнение.

Добавить линию связи от появившихся выходных портов блока **Язык программирования**. Снова запустить программу на выполнение.

На рис. 5 показаны результаты работы этой программы в виде значений векторов на соответствующих портах блока **Язык программирования**.

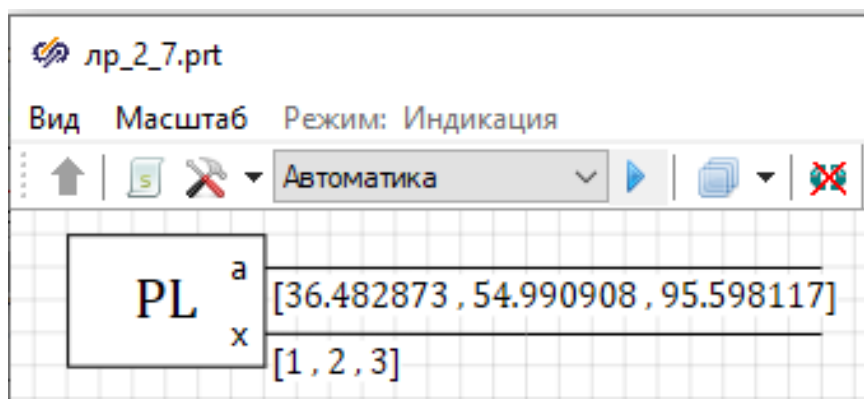


Рис. 5

Задание для отчета по лабораторной работе



Отчет

Составьте три варианта циклической программы: с операторами **for**, **while** и **repeat** для вычисления значения функции $A(x)$ в формате **double** из табл. 2 для своего варианта задания N при x , равном $N-1$, N , $N+1$.

Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы, а также результаты ее выполнения в схемном окне проекта, где N – номер варианта.

Таблица 2

Номер варианта	Функция	Номер варианта	Функция
1	2	3	4
1	$A = \sum_{n=1}^5 \sqrt[4]{10n + x^4}$	17	$A = \sum_{n=1}^5 \sqrt{n^2 + 2x}$
2	$A = \prod_{n=1}^8 (x^2 - n)$	18	$A = \prod_{k=1}^5 \sqrt{(xk + 1)^2}$
3	$A = \prod_{k=1}^6 \frac{\cos x}{2k}$	19	$A = \sum_{k=0}^5 \operatorname{tg}(x + k)$
4	$A = \sum_{k=1}^4 (x^4 + 4k)$	20	$A = \prod_{k=1}^3 (2^x + kx)$
5	$A = \prod_{k=1}^5 (e^x + kx)$	21	$A = \prod_{k=1}^4 \sin^x(3k)$
6	$A = \sum_{k=1}^{10} \frac{1-x}{k}$	22	$A = \sum_{n=1}^5 \sqrt{n^2 + 2x}$
7	$A = \sum_{i=1}^8 \frac{1+x}{i^3}$	23	$A = \prod_{k=1}^8 \sqrt{(xk + 2)^3}$
8	$A = \prod_{k=1}^6 \frac{\sin x}{2k + 1}$	24	$A = \sum_{n=1}^6 \frac{x^n}{n}$
9	$A = \sum_{i=1}^7 \frac{x}{(2i + 1)^2}$	25	$A = \sum_{k=0}^5 \operatorname{tg}(x + k)$
10	$A = \sum_{i=1}^8 \frac{(-1)^i}{(x + 1)^5}$	26	$A = \sum_{i=1}^{10} \frac{1+x}{i + 1}$
11	$A = \prod_{n=1}^6 (2x^2 - n)$	27	$A = \prod_{k=1}^5 \sqrt{(xk + 1)^2}$
12	$A = \sum_{n=1}^6 \frac{x^3}{n}$	28	$A = \sum_{i=1}^8 \frac{(x)}{i(i + 1)}$
13	$A = \sum_{k=0}^3 (x + k)^2$	29	$A = \sum_{i=1}^6 \frac{(x + 2)}{i(i - 1)}$
14	$A = \prod_{k=1}^4 \sqrt{(xk + 1)}$	30	$A = \sum_{i=1}^5 \frac{(x)^2}{4^i + 5^i}$
15	$A = \sum_{i=1}^6 \frac{(-1)^x}{x(i - 3)}$	31	$A = \sum_{k=1}^4 (x^2 + 2k)$
16	$A = \prod_{k=1}^8 \sqrt{(xk + 2)^3}$	32	$A = \sum_{n=1}^5 \sqrt[4]{n^2 + x^2}$

1	2	3	4
33	$A = \sum_{i=1}^8 \frac{(i+1)}{(2x+1)}$	37	$A = \prod_{k=1}^6 \frac{\cos x}{2k}$
34	$A = \prod_{k=1}^3 (2^x + kx)$	38	$A = \sum_{k=1}^{10} \frac{1-x}{k}$
35	$A = \sum_{n=1}^4 \sqrt{n^4 + x}$	39	$A = \prod_{k=1}^6 \frac{\sin x}{2k+1}$
36	$A = \sum_{n=1}^5 \sqrt[4]{2n+x^3}$	40	$A = \sum_{i=1}^8 \frac{(-1)^x}{(x+1)(i+2)}$

Контрольные вопросы

1. Каковы назначение и формат оператора **if** в языке программирования SimInTech?
2. Какова последовательность выполнения команд после выполнения оператора, стоящего после ключевого слова **then**?
3. Примеры условий и логических выражений, которые могут стоять после ключевого слова **if**.
4. Для чего используются операторные скобки и какой у них формат?
5. Какие операторы используются в SimInTech для организации циклов и какой их формат?
6. Для чего и как организуются и вложенные циклы в SimInTech?

Лабораторная работа № 3

РАБОТА С МАССИВАМИ В СРЕДЕ SimInTech

Цель работы: знакомство с алгоритмами и функциями по обработке векторов и матриц, а также с основными операциями над ними.

В математике массивом называется упорядоченный набор элементов, каждый из которых хранит одно значение, идентифицируемое с помощью одного или нескольких индексов. Под двумерным массивом X_{ij} или матрицей понимают прямоугольный набор чисел, который характеризуется двумя целыми числами: числом строк ($i \geq 1$) и столбцов ($j \geq 1$). Частными случаями матриц являются: вектор-строка (состоит из одной строки) и вектор-столбец (состоит из одного столбца). В SimInTech оператор [], используемый для упаковки элементов в массив, имеет следующий формат:

"массив или матрица"["номер элемента или строки"
{"номер столбца"}]

Если элементы – действительные или целые числа, то результат – вектор действительных чисел. Если элементы – векторы действительных чисел, результат – матрица действительных чисел [1].

Рабочее задание

1. Запустить программу SimInTech, создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования**, открыть его диалоговое окно и записать в нем программу:

```
output v1[2], // массив v1 размером 2
      m[2,3], // матрица m 2 строки на 3 столбца
      v2[2,1], // вектор-столбец v2 из двух строк
      v3[1,2]; // вектор-строка v3 из двух столбцов
v1=[1,2];      m=[[1,2,3];[4,5,6]];
v2=[[1];[2]];  v3=[[1,2]];
```

В программе иницируются выходные переменные блока в виде матриц, элементам строк которых затем присваиваются значения путем указания их внутри квадратных скобок через запятую. Для матрицы и вектора-столбца каждая строка располагается внутри

внешних квадратных скобок и отделяется от соседней строки точкой с запятой.

Сохранить проект в своей папке в виде файла под именем **лр_3_1** и запустить программу на выполнение. Добавить линии связи от появившихся выходных портов блока **Язык программирования**. Снова запустить программу на выполнение.

На рис. 1 показаны результаты работы этой программы в окне **Просмотр значений переменных**, из которого видно, что в памяти компьютера хранятся локальные переменные, значения элементов которых в математике записываются следующим образом:

- одномерный массив $v1 = [1 \ 2]$;
- двумерный массив (матрица) $m = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$;
- вектор-столбец $v2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$;
- вектор-строка $v3 = [1 \ 2]$.

Просмотр значений переменных			
Имя переменной	Тип	Значение	Флаги
Локальные переменные			
v1	Массив	[1 , 2]	output;
m	Матрица	[[1 , 2 , 3];[4 , 5 , 6]]	output;
v2	Матрица	[[1];[2]]	output;
v3	Матрица	[[1 , 2]]	output;
Основная секция кода			
0 : v1=[1, 2]			
1 : m=[[1, 2, 3], [4, 5, 6]]			
2 : v2=[[1], [2]]			
3 : v3=[[1, 2]]			
Секция инициализации			
0 : v1=vector(2)			
1 : m=matrix(2, 3)			
2 : v2=matrix(2, 1)			
3 : v3=matrix(1, 2)			

Рис. 1

Обратите внимание, что в SimInTech элементы матрицы выводятся в квадратных скобках в одну строку: сначала элементы первой строки в квадратных скобках, затем – после точки с запятой – элементы второй строки в квадратных скобках и т.д.

На рис. 2 показаны значения элементов массива и матриц, которые выводятся в окне проекта в соответствующие выходные порты блока **Язык программирования**.

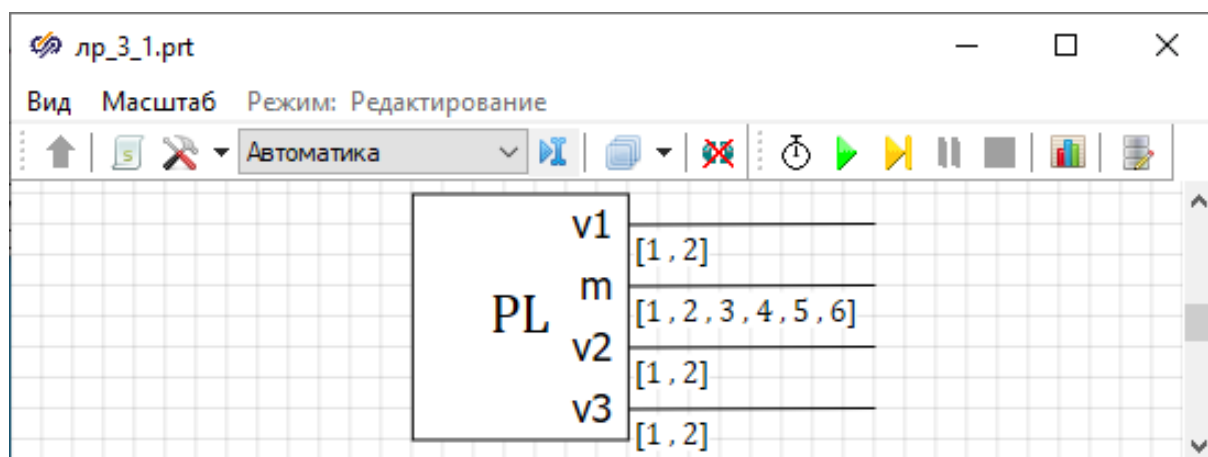


Рис. 2

Элементы матрицы выводятся в выходные порты в квадратных скобках в одну строку: сначала элементы первой строки, затем – после запятой – элементы второй строки и т.д.

2. Создать новый проект в SimInTech на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования**, открыть его диалоговое окно и записать в нем программу:

```

1  var a[2,2] ;           //матрица a размером 2x2
   output a1[2],a2[2]; //массивы a1 и a2 размером 2
   for (i=1,2,1)         //условие внешнего цикла
   begin
       for (j=1,2,1)     //условие внутреннего цикла
       a[i,j]=trunc(100*rand)-50;
       end;
   a1=a[1]; //1 строка матрицы a заносится в массив a1
   a2=a[2]; //2 строка матрицы a заносится в массив a2

```

В программе формируется матрица a размером 2×2, а также два одномерных массива a1 и a2 длиной 2 в качестве выходных переменных

блока **Язык программирования**. Затем с помощью двух вложенных циклов элементам матрицы *a* присваиваются значения, формируемые генератором случайных чисел (функция **rand**). Значения элементов первой строки матрицы *a* заносятся в массив *a1*, а второй строки – в массив *a2*.

Функция **rand** возвращает случайное число действительного типа в диапазоне от 0 до 1. Для получения случайного числа в диапазоне от 0 до 100 умножим **rand** на 100 и округлим число с помощью функции **trunc**. Если вычесть из полученного числа 50, то получается целое случайное число в диапазоне от –50 до 50, которое присваивается элементу матрицы *a*(*i*,*j*).

Обратите внимание, что при обращении к элементу матрицы в SimInTech используется оператор **[]**, имеющий следующий формат:

"массив или матрица"["номер элемента или строки"
{"номер столбца"}]

Оператор **[]** возвращает значение элемента с указанным номером. Нумерация начинается с 1. Тип возвращаемого значения соответствует типу элементов массива. Если для матрицы указывается один индекс, то возвращается вектор-строка матрицы, если два – возвращается значение соответствующего элемента матрицы.

Возвращаемое оператором значение может быть присвоено другой переменной. Разделителем элементов может быть запятая, точка с запятой или пробел.

Сохранить проект в своей папке в виде файла под именем **лр_3_2** и запустить программу на выполнение.

Добавить линии связи от появившихся выходных портов блока **Язык программирования**. Снова запустить программу на выполнение.

На рис. 3 показаны значения локальных переменных программы в окне **Просмотр значений переменных**.

Имя переменной	Тип	Значение	Флаги
Локальные переменные			
a	Матрица	[[-12 , -28];[-43 , 12]]	timed;
a1	Массив	[-12 , -28]	timed;output;
a2	Массив	[-43 , 12]	timed;output;
i	Вещественное	3	

Рис. 3

На рис. 4 значения элементов массивов a1 и a2 выводятся в окне проекта в соответствующие выходные порты блока **Язык программирования**.

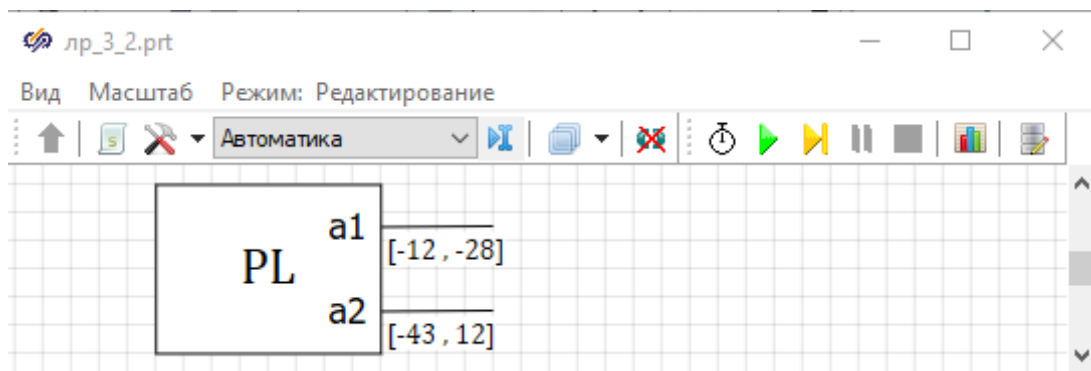


Рис. 4

Добавить два примитива **AutoLabel** рядом с выходными портами блока **Язык программирования**, выделив ЛКМ верхнюю линию связи и открыть для нее окно **Свойства**. Выделить на закладке **Параметры** строку с указанием имени сигнала (на рис. 5 – **y**) и нажать кнопку **A**.

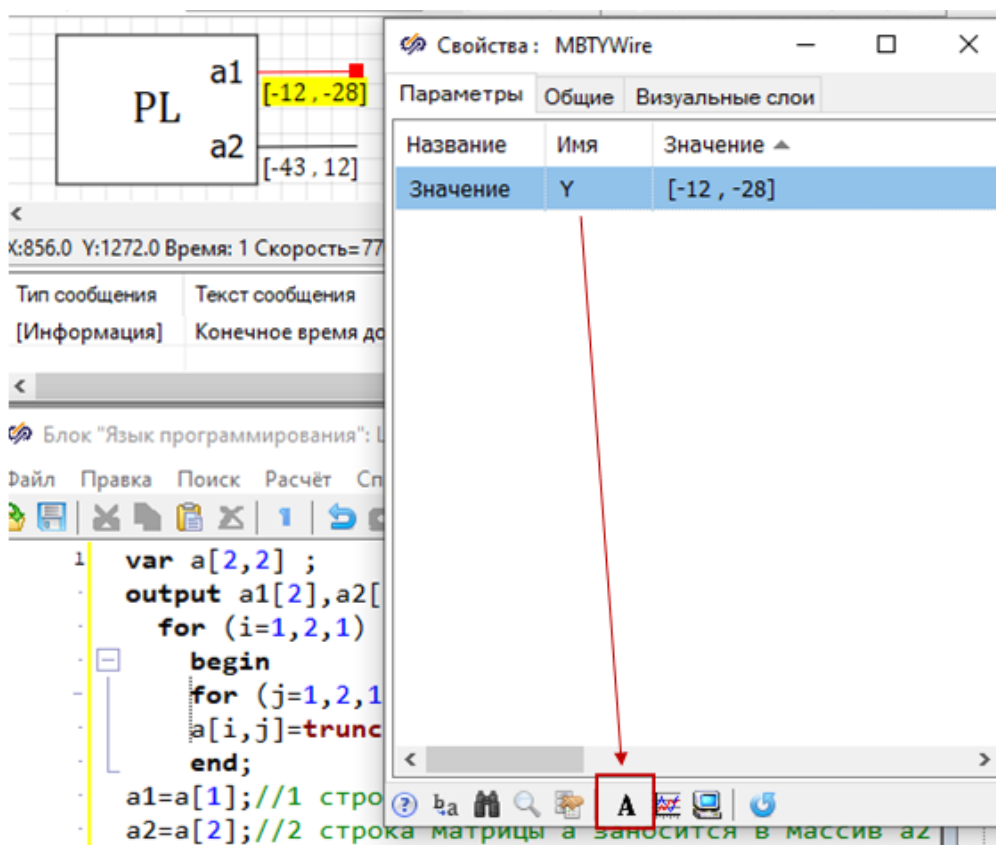


Рис. 5

В открывшемся окне **Подпись объектов** (рис. 6) указать на панели **Стиль** требуемый **Текст подписи**, который будет отображаться перед выводимым числовым значением выводимого параметра.

В поле **Выводимые параметры** будет указано имя параметра блока, значение которого будет выведено в числовом виде. Поле **Имя класса подписи** содержит уникальный класс блока.

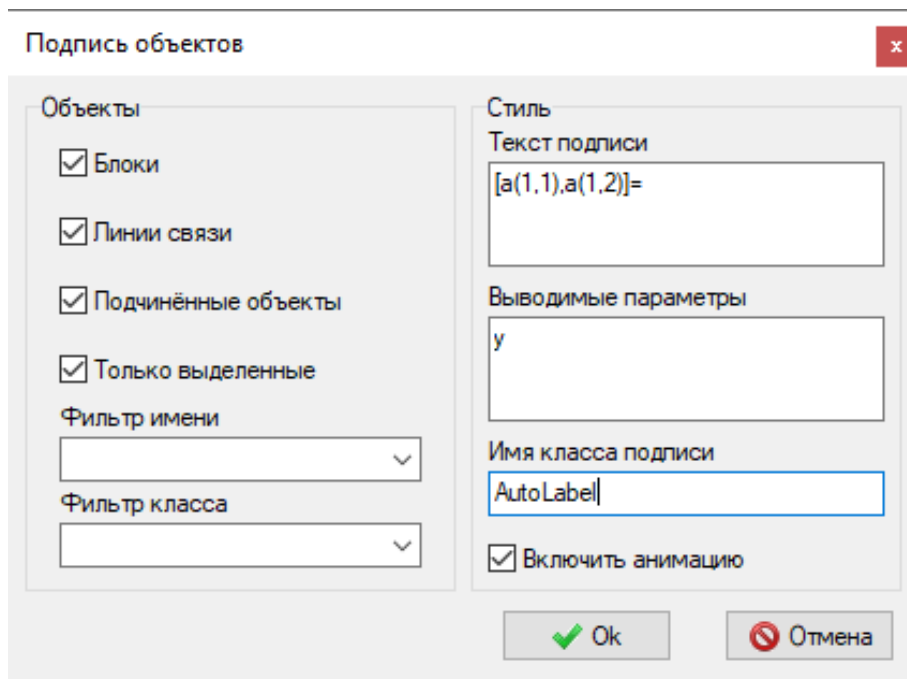


Рис. 6

Чтобы отображаемое значение изменялось при изменении значения параметра, должен быть активирован пункт **Включить анимацию** [1].

При нажатии кнопки **Ок** в схемном окне проекта появится блок **AutoLabel** с выведенным значением параметра **y**, формируемым на выходном порте **a1** блока **Язык программирования** (рис. 7).

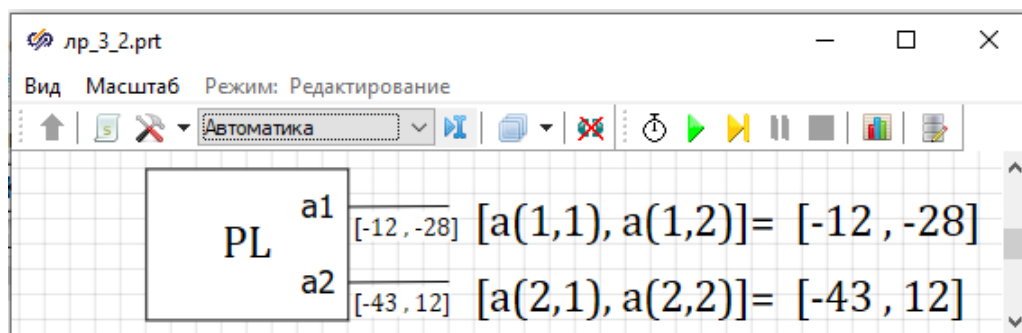


Рис. 7

Аналогично добавить блок **AutoLabel** на второй порт а2 блока **Язык программирования**. Несколько раз запустить программу на выполнение и убедиться, что на выходных портах блока **Язык программирования** и в соответствующих блоках **AutoLabel** синхронно появляются различные случайные значения.

Задание для отчета по лабораторной работе



Написать программу, формирующую матрицу **A1** со следующими значениями элементов для своего номера варианта N :

$$A1 = \begin{bmatrix} N-2 & N+5 & N-3 \\ N-20 & N+10 & N-1 \\ N+2 & N-12 & N-5 \end{bmatrix}.$$

Вывести полученные значения элементов строк матрицы в соответствующие выходные порты блока **Язык программирования**, а также в соответствующие блоки **AutoLabel**.



Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы, а также результаты ее выполнения в схемном окне проекта.

3. Создать новый проект в SimInTech на основе расчетной схемы решателя **Схема модели общего вида**.

Нажать кнопку «Скрипт»  из раздела **Слои** на панели окна редактирования проекта и написать программу в окне **Скрипт страницы**, в которой формируются матрицы и векторы с помощью различных функций SimInTech:

```
1 M=matrix(2,3); //нулевая матрица
- M1=matrix1(2,3); //матрица единиц
- const X = [1, 2, 3];
- M2=diag(X); // диагональная матрица
- M3=eye(3); //все элементы главной диагонали равны 1
- M4=linspace(0,30,7); //вектор арифметической прогрессии
```

Функция **matrix(n, m)** возвращает матрицу размерностью n на m , у которой все элементы равны нулю.

Функция **matrix1**(n, m) возвращает матрицу размерностью **n** на **m**, у которой все элементы равны единице.

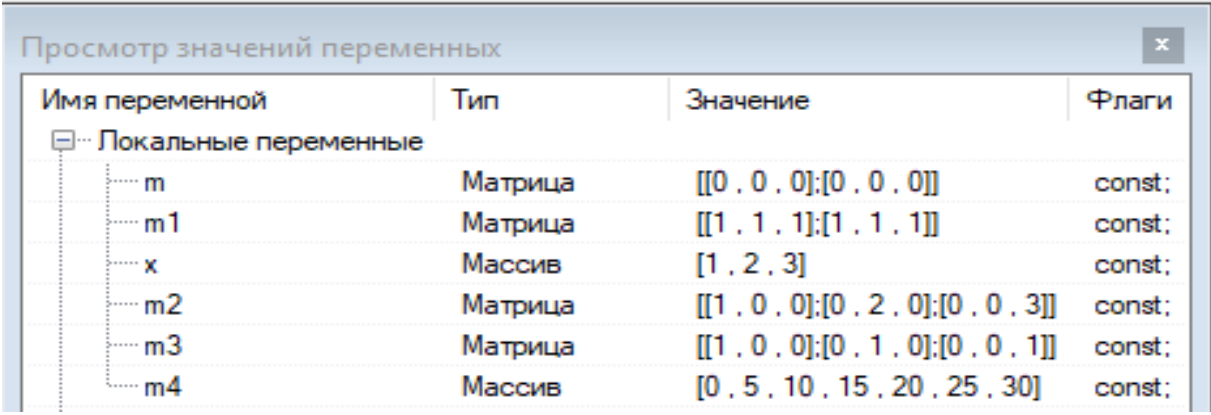
Функция **diag**(X) возвращает квадратную матрицу, у которой диагональные элементы равны элементам вектора X (диагональную матрицу). Все остальные элементы матрицы, стоящие вне главной диагонали, равны нулю.

Функция **eye**(n) возвращает матрицу размерностью n на n, у которой все диагональные элементы равны единице.

Функция **linspace**(xmin, xmax, n) возвращает вектор арифметической прогрессии от значения xmin до значения xmax. Размерность вектора определяется параметром n.

Сохранить проект в своей папке в виде файла под именем **лр_3_3** и запустить программу на выполнение, нажав клавишу «Выполнить скрипт» .

На рис. 8 показаны результаты работы этой программы в окне **Просмотр значений переменных**.



Имя переменной	Тип	Значение	Флаги
Локальные переменные			
m	Матрица	[[0 . 0 . 0];[0 . 0 . 0]]	const;
m1	Матрица	[[1 . 1 . 1];[1 . 1 . 1]]	const;
x	Массив	[1 . 2 . 3]	const;
m2	Матрица	[[1 . 0 . 0];[0 . 2 . 0];[0 . 0 . 3]]	const;
m3	Матрица	[[1 . 0 . 0];[0 . 1 . 0];[0 . 0 . 1]]	const;
m4	Массив	[0 . 5 . 10 . 15 . 20 . 25 . 30]	const;

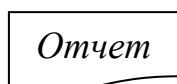
Рис. 8

Задание для отчета по лабораторной работе



Написать для своего варианта задания в окне **Скрипт страницы** программу для создания:

– вектор-строки **V1**, значения элементов которой представляют собой арифметическую прогрессию вида $V1 = (N + 1) (N + 2) \dots (N + 10)$.



Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы, а также результаты ее выполнения в окне **Просмотр значений переменных**.

4. Создать новый проект в SimInTech на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования**, открыть его диалоговое окно и записать в нем программу:

```
1  output a=matrix(3,2),m;
   m=-100;
   for (i=1,3,1)
     for (j=1,2,1)
       begin
         a[i,j]=trunc(100*rand)-50;
         if a[i,j]>m then m=a[i,j];
       end;
```

В программе инициализируются матрица a размером 3×2 и вещественная переменная m в качестве выходных переменных блока **Язык программирования**.

Затем переменной m присваивается значение, заведомо меньшее значений элементов далее формируемой матрицы a . Так как эти значения в программе могут принимать значения в диапазоне от -50 до 50 , то можно взять, например, $m = -100$.

Далее в цикле каждый элемент матрицы сравнивается со значением переменной m и, если выполняется условие оператора `if`, то переменной m будет присваиваться значение этого элемента матрицы.

Сохранить проект в своей папке в виде файла под именем **лр_3_4** и запустить программу на выполнение. Добавить линии связи от появившихся выходных портов и повторно запустить программу.

На рис. 9 приведены результаты работы программы, из которых видно, что выводимое в порт m значение является максимальным из всех значений элементов матрицы.

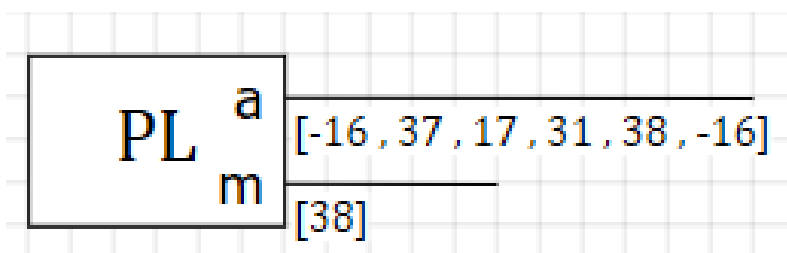


Рис. 9

Несколько раз запустить программу на выполнение и убедиться в правильности ее работы.

5. Для обработки векторов и матриц, а также для получения информации о них в SimInTech существует множество функций, применение которых позволяет существенно упростить программу. Например, для вычисления максимального значения из двух значений или из значений матрицы можно использовать функцию **max**.

Отредактировать ранее написанную программу, добавив в нее вещественную переменную **m1** в качестве выходной переменной блока **Язык программирования** и присвоить ей значение, равное **max(a)**:

```

1  output a=matrix(3,2),m,m1;
   m=-100;
   for (i=1,3,1)
     for (j=1,2,1)
       begin
         a[i,j]=trunc(100*rand)-50;
         if a[i,j]>m then m=a[i,j];
       end;
   m1= max(a);

```

Сохранить проект в своей папке в виде файла под именем **лр_3_5** и запустить программу на выполнение. Добавить линии связи от появившегося выходного порта **m1** и повторно запустить программу.

На рис. 10 приведены результаты работы программы, из которых видно, что значения, выводимые в порты **m1** и **m**, совпадают.

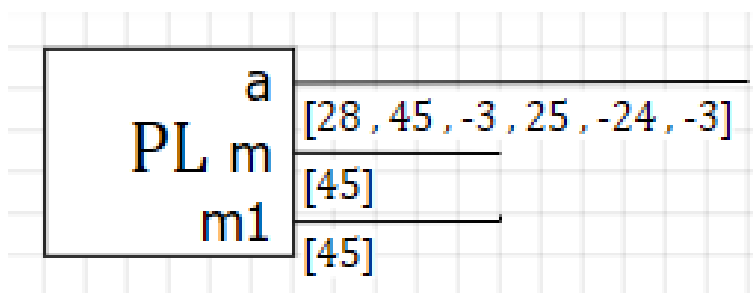


Рис. 10

Несколько раз запустить программу на выполнение и убедиться в правильности ее работы.

6. В табл. 1 приведены другие функции SimInTech, предназначенные для обработки векторов и матриц, а также для получения информации о них [1].

Таблица 1

Функция	Описание
transp(M) transp(V)	Функция производит транспонирование входной матрицы M или вектора V
det(M)	Функция возвращает значение детерминанта матрицы. Матрица M должна быть квадратной
inv(M)	Функция возвращает матрицу, являющуюся обратной матрице M . Матрица M должна быть квадратной. При умножении обратной матрицы на исходную матрицу получается единичная матрица (все диагональные элементы которой равны 1)
sort(V)	Функция производит сортировку массива по возрастанию элементов. Для комплексных векторов сортировка ведется по модулям чисел
invsort(V)	Функция производит сортировку массива по убыванию элементов. Для комплексных векторов сортировка ведется по модулям чисел
mean(V)	Функция возвращает среднее значение элементов входного вектора V . Элементы вектора V должны быть вещественными числами
activeelement(V)	Функция возвращает индекс первого ненулевого элемента в массиве V . Если все элементы нулевые, то возвращает ноль. Индексы начинаются с единицы
rows(M)	Функция возвращает количество строк матрицы
cols(M)	Функция возвращает количество столбцов матрицы
cols(V)	Функция возвращает количество элементов массива V

Создать новый проект в SimInTech на основе расчетной схемы решателя **Схема модели общего вида** и написать программу в окне **Скрипт страницы**:

```

1 | output a1[3,3],v1[1,10],v2[10,1];
-   a1=[[-2,5,-3];[-20,10,-1];[2,-12,-5]];
-   v1=linspace(1,10,10); //вектор-строка
-   v=linspace(100,10,10); //массив
-   v2=transp(v); //вектор-столбец из вектора-строки
-   a1_t=transp(a1); //транспонированная матрица
-   a1_inv=inv(a1); //обратная матрица
-   aa=a1*a1_inv; //единичная матрица
-   s=sort(v); //сортировка вектора по возрастанию
10 s1=sort(a1); //сортировка матрицы по возрастанию
-   vcr=mean(v1); //среднее значение элементов вектора

```

В программе инициализируются матрица **a1** размером 3×3 , вектор-строка **v1** и вектор-столбец **v2** длиной 10.

Значения элементов матрицы **a1** присваиваются внутри внешних квадратных скобок путем указания их внутри квадратных скобок через запятую для строк, которые разделяются точкой с запятой.

Значения элементов вектор-строки **v1** и массива **v** задаются в виде арифметической прогрессии с помощью функции **linspace**.

Значения элементов вектор-столбца **v2** формируются с помощью функции транспонирования вектора **transp(v)**, которая заменяет строку на столбец. При транспонировании матрицы **a1** каждая из строк заменяется на столбец.

Далее в программе для матрицы **a1** с помощью функции **inv** формируется обратная матрица **a1_inv**. Затем переменной **aa** присваивается значение произведения матрицы **a1** и обратной матрицы **a1_inv**, которое возвращает единичную матрицу.

С помощью функции **sort** выполняется сортировка в порядке возрастания элементов вектора **v** и матрицы **a1**, а функция **mean** возвращает среднее значение элементов вектора **v1**.

Сохранить проект в своей папке в виде файла под именем **лр_3_6** и запустить программу на выполнение.

На рис. 11 приведены результаты работы программы в окне **Просмотр значений переменных**.

Просмотр значений переменных		
Имя переменной	Тип	Значение
☐ Локальные пер...		
... a1	Матрица	<code>[-2 , 5 , -3];[-20 , 10 , -1];[2 , -12 , -5]]</code>
... v1	Матрица	<code>[1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10]]</code>
... v2	Матрица	<code>[[100];[90];[80];[70];[60];[50];[40];[30];[20];[10]]</code>
... v	Массив	<code>[100 , 90 , 80 , 70 , 60 , 50 , 40 , 30 , 20 , 10]</code>
... a1_t	Матрица	<code>[-2 , -20 , 2];[5 , 10 , -12];[-3 , -1 , -5]]</code>
... a1_inv	Матрица	<code>[[0.059273423 , -0.0583174 , -0.023900574];[0.09751434</code>
... aa	Матрица	<code>[[1 , 0 , 5.5511151E-17];[0 , 1 , 2.7755576E-17];[0 , 0 , 1]]</code>
... s	Массив	<code>[10 , 20 , 30 , 40 , 50 , 60 , 70 , 80 , 90 , 100]</code>
... s1	Массив	<code>[-20 , -12 , -5 , -3 , -2 , -1 , 2 , 5 , 10]</code>
... vcr	Вещест...	5.5

Рис. 11

Обратите внимание, что для матрицы **a1_inv** на рис. 11 выведены только элементы первой строки и один элемент второй строки. Значения некоторых элементов единичной матрицы **aa** приводятся с учетом погрешности расчета в формате с плавающей запятой.

Задание для отчета по лабораторной работе



Написать для своего варианта задания N программу из п. 6 для следующих массивов:

– матрицы **A1**:

$$\mathbf{A1} = \begin{bmatrix} N-2 & N+5 & N-3 \\ N-20 & N+10 & N-1 \\ N+2 & N-12 & N-5 \end{bmatrix};$$

– вектор-строки **V1**, значения элементов которой заданы в виде арифметической прогрессии:

$$\mathbf{V1} = (N+1) (N+2) \dots (N+10);$$

– вектор-столбца **V2**, значения элементов которой заданы в виде арифметической прогрессии:

$$\mathbf{V2} = (100+N) (90+N) (80+N) \dots (10+N).$$



Привести примеры использования других функций из табл. 1 для обработки массивов для своего варианта задания.

Отчет

Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы, а также результаты ее выполнения в окне **Просмотр значений переменных**.

7. В языке программирования SimInTech могут выполняться как традиционные действия над векторами и матрицами, предусмотренные векторным вычислением в математике, так и их поэлементные преобразования. При поэлементном сложении, вычитании, умножении и делении векторов и матриц они должны иметь одинаковую размерность.

Создать новый проект в SimInTech на основе расчетной схемы решателя **Схема модели общего вида** и написать программу в окне **Скрипт страницы**:

```
1 | output v1[3],v2[3];  
  | v1=[1,2,3]; v2=[3,4,5];  
  | s=v1+v2; //Поэлементное сложение векторов  
  | r=v1-v2; //Поэлементное вычитание векторов  
  | p=v1*v2; //Поэлементное умножение векторов  
  | d=v1/v2; //Поэлементное деление векторов
```

Сохранить проект в своей папке в виде файла под именем **лр_3_7** и запустить программу на выполнение.

На рис. 12 приведены результаты работы программы в окне **Просмотр значений переменных**.

Имя переменной	Тип	Значение	Флаги
☐ Локальные переменные			
v1	Массив	[1, 2, 3]	output;
v2	Массив	[3, 4, 5]	output;
s	Массив	[4, 6, 8]	
r	Массив	[-2, -2, -2]	
p	Массив	[3, 8, 15]	
d	Массив	[0.33333333, 0.5, 0.6]	

Рис. 12

Создать новый проект в SimInTech на основе расчетной схемы решателя **Схема модели общего вида**. Нажать кнопку «Скрипт» из раздела **Слои** на панели окна редактирования проекта и написать программу в окне **Скрипт страницы**:

```

1 | output a1[2,3],a2[2,3];
  | a1=[[1,5,-3];[4,10,-1]];
  | a2=[[-1,5,-3];[10,10,3]];
  | s=a1+a2; //Поэлементное сложение матриц
  | r=a1-a2; //Поэлементное вычитание матриц
  | p=a1.*a2; //Поэлементное умножение матриц
  | d=a1./a2; //Поэлементное деление матриц

```

Сохранить проект в своей папке в виде файла под именем **лр_3_8** и запустить программу на выполнение. На рис. 13 приведены результаты работы программы в окне **Просмотр значений переменных**.

Для векторного умножения и деления матриц количество столбцов первой матрицы должно совпадать с количеством строк второй матрицы. Деление производится умножением на обратную матрицу.

Просмотр значений переменных			
Имя переменной	Тип	Значение	Флаги
Локальные переменные			
a1	Матрица	[[1 , 5 , -3];[4 , 10 , -1]]	output;
a2	Матрица	[[-1 , 5 , -3];[10 , 10 , 3]]	output;
s	Матрица	[[0 , 10 , -6];[14 , 20 , 2]]	
r	Матрица	[[2 , 0 , 0];[-6 , 0 , -4]]	
p	Матрица	[[-1 , 25 , 9];[40 , 100 , -3]]	
d	Матрица	[[-1 , 1 , 1];[0.4 , 1 , -0.33333333]]	

Рис. 13

Создать новый проект в SimInTech на основе расчетной схемы решателя **Схема модели общего вида** и написать программу в окне **Скрипт страницы**:

```

1  output a1[2,2],a2[2,2];
   a1=[[1,-3];[4,11]];
   a2=[[-1,5];[6,-3]];
   p=a1*a2;           //умножение матриц
   d=a1/a2;           //деление матрицы a1 на a2
   d1=a1*inv(a2);     //деление матрицы a1 на a2
   d2=a1*a2^-1;       //деление матрицы a1 на a2

```

Сохранить проект в своей папке в виде файла под именем **лр_3_9** и запустить программу на выполнение. На рис. 14 приведены результаты работы программы в окне **Просмотр значений переменных**.

Просмотр значений переменных			
Имя переменной	Тип	Значение	
Локальные ...			
a1	Матрица	[[1 , -3];[4 , 11]]	
a2	Матрица	[[-1 , 5];[6 , -3]]	
p	Матрица	[[-19 , 14];[62 , -13]]	
d	Матрица	[[-0.55555556 , 0.07407407];[2.8888889 , 1.1481481]]	
d1	Матрица	[[-0.55555556 , 0.07407407];[2.8888889 , 1.1481481]]	
d2	Матрица	[[-0.55555556 , 0.07407407];[2.8888889 , 1.1481481]]	

Рис. 14

8. Обращение матрицы используется для решения системы линейных уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

при задании ее в матричной форме как

$$\mathbf{AX} = \mathbf{B}, \quad (1)$$

где $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}; \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}; \mathbf{A} = [a_{ij}]; i, j = \overline{1, n}.$

Система (1) имеет единственное решение, равное

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B},$$

если ее главный определитель отличен от нуля $\det(\mathbf{A}) \neq 0$.

Задание для отчета по лабораторной работе



Написать для своего варианта задания N программу для решения системы линейных уравнений:

– главный определитель которой равен:

$$\mathbf{A} = \begin{bmatrix} N-2 & N+5 & N-3 \\ N-20 & N+10 & N-1 \\ N+2 & N-12 & N-5 \end{bmatrix};$$

– вектор-столбец свободных членов равен:

$$\mathbf{B} = \begin{bmatrix} N+3 \\ N-2 \\ N+1 \end{bmatrix}.$$

Отчет

Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы, а также результаты ее выполнения в окне **Просмотр значений переменных**.

Контрольные вопросы

1. Каким образом в SimInTech могут задаваться размерности массивов?
2. Каким образом в SimInTech могут задаваться значения элементов массивов?
3. Как в SimInTech формируются упорядоченные массивы?
4. Какие функции можно использовать для формирования векторов и матриц в SimInTech?
5. Какие функции SimInTech можно использовать для обработки векторов и матриц, для получения информации о них?
6. Какие способы обращения к элементам, строкам и столбцам заданной матрицы существуют в системе SimInTech?
7. Каким образом в SimInTech могут выполняться традиционные действия над векторами и матрицами, предусмотренные векторным вычислением в математике, и поэлементные преобразования векторов и матриц?
8. Как в SimInTech решается система линейных уравнений, заданная в матричной форме?

Лабораторная работа № 4

ВИЗУАЛИЗАЦИЯ ДАННЫХ В СРЕДЕ SimInTech

Цель работы: знакомство с возможностями по визуализации данных и построению графиков.

Рабочее задание

1. Запустить программу SimInTech, создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования** из закладки **Динамические** и блок **График Y от X** из закладки **Вывод данных** (рис. 1).

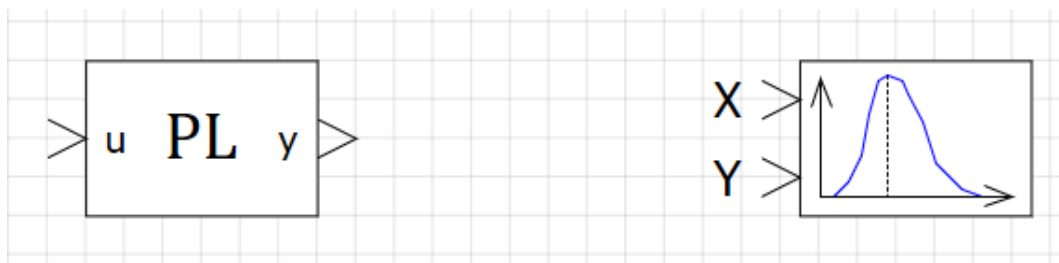


Рис. 1

Записать в диалоговом окне блока **Язык программирования** программу:

```
x=linspace(-5,5,11);  
y = x^2;  
output x,y;
```

В программе с помощью функции **linspace** формируется 11 значений вектора x , меняющихся от -5 до 5 с шагом 1 , а затем вычисляются значения функции $y = x^2$.

Значения векторов x и y выводятся в соответствующие порты блока **Язык программирования**.

Сохранить проект в своей папке в виде файла под именем **лр_4_1** и запустить программу на выполнение.

Соединить появившиеся выходные порты блока **Язык программирования** с соответствующими входными портами блока **График Y от X**.

Если повторно запустить программу на выполнение и нажать клавишу «Показать значения на линиях связи» в окне проектов, то под линиями связи выведутся соответствующие значения выходных переменных x и y блока **Язык программирования** (рис. 2).

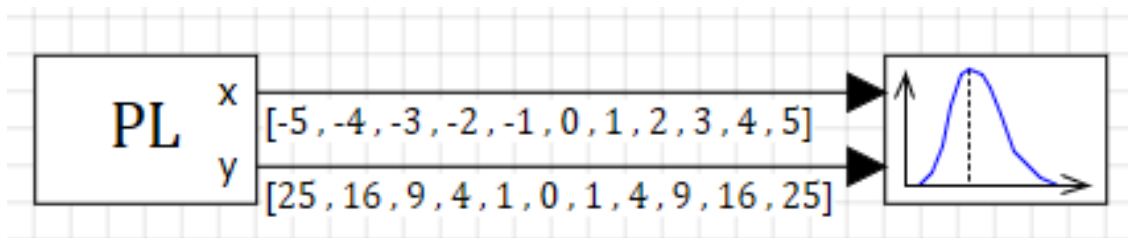


Рис. 2

Результаты моделирования можно увидеть также на экране блока **График Y от X**, если выполнить по нему двойной щелчок ЛКМ (рис. 3).



Рис. 3

Переместить курсор в центральную часть окна графика и сделать щелчок правой клавишей мыши. В результате появится контекстное меню, пункт **Свойства** которого позволяет открыть диалоговое окно с заголовком **Свойства графика**, в котором задаются настройки его параметров.

На рис. 4 приведен пример окна **Свойства графика** со стандартными настройками по умолчанию, в котором на закладке **Графики и оси** был выбран из списка **Стиль линии** на **Штриховая** и вид **Точки** на **Квадратики**.

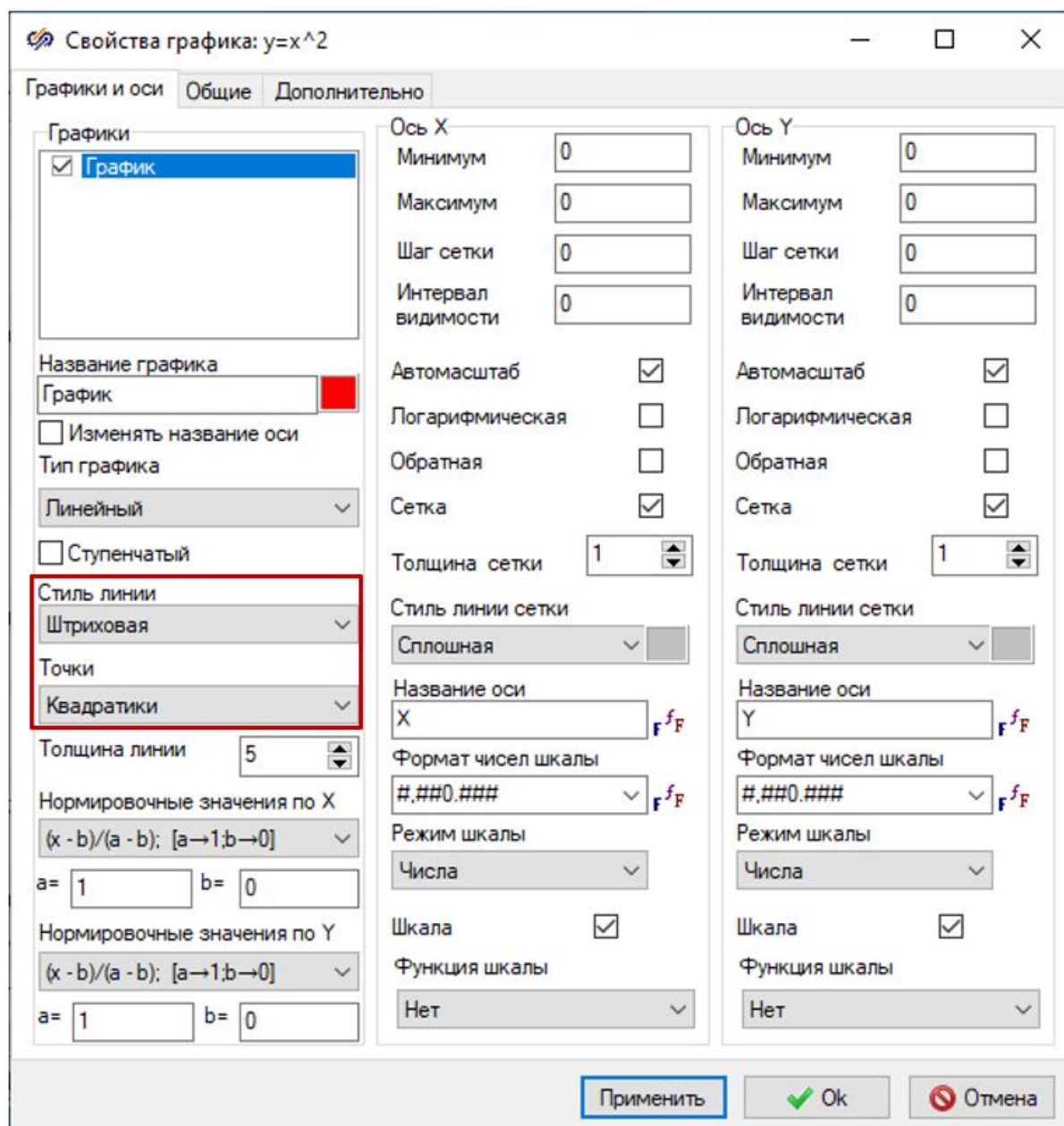


Рис. 4

Обратите внимание, что если убрать флажок в окне **Автомасштаб**, то можно вывести в окно часть графика, указав в окнах **Минимум** и **Максимум** требуемые координаты по оси **Y** и **X**.

На рис. 5 приведен пример окна **Свойства графика**, в котором, для того чтобы убрать легенду графика, был удален флажок в окне

справа от настройки **Показывать легенду**. Легенду целесообразно применять, если на экране блока **График Y от X** выводятся более одного графика.

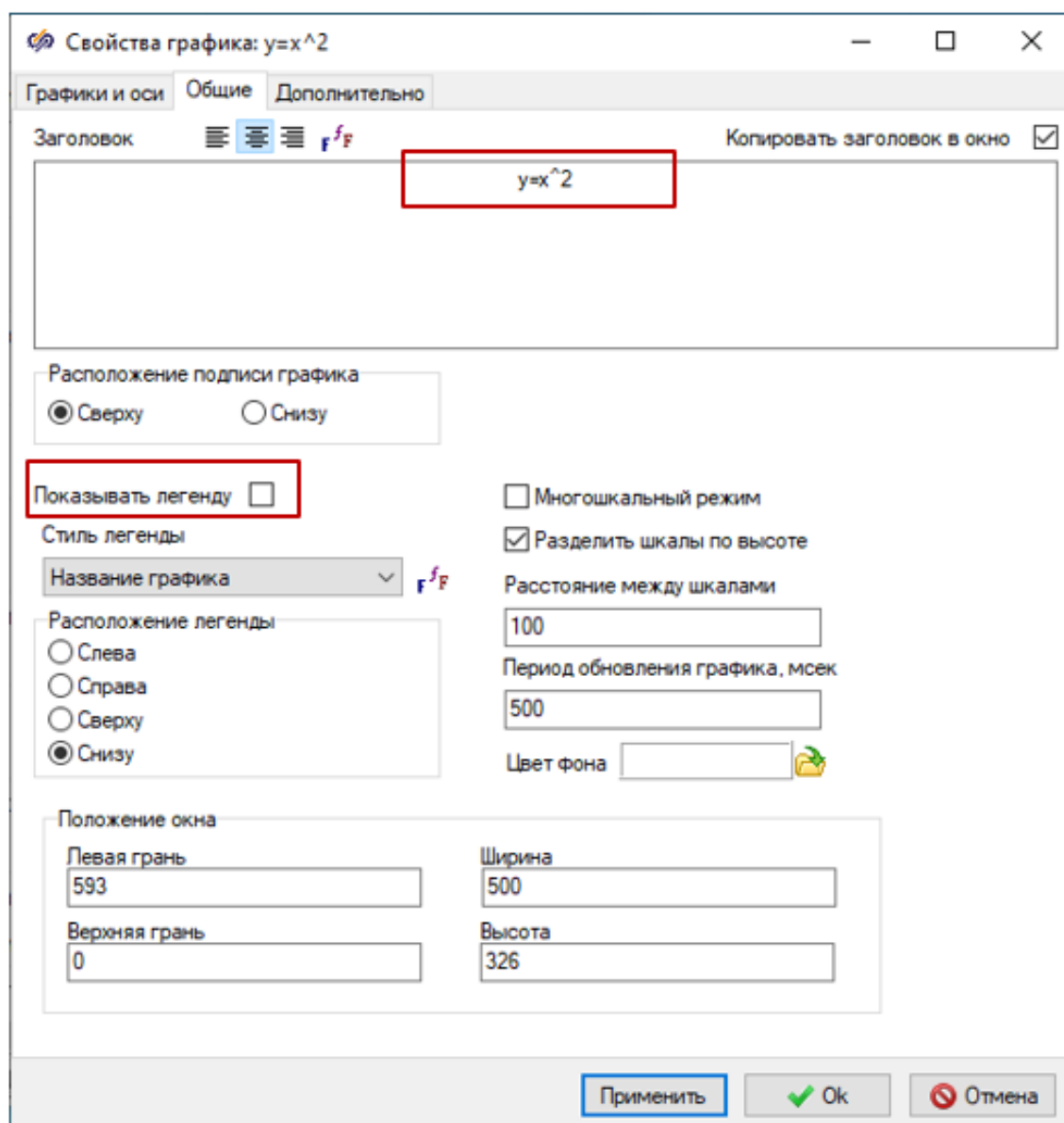


Рис. 5

Кроме того, был изменен **Заголовок** графика на $y=x^2$. При необходимости можно изменить форматирование заголовка: варианты выравнивания и различные настройки шрифта приведены в окне **Редактор шрифта**.

Все внесенные изменения в настройки в окне **Свойства графика** вступят в силу после нажатия на кнопку **Применить**. На рис. 6 показано, как изменится внешний вид графика при этом.

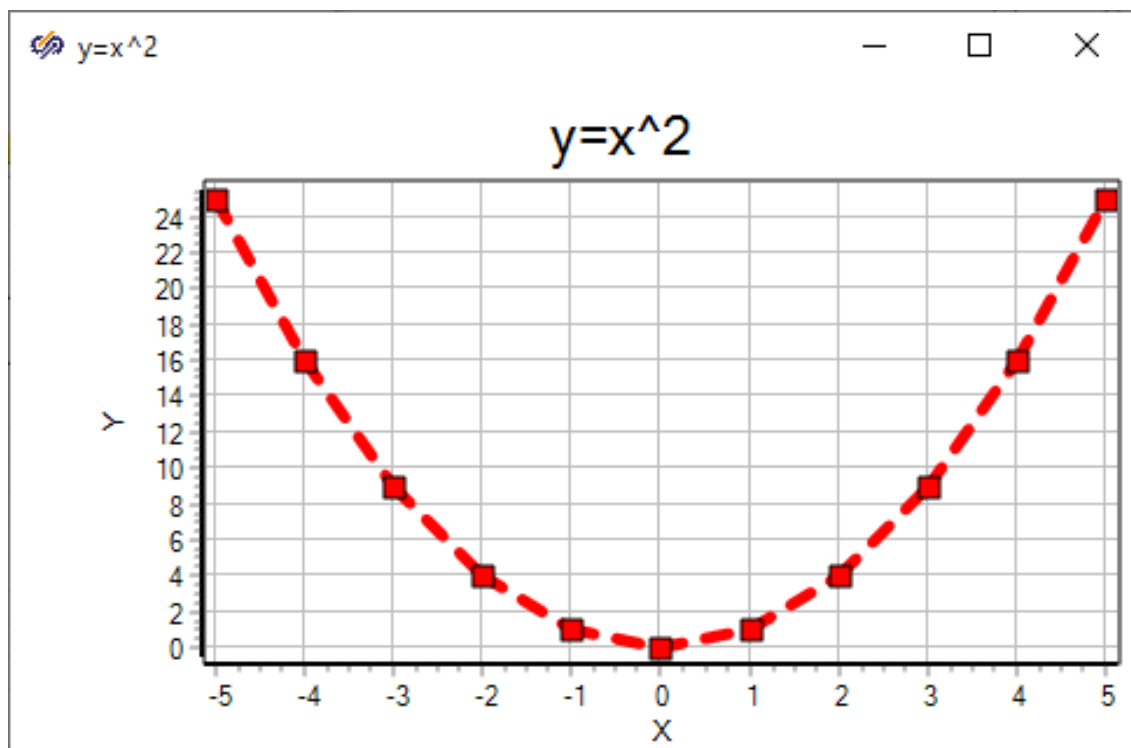


Рис. 6

Задание для отчета по лабораторной работе



Построить в декартовой системе координат для заданного варианта задания график функции в указанном диапазоне по оси x и с заданным шагом (табл. 1). Задать в окне свойств графического окна настройки по аналогии с рис. 4, 5.



Вставить в отчет по лабораторной работе формулировку заданий, текст полученных программ, а также результаты их выполнения на экране блока **График Y от X** .



Рассмотреть другие настройки, которые можно задавать в окне свойств графического окна, применить их для своего графика, вставить в отчет по лабораторной работе не менее трех вариантов графика с различными настройками.

Таблица 1

Номер варианта	Функция	Диапазон изменения x	Шаг изменения x
1	2	3	4
1	$y = x^2 + x - 2 $	0–10	2
2	$y = 13x^2$	0–24	8
3	$y = (1 - x) x + 1 $	–10–8	2
4	$Y = \frac{1}{x + 4}$	–1–3	0,5
5	$y = 1 - e^{2x}$	0–5	0,1
6	$Y = \frac{x + 44}{x - 23}$	–8–8	1
7	$Y = \frac{x}{x - 31}$	–10–30	5
8	$y = x^2 - 2 x + 1$	0–16	2
9	$y = (x^3 - x)$	0–200	20
10	$y = x^2 + x $	0–24	3
11	$y = 1 - e^{2x}$	0–5	0,1
12	$y = (1 + x)(2 - x)$	0–36	3
13	$y = (x - 1)(2 - x)$	0–5	0,1
14	$y = \frac{x^3 + x}{ x }$	10–50	2
15	$y = \frac{x^3 - x^2}{2 x - 1 }$	2–18	1
16	$Y = \frac{2x}{x + 100}$	–80–80	10
17	$Y = (3x^2 + 43)$	0–40	4
18	$y = 1 - x^2$	0–1	0,01
19	$y = 1 + e^x$	0–5	0,1
20	$y = (x - 1) x - 2 $	0–36	3
21	$y = x^2 + x - 2 $	–10–10	4

1	2	3	4
22	$y = 1 - e^{2x}$	0–3	0,3
23	$y = x^4 - 2x^2 + 2$	0–2	0,1
24	$y = 13x^2$	–10–10	2
25	$y = 1 + 2e^{2x}$	0–2	0,1
26	$y = x^2 + x $	–5–5	1
27	$y = (x^3 - x)$	0–30	2
28	$y = (x - 1)(2 - x)$	–5–5	0,1
29	$y = (1 - x) x + 1 $	0–10	0,5
30	$y = x^2 + x $	0–14	2
31	$y = (1 - x) x + 1 $	–10–8	2
36	$y = 1 - x^2$	0–1	0,01
37	$y = x^2 + 2x + 2$	–15–15	0,3
38	$y = x^2 + x - 2 $	–10–10	0,2
39	$y = (x^3 - x)$	0–30	2
40	$Y = \frac{x + 44}{x - 23}$	–8–8	1

2. Создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования** из закладки **Динамические** и блок **График Y от X** из закладки **Вывод данных**. Записать в диалоговом окне блока **Язык программирования** программу:

```
x=linspace(0,4*pi,61);
fi=linspace(pi/6,pi/6,61);
y = sin(x/2+fi);
output x,y;
```

В программе строится в декартовой системе координат один период тригонометрической функции $y = \sin(x/2 + 30^\circ)$ по 61 точке на оси x . Аргумент тригонометрических функций в SimInTech должен задаваться в радианах.

В программе аргумент тригонометрической функции без учета начальной фазы формируется в виде вектора длиной 61 с помощью функции **linspace**. Начальная фаза гармонического сигнала **fi** также формируется с помощью функции **linspace** в виде вектора длиной 61.

Значения векторов **x** и **y** выводятся в соответствующие порты блока **Язык программирования**.

Сохранить проект в своей папке в виде файла под именем **лр_4_2** и запустить программу на выполнение. Соединить появившиеся выходные порты блока **Язык программирования** с соответствующими входными портами блока **График Y от X**.

Повторно запустить программу, выполнить двойной щелчок ЛКМ по блоку **График Y от X**. На рис. 7 приведены результаты моделирования на экране блока **График Y от X** после изменения в окне **Свойств графика** заголовка и выбора параметра вида **Точки** на **Квадратики**.

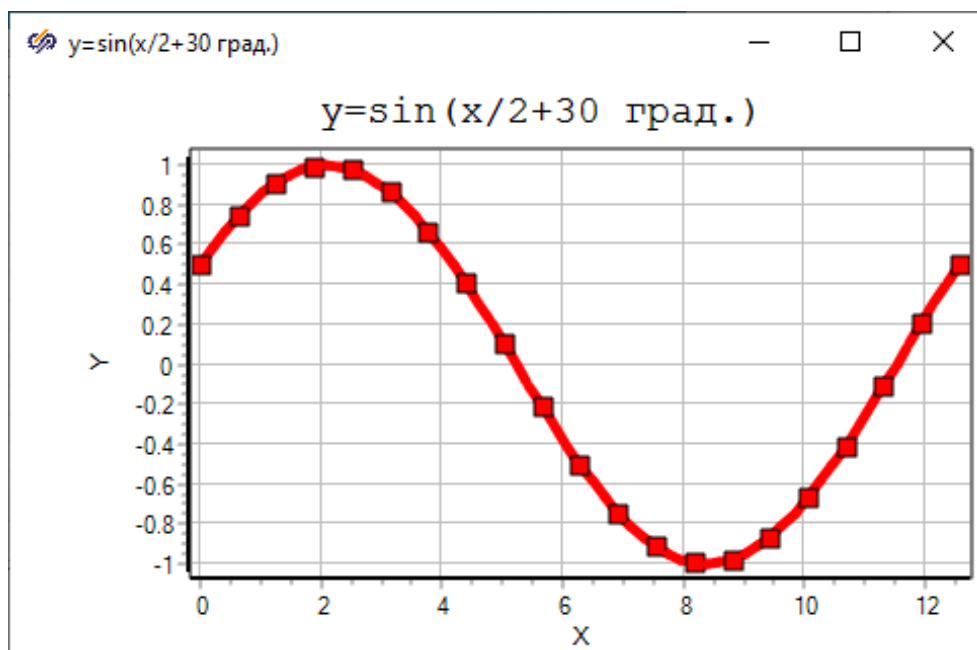


Рис. 7

Задание для отчета по лабораторной работе



Построить в декартовой системе координат для указанного варианта задания один период тригонометрической функции (табл. 2) по 51 точке по оси **x**. Задать в окне свойств графического окна настройки по своему усмотрению.

Вставить в отчет по лабораторной работе формулировку заданий, текст полученных программ, а также результаты их выполнения на экране блока **График Y от X**.

Таблица 2

Номер варианта	Функция	Номер варианта	Функция
1	$y = \cos(x/2 + 60^\circ)$	21	$y = \sin(x/2 + 30)$
2	$y = \left(\cos^4 \frac{x}{2} + 2 \right)$	22	$y = \left(1 + \sin^2 \frac{x}{3} \right)$
3	$y = \cos(4x - 60^\circ)$	23	$y = \sin(3x + 60)$
4	$y = \cos 2x + \sin x$	24	$y = 2 \cos x - \sin x$
5	$y = 2 \sin x - \cos x$	25	$y = \sin 2x + 2 \cos x$
6	$y = \cos(2x) + \sin x$	26	$y = 1 + \cos^2(2x)$
7	$y = \cos(3x - 45^\circ)$	27	$y = \cos(x - 15)$
8	$y = 2 \sin(x + 90^\circ)$	28	$y = 2 \sin x + 2 \cos x$
9	$y = \sin(2x - 30^\circ)$	29	$y = 2 + \sin(x)$
10	$y = \cos(4x - 45^\circ)$	30	$y = \sin(x + 45)$
11	$y = 2 \sin(x + 90^\circ)$	31	$y = \cos(5x + 45^\circ)$
12	$y = \sin(3x + 60^\circ)$	32	$y = \cos(x + 60)$
13	$y = x + \sin x $	33	$y = \sin x + \sin x $
14	$y = \cos x + \cos x $	34	$y = \sin x + \sin x $
15	$y = 2 \sin x \cos x $	35	$y = 2 \cos x \sin x $
16	$y = \cos(4x - 60^\circ)$	36	$y = \cos(4x - 60)$
17	$y = \cos(3x - 30^\circ)$	37	$y = \sin(x + 45^\circ)$
18	$y = \sin(5x + 45^\circ)$	38	$y = \cos(2x + 60)$
19	$y = \sin(3x + 60^\circ)$	39	$y = \cos(x - 15)$
20	$y = 2 \sin x \cos x $	40	$y = \sin(3x + 30)$

3. Сохранить предыдущий проект из п. 2 в своей папке в виде файла под именем **лр_4_3**.

Удалить все линии связи, открыть окно свойств блока **График Y от X** и задать на закладке **Свойства** значение **2** для свойства **Количество графиков** (рис. 8).

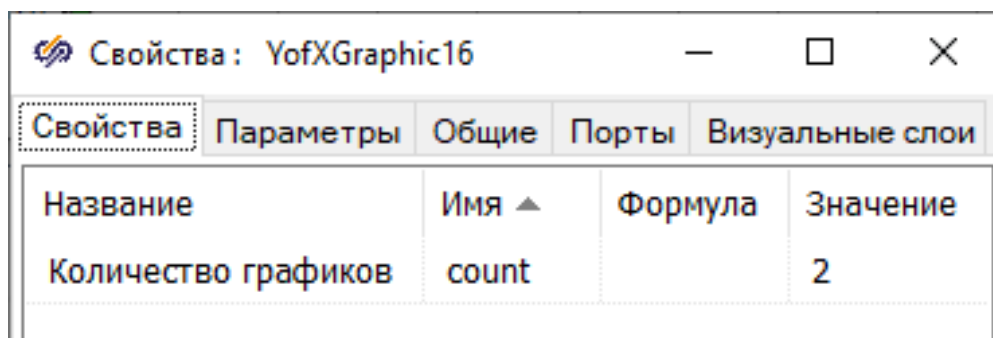


Рис. 8

Записать в диалоговом окне блока **Язык программирования** программу:

```
x=linspace(-5,5,11);
y = x^2;
y1=x^3;
output x,y,y1;
```

Сохранить изменения в проекте и запустить программу на выполнение. В результате у блоков появятся дополнительные выходные порты (рис. 9).



Рис. 9

Соединить выходные порты блока **Язык программирования** с соответствующими входными портами блока **График Y от X** в соответствии с рис. 10.

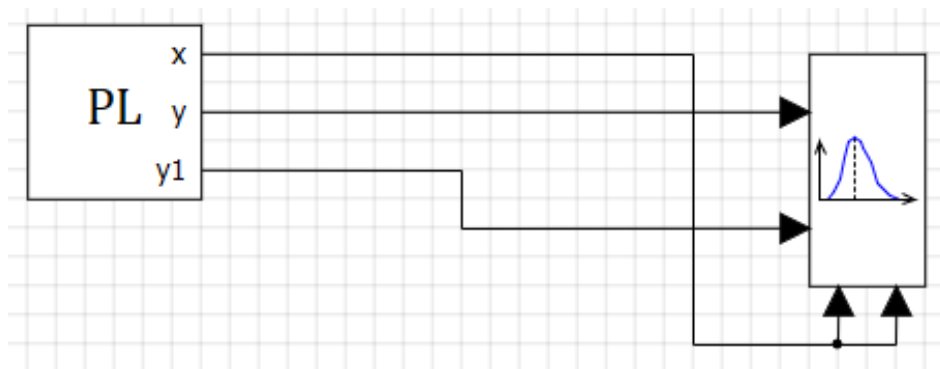


Рис. 10

Сохранить изменения в проекте и запустить программу на выполнение. Результаты моделирования представлены на рис. 11, 12.

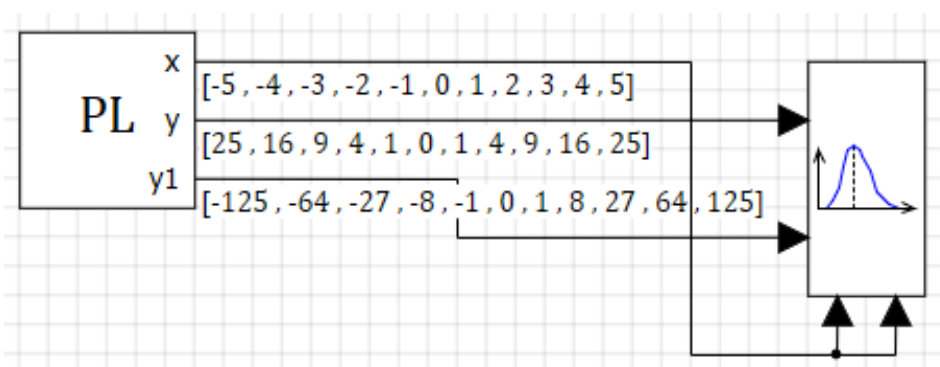


Рис. 11

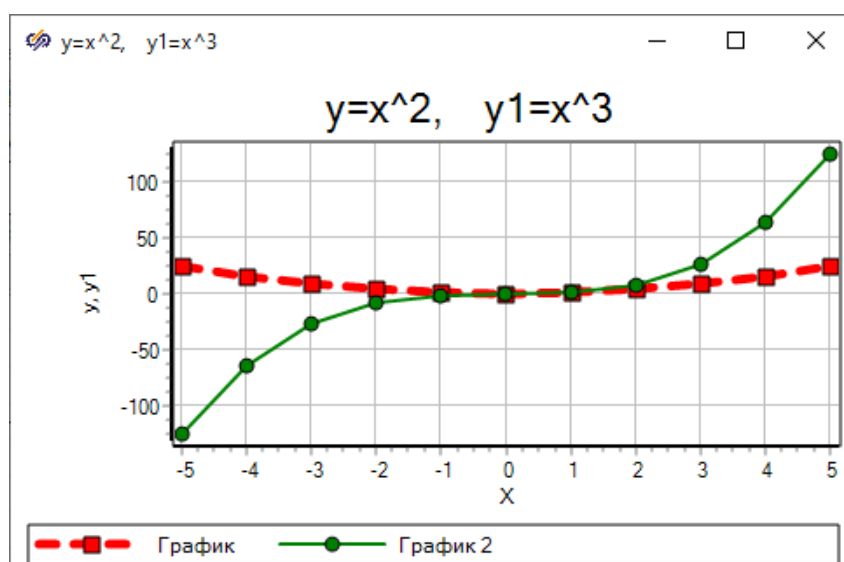


Рис. 12

Отредактировать настройки в окне **Свойства графика** в соответствии с рис. 12, изменив название графика, название оси **Y** и добавив легенду.

Задание для отчета по лабораторной работе.



Отчет

Построить в декартовой системе координат для заданного варианта задания два графика: $y=f(x)$ из таблицы 1 и $y=2*f(x)$ в указанном диапазоне по оси x и с заданным шагом.

Вставить в отчет по лабораторной работе формулировку задания, текст полученной программы, а также результаты ее выполнения на экране блока **График Y от X**.

4. Создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования** из закладки **Динамические** и блок **Временной график** из закладки **Вывод данных**. Записать в диалоговом окне блока **Язык программирования** программу:

```
output i[];var R=[2,10];
E=10;C=0.0126;L=2; g=R/(2*L);
w0=[1/sqrt(L*C);1/sqrt(L*C)];
wcb=sqrt(w0^2-g^2);
i = E/(L*wcb)*exp(-time*g)*sin(wcb*time);
```

В качестве выходной переменной блока **Язык программирования** задается вектор тока i нулевой длины, значения которого формируются для двух значений вектора R как функция времени t , которое задается в программе с помощью системной переменной **time**, возвращающей модельное время.

В программе рассчитывается, как будет протекать переходный процесс в RLC цепи при подключении ее к источнику постоянного напряжения $E = 10$ В для двух значениях сопротивления R в омах, равных 2 и 10. Из курса теории электрических цепей известно, что значение тока в этой цепи будет изменяться в соответствии со следующим выражением:

$$i = \frac{E}{L \cdot w_{cb}} e^{-t \cdot g} \sin(w_{cb} \cdot t),$$

где $g = \frac{R}{2L}$ – коэффициент затухания; $w_{cb} = \sqrt{w_0^2 - g^2}$, – частота свободных колебаний; $w_0 = \frac{1}{\sqrt{L \cdot C}}$ – резонансная частота.

Сохранить проект в своей папке в виде файла под именем **лр_4_4** и запустить программу на выполнение.

Соединить появившийся выходной порт блока **Язык программирования** с соответствующим входным портом блока **Временной график**.

Повторно запустить программу на выполнение и посмотреть результаты моделирования на экране блока **Временной график** (рис. 13), выполнив по нему двойной щелчок ЛКМ.

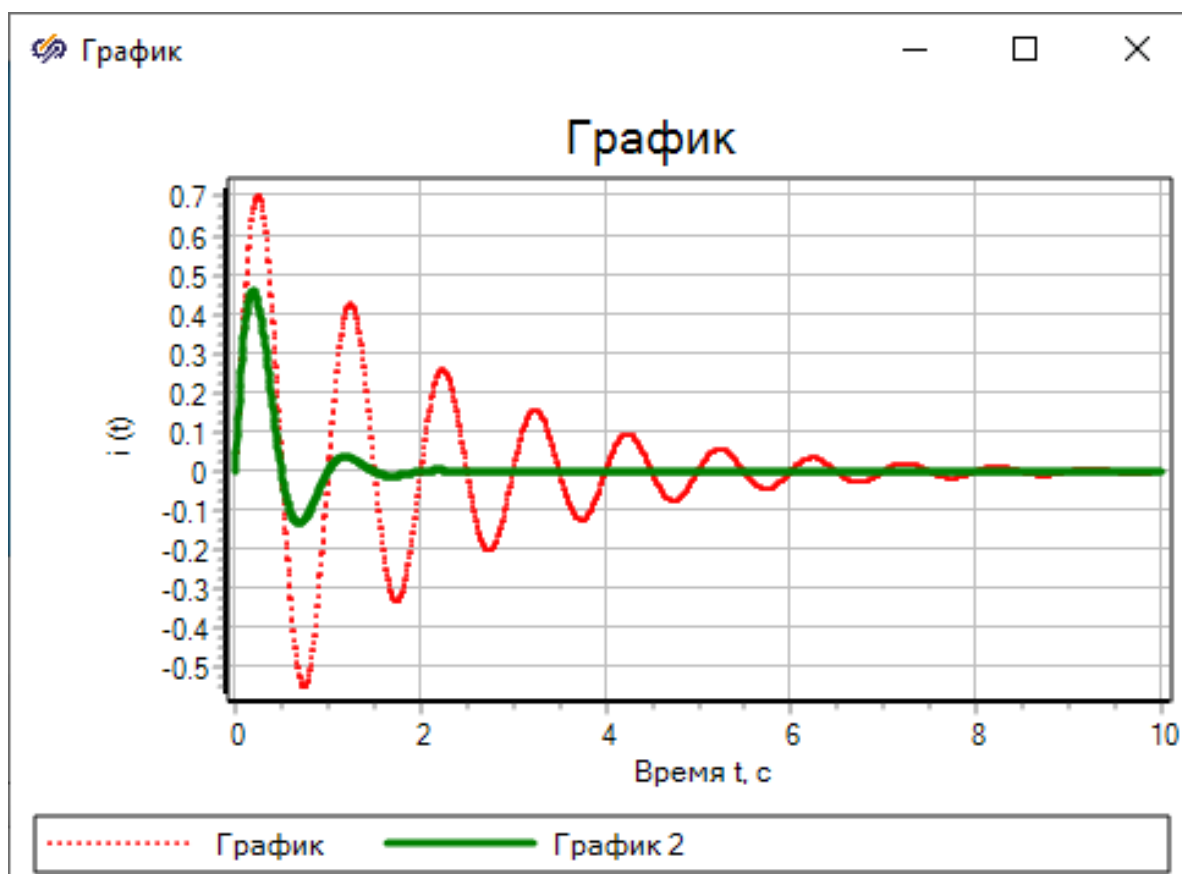


Рис. 13

На рис. 13 задано конечное время расчета, равное 10 с. Из рис. 13 видно, как будет протекать переходный процесс в RLC цепи при двух значениях сопротивления R . Чем больше R , тем быстрее будет затухать переходный процесс.

Для изменения конечного времени расчета нужно нажать кнопку **Параметры расчета**  схемного окна проектов. В результате откроется диалоговое окно **Параметры проекта** (рис. 14), на закладке **Параметры расчета** которого необходимо задать требуемые значения параметров расчета.

Параметры проекта: H:\умк\компьютерные те...			
Параметры расчёта			
Синхронизация Рестарт База данных Вид			
Название ▲	Имя	Форму...	Значение
Основные параметры			
Абсолютная ошибка	abserr		1E-6
Конечное время расчёта	endtime		10
Максимальный шаг	hmax		0.01
Метод интегрирования	intmet		Эйлера
Минимальный шаг	hmin		0.01
Начальное время расчёта	starttime		0

Рис. 14

На рис. 15 показано, как изменятся результаты моделирования, если задать параметру **Конечное время расчета** значение 5.

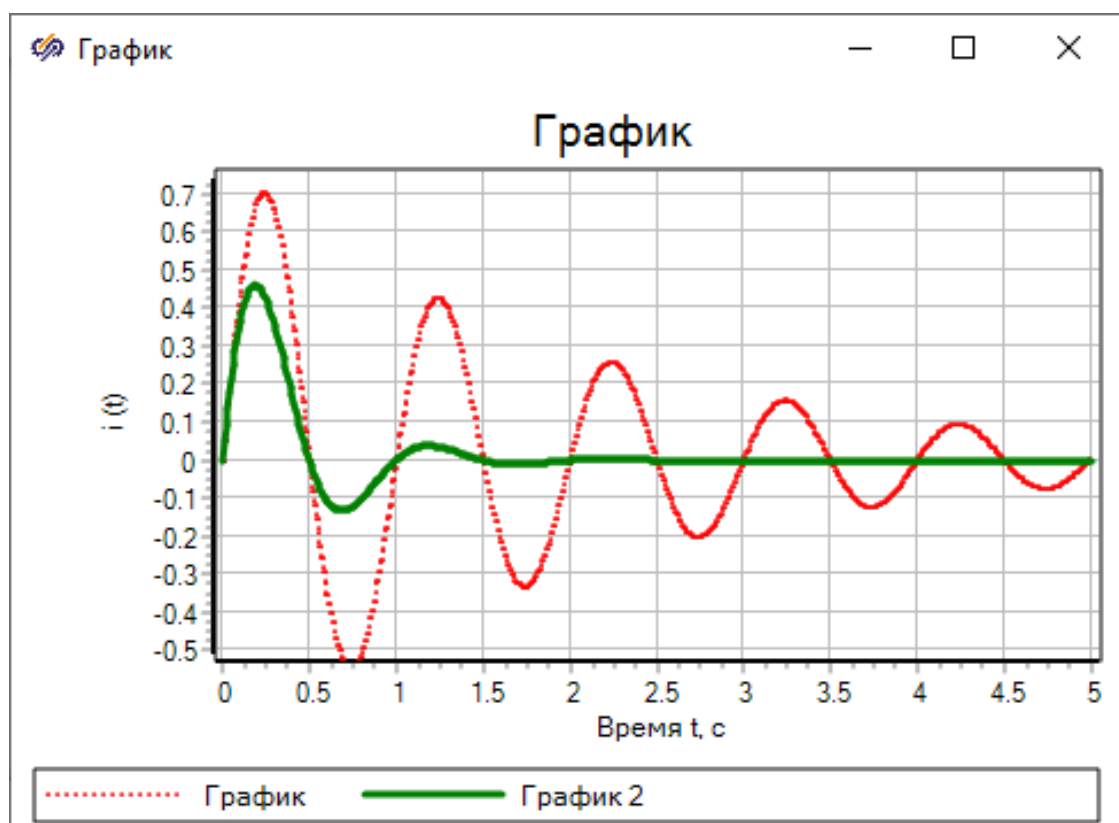


Рис. 15

Задание для отчета по лабораторной работе



Построить в декартовой системе координат для заданного варианта задания N семейство кривых

$$u = Ne^{-t},$$

для N , заданного в виде вектор-строки $[N, 2N, 3N]$.

Задать значение параметру **Конечное время расчета**, равное 5.



Вставить в отчет по лабораторной работе формулировку заданий, текст полученных программ, а также результаты их выполнения на экране блока **Временной график**.

5. Создать новый проект на основе расчетной схемы решателя **Схема модели общего вида**. Добавить в окно проекта блок **Язык программирования** из закладки **Динамические** и блок **График Y от X** из закладки **Вывод данных** (рис. 16).

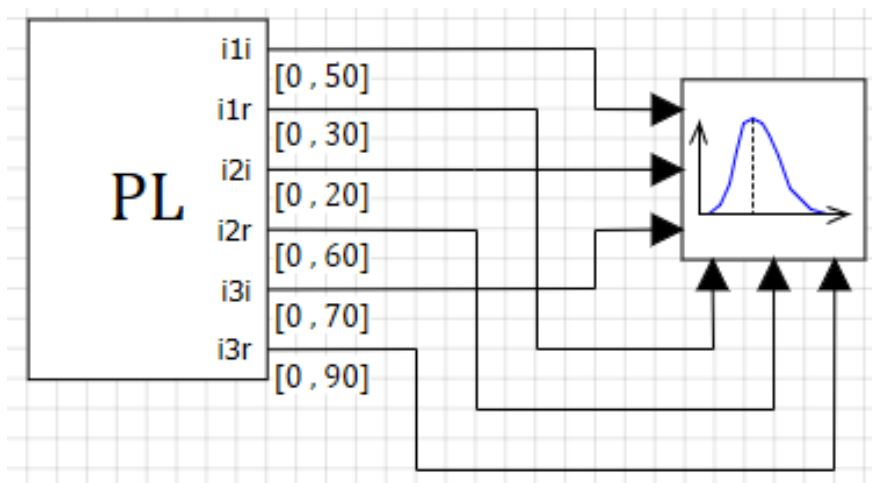


Рис. 16

Записать в диалоговом окне блока **Язык программирования** программу:

```
1  output
-   I1i=[0,50], I1r=[0,30],
-   I2i=[0,20], I2r=[0,60],
-       I3i[2],I3r[2];
-   I3i=I1i+I2i; I3r=I1r+I2r;
```

В программе в качестве выходных переменных блока **Язык программирования** задаются:

- четыре вектора: $I1i$, $I2i$ и $I1r$, $I2r$ размерностью 2, в которых указываются мнимые и вещественные части двух комплексных чисел;
- размерность мнимой и вещественной частей третьего вектора $I3i$ и $I3r$ равна 2.

В основной части программы в векторах $I3i$ и $I3r$ находится векторная сумма соответственно для мнимой и вещественной частей комплексного числа.

Вещественные значения векторов из выходных портов блока **Язык программирования** подаются на **X** входы блока **График Y от X** и формируют значения графика функции по оси абсцисс. Мнимые значения векторов из выходных портов блока **Язык программирования** подаются на **Y** входы блока **График Y от X** и формируют значения графика функции по оси ординат.

На рис. 17 показаны результаты выполнения программы на экране блока **График Y от X** в виде двух векторов $I1$ и $I2$, а третий вектор $I3$ соответствует векторной сумме $I1$ и $I2$. На векторной диаграмме все векторы выходят из нуля.

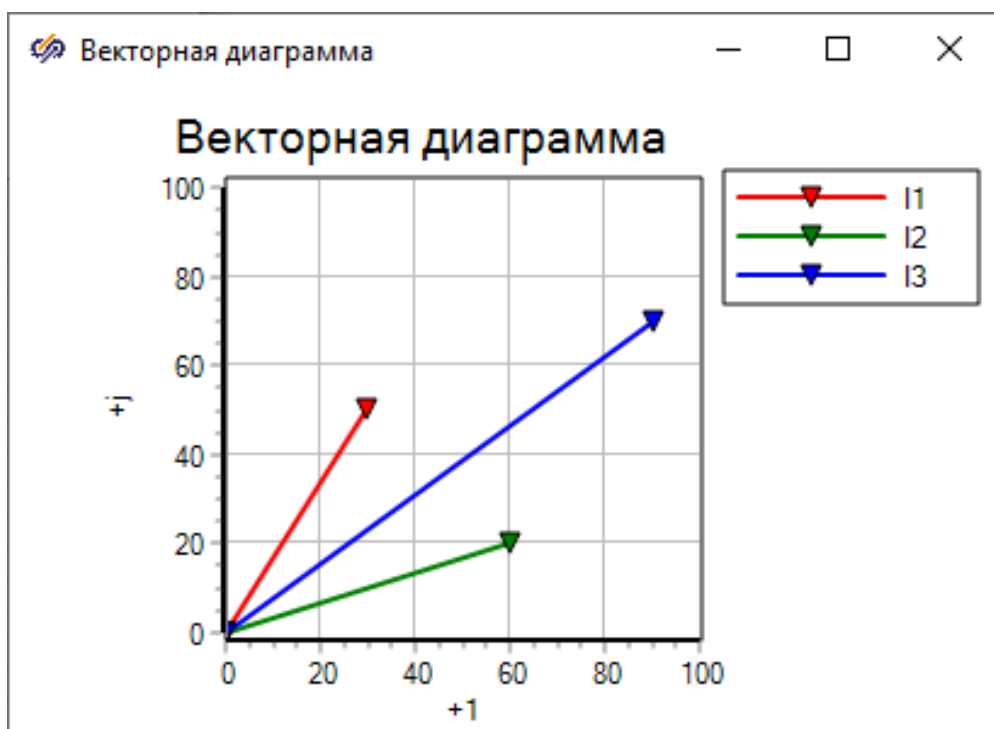


Рис. 17

6. Создать новый проект на основе расчетной схемы решателя
Схема модели общего вида (рис. 18).

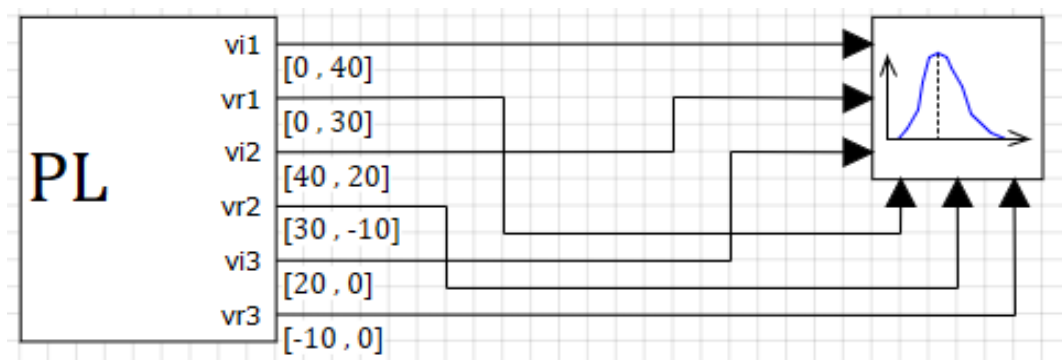


Рис. 18

Для построения топографической диаграммы записать в диалоговом окне блока **Язык программирования** программу:

```
var vr=[0,30,-10,0], vi=[0,40,20,0];
output vi1=[vi[1],vi[2]], vr1=[vr[1],vr[2]],
       vi2=[vi[2],vi[3]], vr2=[vr[2],vr[3]],
       vi3=[vi[3],vi[4]], vr3=[vr[3],vr[4]];
```

На рис. 19 показаны результаты выполнения программы на экране блока **График Y от X** в виде топографической диаграммы, на которой каждый последующий вектор выходит из конца предыдущего вектора.

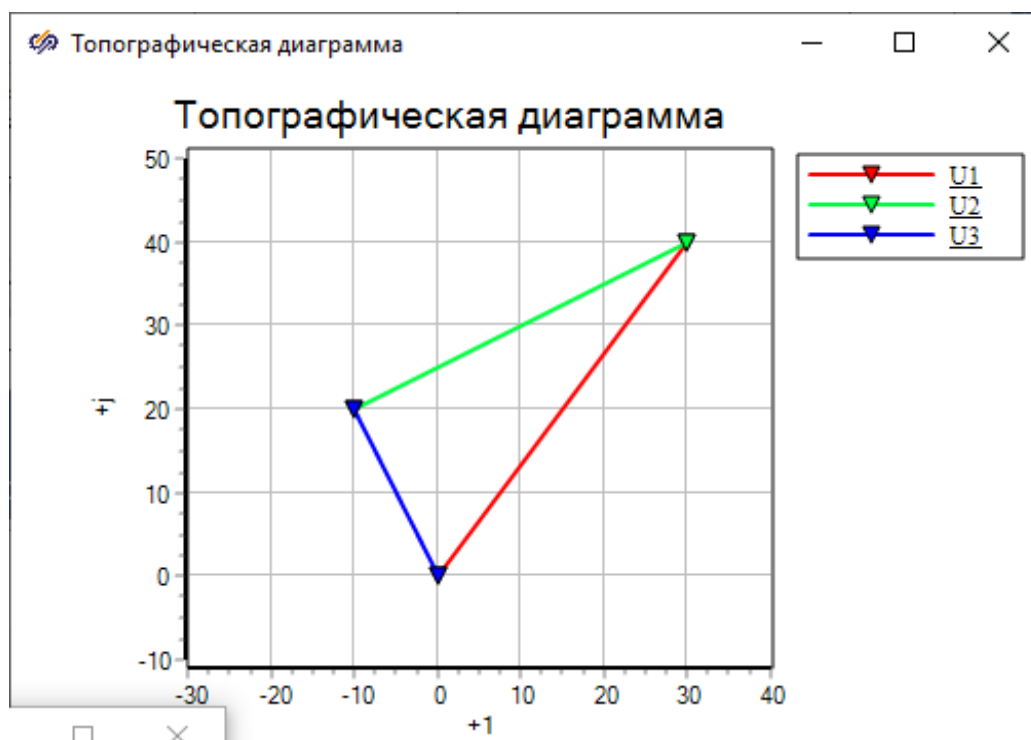
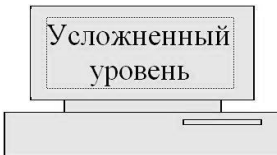


Рис. 19

Задание для отчета по лабораторной работе



В зависимости от номера варианта N построить в декартовой системе координат с помощью блока **График Y от X** :

– векторную диаграмму для комплексных чисел $\underline{I}_1 = N + j(N + 5)$, $\underline{I}_2 = N + jN$; $\underline{I}_3 = \underline{I}_1 + \underline{I}_2$;

– топографическую диаграмму для комплексных чисел $\underline{U}_1 = N + j(N + 5)$, $\underline{U}_2 = (N + 10) + jN$; $\underline{U}_3 = \underline{U}_1 + \underline{U}_2$.

Вставить в отчет по лабораторной работе формулировку заданий, текст полученных программ, а также результаты их выполнения на экране блока **График Y от X** , добавив название графика, легенду и названия осей.

Контрольные вопросы

1. Какие блоки используются в SimInTech для построения одного и нескольких графиков в одних осях?
2. Как задаются различные настройки для графиков в SimInTech?
3. Как добавить к графикам сетку из координатных линий, названия осей, легенду и заголовков?
4. Как увеличить требуемую часть графика?
5. Как построить несколько графиков в одних осях с помощью блока **График Y от X** ?
6. Как построить несколько графиков в одних осях с помощью блока **Временной график**?
7. Для каких целей можно использовать в программе системную переменную **time**?
8. Каким образом можно изменить конечное время расчета?
9. Каким образом можно построить векторную и топографическую диаграммы в SimInTech?

СПИСОК ЛИТЕРАТУРЫ

1. SimInTech.ru [сайт] URL: <http://simintech.ru/>
2. Справочная система SimInTech. URL: <https://help.simintech.ru>
3. Технология SimInTech: моделирование, разработка и создание САУ. URL: <https://miem.hse.ru/data/2018/10/29/1141996123/SimInTech.pdf>
4. Основы моделирования в SimInTech : метод. пособие / сост.: А. И. Ляшенко, Д. П. Вент, Н. В. Маслова ; ФГБОУ ВО Российский химико-технологический университет им. Д. И. Менделеева, Новомосковский институт (филиал). Новомосковск, 2018. 42 с.

Учебное издание

**Регеда Владимир Викторович,
Регеда Ольга Николаевна**

Программирование в SimInTech

Редактор *А. Г. Темникова*
Технический редактор *Р. Б. Бердникова*
Компьютерная верстка *Р. Б. Бердниковой*

Подписано в печать 31.01.2023. Формат 60×84¹/₁₆.
Усл. печ. л. 4,42. Заказ № 12. Тираж 100.

Издательство ПГУ.
440026, г. Пенза, ул. Красная, 40.
Тел.: (8412) 66-60-49, 66-67-77; e-mail: iic@pnzgu.ru

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

В. В. Рееда, О. Н. Рееда

ПРОГРАММИРОВАНИЕ В SimInTech

**Методические указания
к лабораторным работам**

ПЕНЗА 2023

