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

В. В. Регеда, О. Н. Регеда

Программирование в SimInTech

Учебно-методическое пособие

V. V. Regeda, O. N. Regeda

Programming in SimInTech

Teaching manual

ПЕНЗА 2025

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

В. В. Регеда, О. Н. Регеда

Программирование в SimInTech

Учебно-методическое пособие

V. V. Regeda, O. N. Regeda

Programming in SimInTech

Teaching manual

Пенза
Издательство ПГУ
2025

УДК 001.8 (083.95)
Р32

Р е ц е н з е н т

кандидат технических наук, доцент,
проректор по учебной работе Пензенского
государственного технологического университета

О. А. Голышевский

Регеда, Владимир Владимирович.

Р32 Программирование в SimInTech : учеб.-метод. пособие / В. В. Регеда, О. Н. Регеда = Programming in SimInTech : teaching manual / V. V. Regeda, O. N. Regeda. – Пенза : Изд-во ПГУ, 2025. – 76 с.

Рассмотрены основы программирования на языке SimInTech, встроенном в систему модельно-ориентированного проектирования SimInTech, с помощью которой можно написать программу (скрипт) для выполнения определенных действий с объектами схемы в процессе расчета при наступлении определенного события, а также для задания глобальных или локальных констант и переменных для текущей страницы проекта. Приведены методические указания, содержащие необходимый теоретический материал по соответствующей теме и перечень контрольных вопросов.

Издание подготовлено на кафедре электроэнергетики и электротехники Пензенского государственного университета и предназначено для лабораторных и индивидуальных занятий по курсу «Компьютерные технологии в электроэнергетике» для обучающихся англоязычных вузов по направлению подготовки 13.03.02 «Энергетика и электротехника».

The basics of programming in the SimInTech language embedded in the SimInTech model-oriented design system are considered, with which you can write a program (script) to perform certain actions with schema objects during the calculation process upon the occurrence of a certain event, as well as to set global or local constants and variables for the current project page. The teaching manual is provided with methodological guidelines containing the necessary theoretical material on the relevant topic and a list of control questions.

The publication was prepared at the Department of Electric Power Engineering and Electrical Engineering of Penza State University and is intended for laboratory and individual classes in the course «Computer technologies in electrical power engineering» for English-speaking university students majoring in 13.03.02 "Power and Electrical Engineering".

УДК 001.8 (083.95)

Contents

Introduction

Currently, various systems such as Mathcad, MATLAB, etc. are used in engineering practice to automate mathematical calculations and modeling. Among them, we can single out the Russian model-oriented design system SimInTech. This system is an environment that allows you to create mathematical models, control algorithms, control interfaces and automatic code generation for programmable controllers and graphical displays [1].

SimInTech has extensive computing capabilities related to solving algebraic and ordinary differential equations. Computational mathematical models in SimInTech are created through functional block programming using blocks that are contained in various libraries [2].

SimInTech can use the built-in specialized high-level programming language SimInTech for functional and dynamic modeling of various systems. It allows you to write a program (script) to perform certain actions with schema objects during the calculation process when an event occurs, as well as to set local or global variables and constants of the project. The SimInTech language describes the functioning of a typical Programming Language block from the Dynamic palette.

The programming language contains built-in keywords, constants, declarations, operators, and functions/procedures. The main distinguishing feature of the SimInTech programming language is that the program text is designed to be executed at each computational step during circuit modeling [2].

In terms of calculation speed of complex or mathematically rigid models, SimInTech surpasses foreign modeling programs by 20 %. The set of methods for solving differential equations contains both classical methods used in competing software products and proprietary methods that provide advantages in calculating complex systems [3].

Laboratory work № 1
Getting to know the SimInTech environment and the SimInTech programming language

The purpose of the work: introduction to the interface of the SimInTech system and general information on using the **Programming Language** block for writing programs in the SimInTech programming language.

Work assignment

1. Launch the SimInTech program. As a result, the main window with the SimInTech interface appears on the screen, shown in Figure 1 [2].

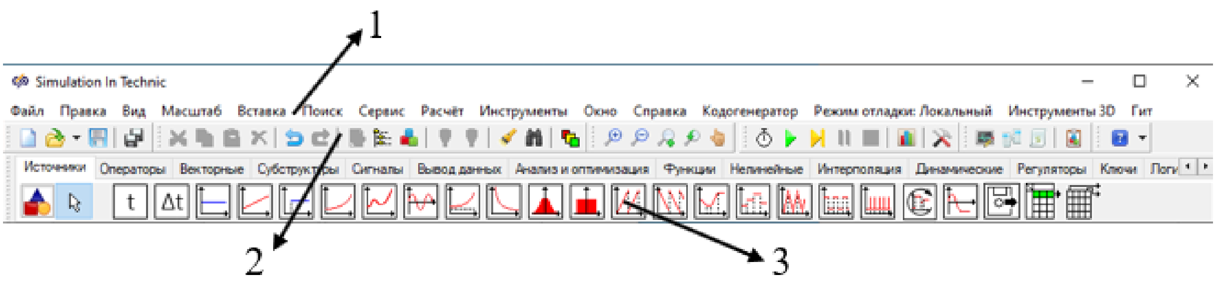


Figure 1

The first line of the main window contains the **Menu area of the main window** (1), from which, in particular, you can control the creation, editing and modeling modes of created models, which are called calculation schemes in SimInTech.

The second line of the window contains the **Button Panel area** (2), with grouped sets of buttons that invoke the most frequently used commands.

The third line of the window contains the **Block Palette area** (3), where the blocks corresponding to the selected calculation scheme appear on the tabs.

2. To solve a specific engineering problem, an appropriate calculation scheme with a solver and its own settings should be selected.

Select the File command in the first menu bar of the main window **File → New project → Control system diagram** (Figure 2).

As a result, a project window opens (Figure 3), in the header of which the file name is indicated, the **Control system diagram** and the default file extension is **prt**.

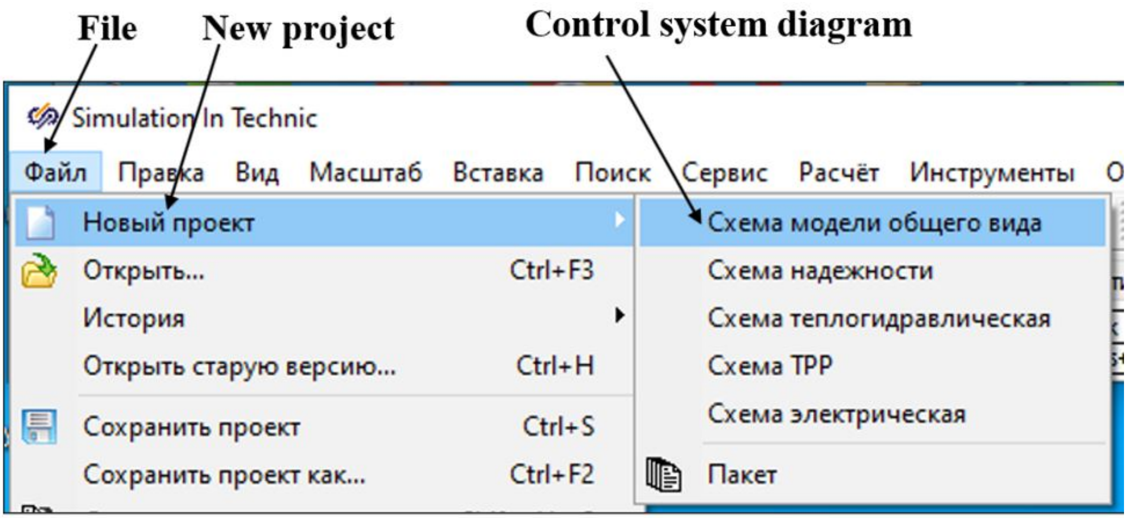


Figure 2

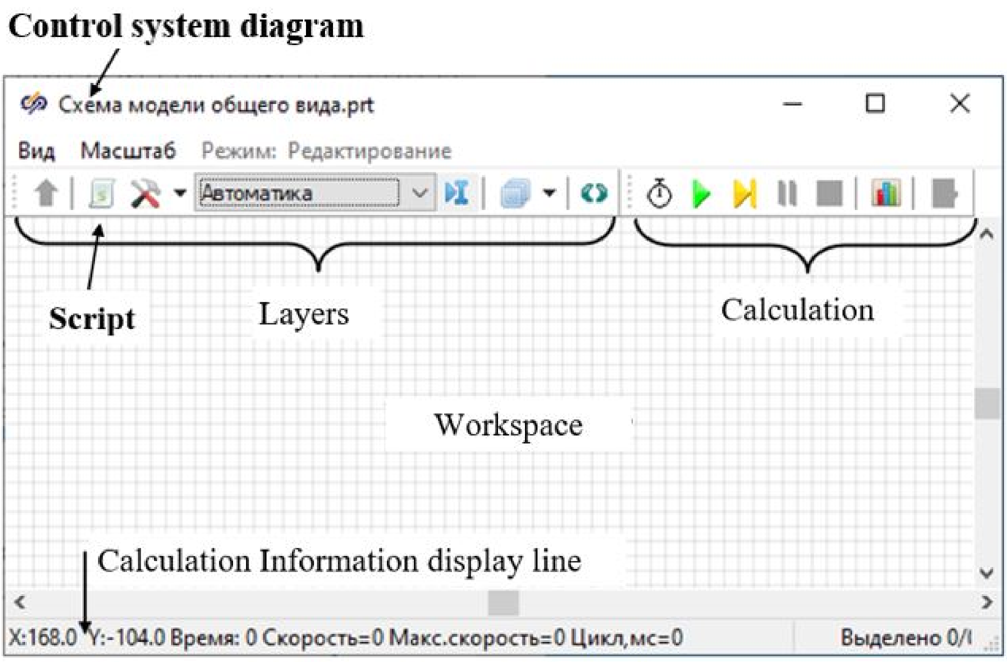


Figure 3

The button panel at the top of the project window contains tools for configuring the «Layers» project and managing the «Calculation». At the bottom of the project window is the «Calculation Information display line». In the center of the project window there is a «Workspace» in which, using blocks selected from the block palette, you can create and edit previously created calculation models, create your own blocks, etc.

To perform the specified actions with the model blocks, as well as to set local or global variables and constants of the project, you can write a program (script) in the SimInTech language. To do this, click the Script button  from the Layers section on the project editing window panel.

As a result, the **Page Script** window opens (Figure 4), containing a button panel, a script text input field, and a communication panel at the bottom of the window.

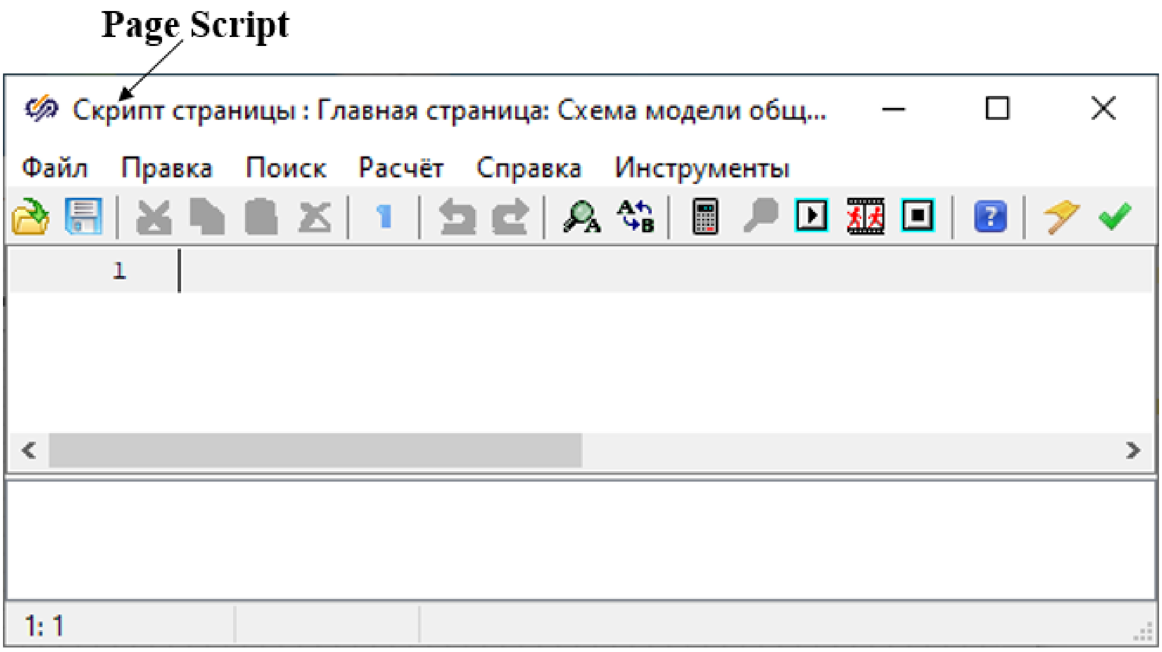


Figure 4

Type the script text shown in the Figure 5 in the input field. Click View Script Variables (1), or View the value of a pre-selected variable (2), or Run Script (3).

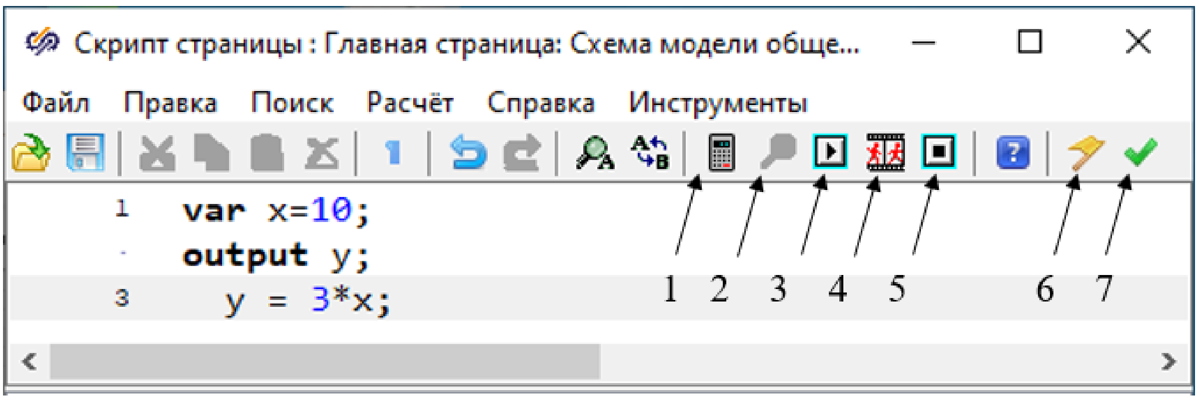


Figure 5

In this case, the **View Variable Values** (**Просмотр значений переменных**) window opens to the right of the program script (Figure 6).

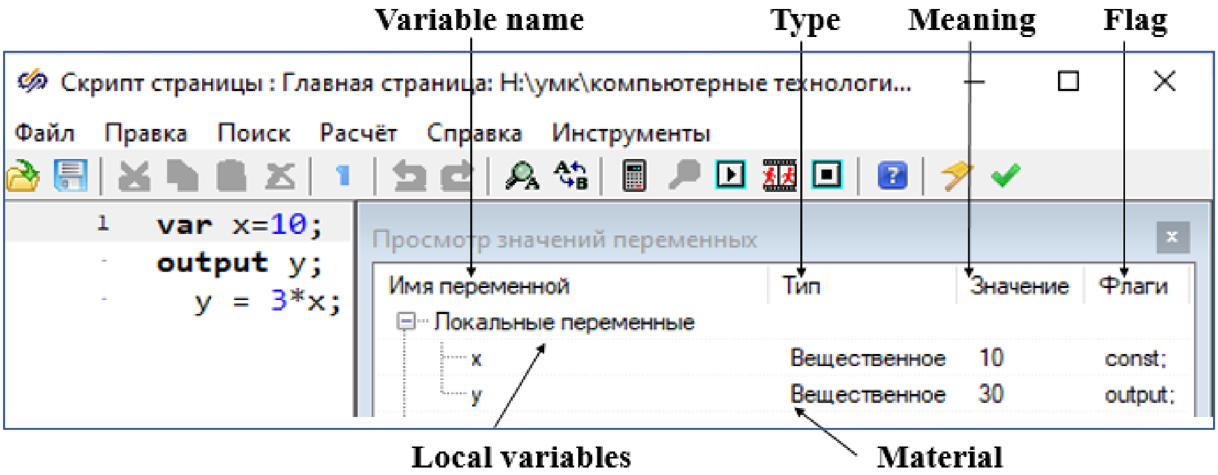


Figure 6

You can run a step-by-step debugging of the script by pressing button 4, and finish debugging it by pressing button 5 (Figure 5). After making changes to the program, click the Apply Changes button 6 or the Apply Changes button and close the editing window 7 (Figure 5).

3. A program in the SimInTech programming language can be linked to a specific block of the schematic project window by selecting the left mouse button (LMB) in the **Block Palette Area** on the tab **Dynamic (Динамические)** block **Programming Language** (**Язык программирования**). Then move the cursor to the desired location of the project window, and click the LMB to specify the place where the block is inserted (Figure 7).

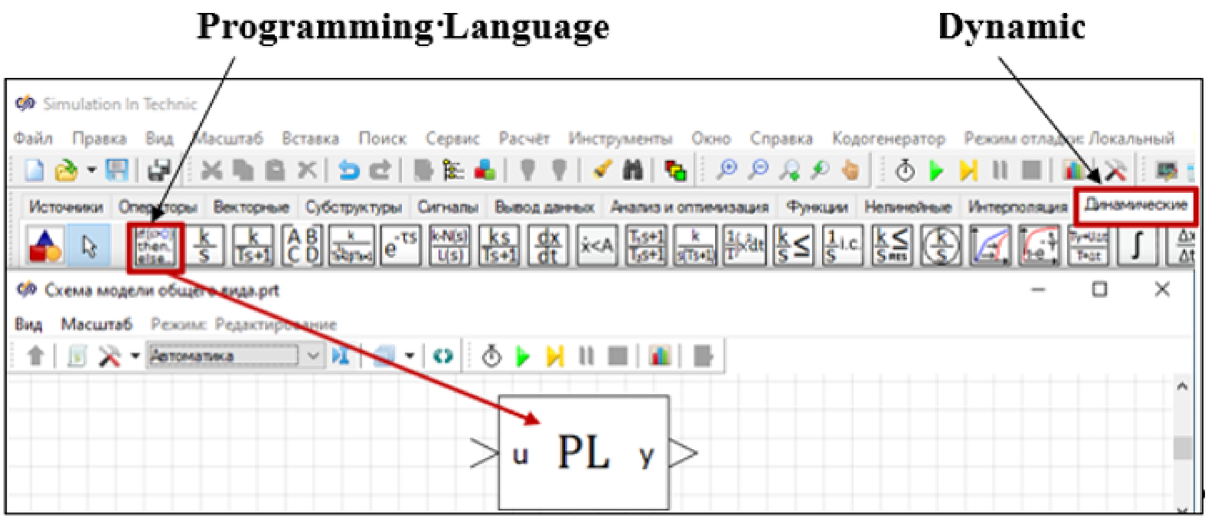


Figure 7

The **Programming Language** block is designed to create blocks that perform complex operations, including operations on matrices, vectors, and complex numbers. Any program in the SimInTech programming language can be written inside the dialog box of the **Programming Language** block, which is a window of the text editor of the algorithm.

Double-clicking on the **Programming Language** block opens the **Programming Language Block** dialog box with the default program code (see Figure 8), which defines the type of block in the projects window.

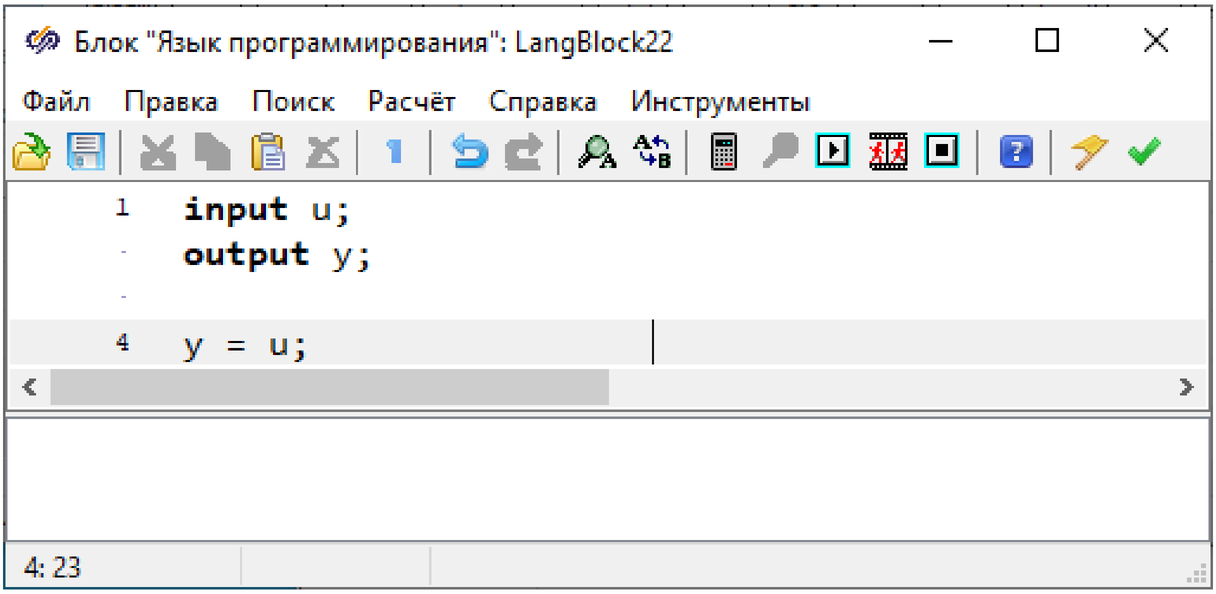


Figure 8

In the program shown in Figure 8, the **input** and **output** keywords are used to set the input u and output y variables of the block, respectively.

As a result, the input and output ports of the **Programming Language** block are formed (Figure 7). The signal received at the input of the unit is transmitted unchanged to its output.

Save the program text as a text file in a folder as directed by the teacher under the named **lw_1_1** with the **TXT** extension. To do this, select the **Save** → **File** command in the **Programming Language** dialog box.

Save the project as a file in a folder as directed by the teacher under the name **lw_1_1** with the default extension **prt**. To do this, select the **File → Save Project As...** command in the project’s schematic window.

4. If necessary, you can get general information about the SimInTech programming language directly from the program by selecting the command **Help** → **Contents** → **Programming Language** in the main menu of the main window or by pressing the **F1** button [2].

A program (script) in the SimInTech programming language consists of *declarations* and *operators* separated by a semicolon «**;**».

The program can contain comments, either enclosed in curly brackets { }, or starting with the characters «//» and ending at the end of the line.

Input, output, and dynamic variables or constants can be used in a program only after they are described (initiated) in a declaration or assigned a value using the assignment operator. The values of the other variables used in the program can be calculated according to the expression that is assigned to the variable. Declarations are recommended to be placed at the beginning of the program.

SimInTech programming language *identifiers* (names of constants, variables, labels, functions, and procedures) can contain letters of the Latin and Russian alphabets, underscores, and digits. The identifier can have any length and must start with a letter or an underscore. Uppercase and lowercase letters in identifiers are not distinguished by default. It is not allowed to use keywords as identifiers (these words are automatically highlighted in bold when entered).

Declarations should start with *keywords*:

const – constants;

var – algebraic variables;

init – dynamic (differential) variables;

input – the block's input variables;

output – the output variables of the block.

After the keyword, the names of variables or constants are listed, separated by commas, to which the initial value is set using the assignment operator =. If the initial value of variables or constants is zero, only their name is indicated.

Constants differ from variables in that they are specified only once in a declaration. If the constant is redefined later, it will be treated as a variable.

Open the **Programming Language Block** dialog box and enter the following program code in the text editor window that opens:

```
const e=4;//initialization of a real-type constant
var i=[], //initializing the zero-length array i
    r=[1,2,3,4]; //initializing the r array
output e,r,i; //initialization of the output variable e,r,i
          i = e/(r);   //the main section of the code
```

In the program, the constant **e=4** is initiated after the **const** keyword.

Then, after the **var** keyword, two variables are initiated separated by commas: a one-dimensional array **i** of variable dimension and a one-dimensional array **r** of length 4, in which four values of its elements are specified.

Next, the program declares output variables after the **output** keyword, the values of which will be output to the corresponding ports of the **Programming Language** block.

In the main section of the code, the elements of the one-dimensional array **i** are calculated.

5. Go back to the project window, select the **Programming Language** block, right-click and select **Object Properties** from the context menu that appears.

In the window that appears (Figure 9), open the **General** tab and click on the icon  in the **Graphic Image** line.

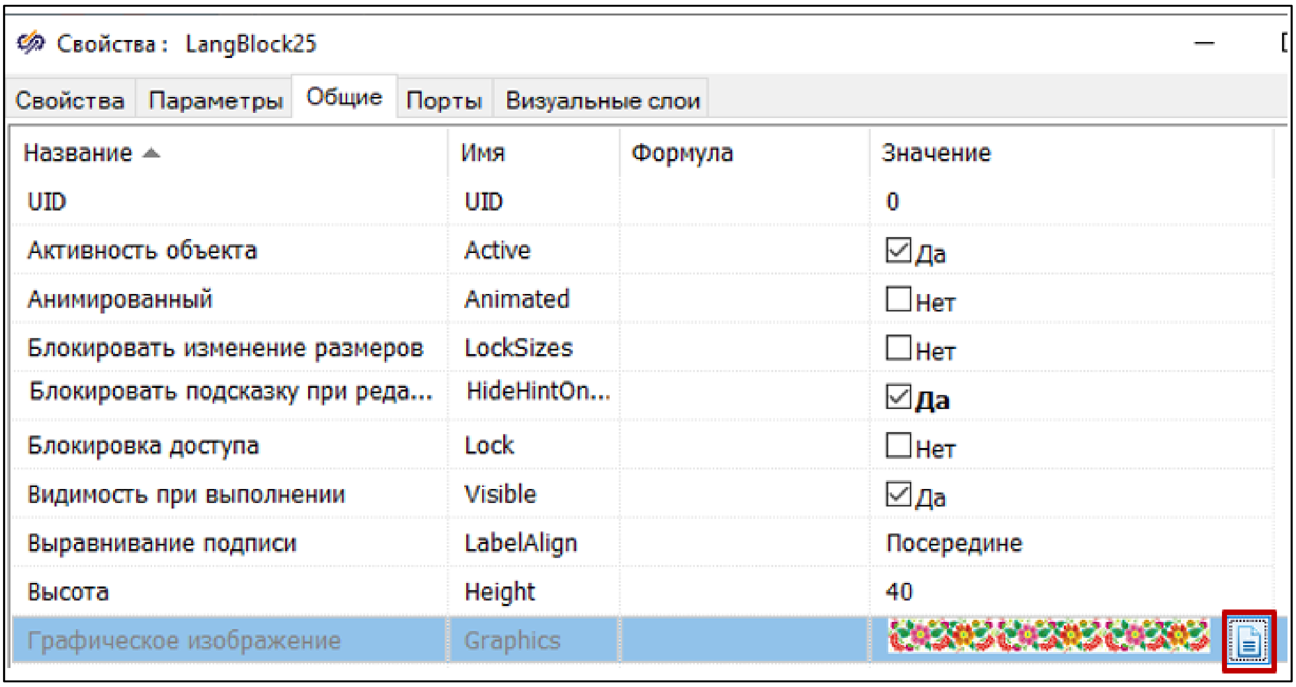


Figure 9

As a result, the **Graphical Editor** window shown in Figure 10 opens.

Double-click the LMB on the text and use the keyboard to edit the text in the **Text Editing** (**Графический редактор**) window that appears in accordance with Figure 11.

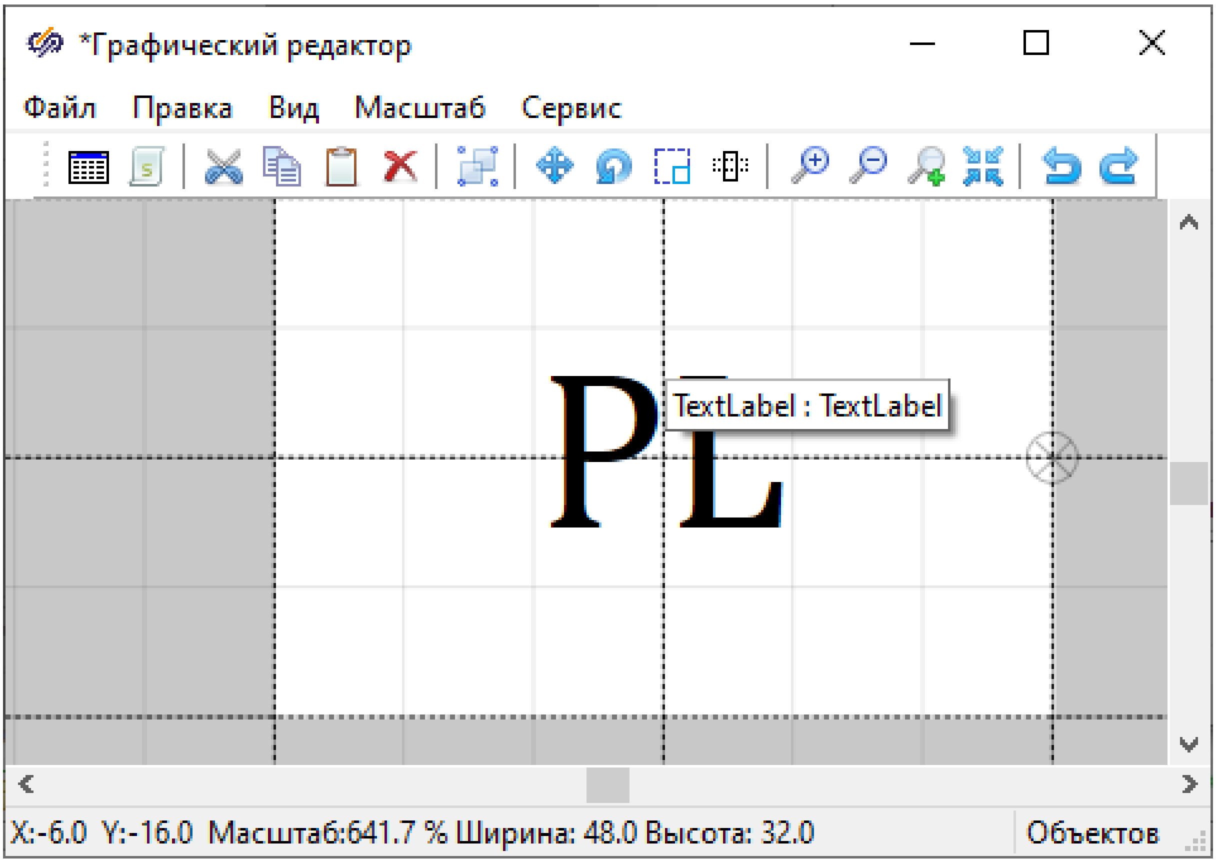


Figure 10

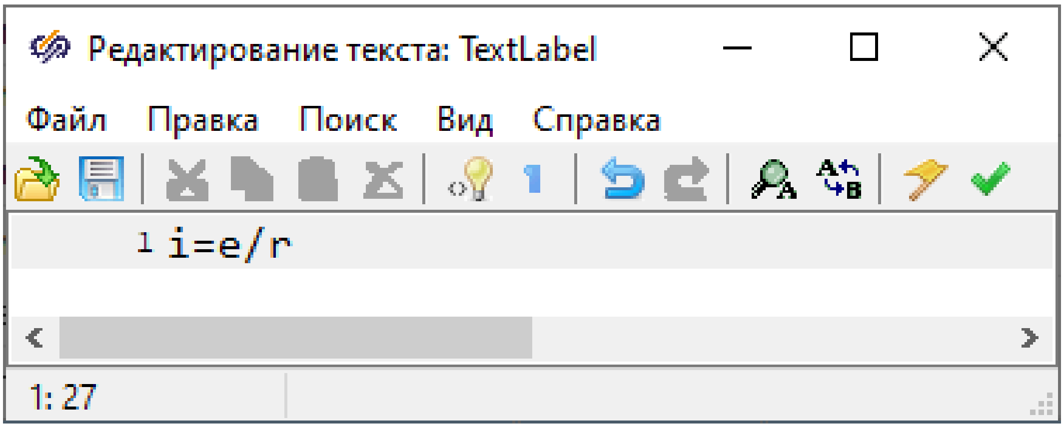
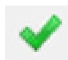


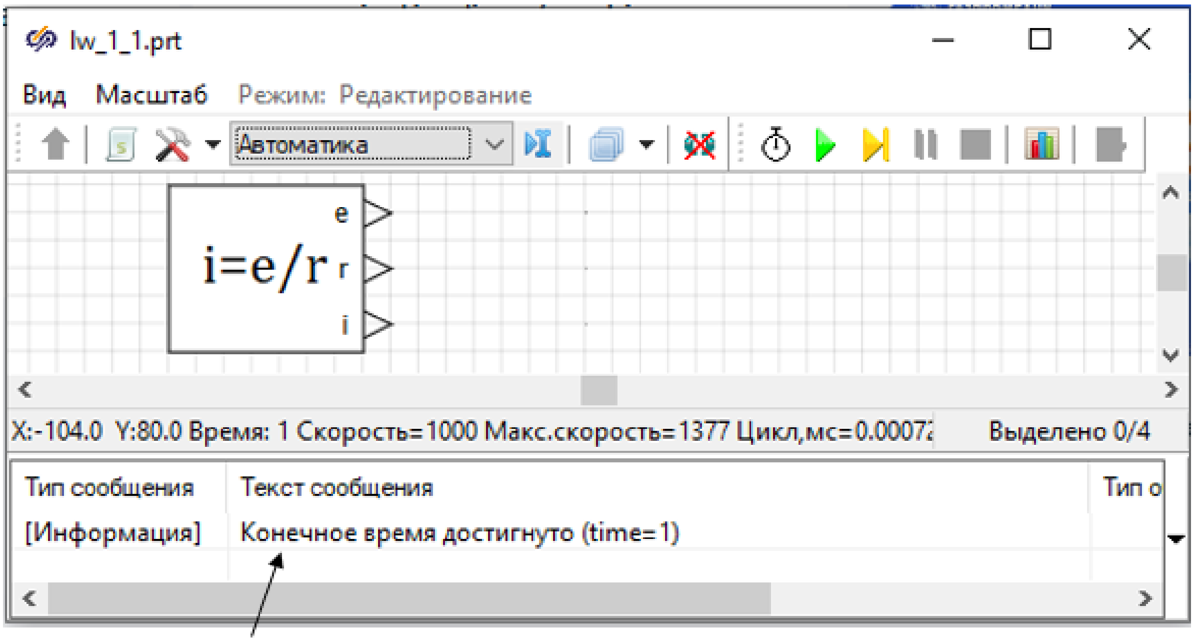
Figure 11

To save the changes and close the text editing window, click . Close the **Graphics Editor** window and confirm the request to save the program text.

6. Go back to the projects window and run the program by clicking on the Start button  on the Toolbar of the projects window or on the **F9** key.

Confirm the request to save the program text. After the program is completed, the message **The end time has been reached (time=1)** will appear in the information display line.

Figure 12 shows how the appearance of the **Programming Language Block** in the project window has changed compared to the appearance of this block in Figure 7.



The end time has been reached (time=1s)

Figure 12

In Figure 12, in addition to the fact that the text inside the block has changed, it has three output ports to which information about the values of the output variables of the **e**, **r**, **i** block can be output.

To display the results of the program in the project window, you must move the cursor to each output port of the block in turn, wait for the cursor to turn into an arrow of the type ↑ and make a 1-fold click on the LMB.

As a result, a communication line will appear at the port output, which you need to extend to the right of the port, and then make a 1-fold right-click at the end of the communication line.

If you restart the program for execution and press the **Show values on communication lines** key in the projects window, the corresponding values of the output variables of the **e**, **r**, **i** block will be displayed under the communication lines (Figure 13).

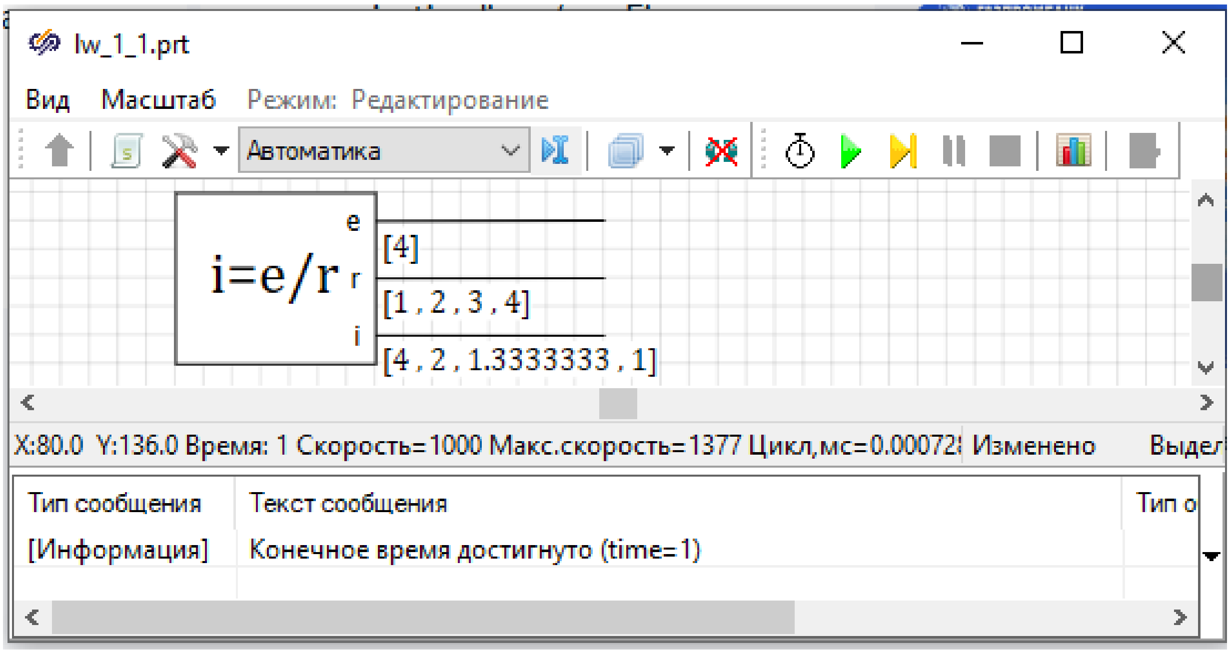
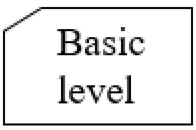


Figure 13

7. Open the text window of the **Programming Language** block editor and run the program again.

As a result, a window **Viewing variable values** (Figure 14) will appear to the right of the program text, in which the names of variables, their types, values, and flags will be displayed as a table.

Assignment for the lab report



Edit the program code from clause 4 in the dialog box of the **Programming Language** block by setting the value of the constant e equal to the parameter number N as directed by the teacher.

Insert the text of the received program into the laboratory report, as well as the results of its execution in the schematic window of the project and in the dialog box of the **Programming Language** block.

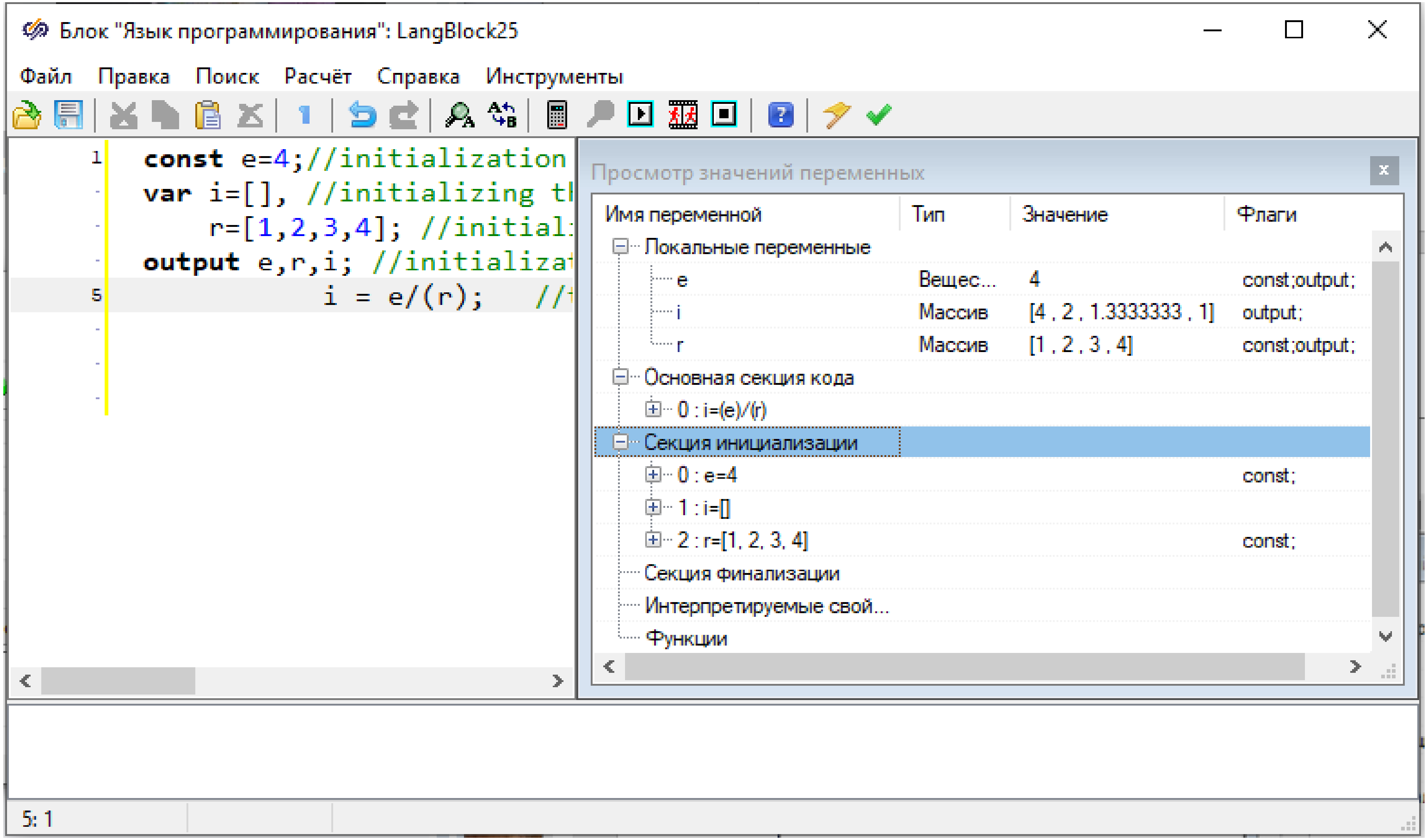

Блок "Язык программирования": LangBlock25
Файл Правка Поиск Расчёт Справка Инструменты
const e=4;//initialization
var i=[], //initializing th
r=[1,2,3,4]; //initiali
output e,r,i; //initializat
i = e/(r); //
Просмотр значений переменных
Имя переменной
Тип
Значение
Флаги
Локальные переменные
e
Вещес...
4
const;output;
i
Массив
[4 , 2 , 1.3333333 , 1]
output;
r
Массив
[1 , 2 , 3 , 4]
const;output;
Основная секция кода
0 : i=(e)/(r)
Секция инициализации
0 : e=4
const;
1 : i=[]
2 : r=[1, 2, 3, 4]
const;
Секция финализации
Интерпретируемые свой...
Функции
5: 1

Figure 14

8. The basic concept of most programming languages is a mathematical expression based on *numbers*, *constants*, *variables*, *operators*, *functions*, *and various special characters*.

Constants in the SimInTech programming language are divided into numeric and string (symbolic) constants. String constants are any set of characters in quotation marks, a «**string of characters**».

Further in the description of command formats, the symbols **<...>** denote mandatory identifiers, and the symbols **{...}** – optional additional identifiers [2]. The format for setting numeric constants:

{<sign>}<integer part>{<decimal part>} {E<exponent>}{i}.

Numerical constants in the SimInTech programming language, similar to other high-level programming languages, can be integers, real (fractional with fixed and floating point), and complex.

A dot is used as a decimal separator in numbers, for example **10.5**.

The large and small real numbers **N** are output in floating-point format, which is represented as follows:

$$\mathbf{N = M \cdot 10^{p}},$$

where **M** – is the mantissa of the number (in the normalized form, the mantissa of decimal numbers takes a value from one to nine); **p** – is the order of the number, taking into account the sign.

In the floating-point format, the mantissa of a number is separated from its order by the character **E** without spaces. At the same time, the sign is not indicated for positive orders, and the «–» symbol is indicated for negative orders. For example, the number **0,0000123** in floating-point format is written as **1.23E-5**.

In SimInTech, the symbols **i**, **j**, **I**, **J** immediately after the number mean that the number is imaginary, for example, the imaginary unit is denoted by **1**i. For example, if you enter the number **5i**, SimInTech accepts it as a complex number **0+5**i, with the real part being 0 and the imaginary part being 5.

Constants, like variables, can be vector and matrix. A string vector is set in square brackets as a set of constants separated by commas, for example, a vector of length 4 – **[1,2,3,4]**. A matrix is defined as a set of vector constants (strings) enclosed in square brackets and separated by a semicolon, for example, a matrix of size 2×3 – **[[1,2,3];[4,5,6]]**.

Named constants are specified using the const keyword, followed by comma-separated names of constants with their assigned values, for example **const a=10.5, p=[1,2,3,4]**.

SimInTech programs often use the following built-in numeric constants:

– **pi** = 3.14159**…** – number π;

– **expbase** =2.7182818284590452353602874713527 – number e.

9. In SimInTech, you can use *variables*, which are named objects designed to store data in PC memory. Depending on this data, variables can be numeric or symbolic, vector or matrix.

To assign values to variables, an assignment operation is used, which has the following format:

<Variable name>=<expression> {;}

The types of variables are determined by the expression whose value is assigned to the variable. The expression to the right of the assignment sign can be just a number, a vector, a matrix, an arithmetic expression, a string of characters in quotation marks, or a symbolic expression.

The format for specifying the type and value of algebraic variables:

var <variable name>{:<variable type> {=<initiating expression>}} {;}

Let's look at the most commonly used types of variables.

The **integer** type is a 32-bit integer. In the program below, three variables are assigned the value 31 in decimal, hexadecimal (prefix **0x**) and binary (prefix **0b**) representations:

```
var dec_x:integer=31,//in decimal representation
hex_x:integer=0x1F,//in hexadecimal representation
binary_x:integer=0b11111;//in binary representation
```

The **double** type is a 64–bit double-precision integer of the IEEE-754 standard with a floating point and a sign. In the program below, the **real_x** variable is assigned a value equal to -0.0001005 in floating point format:

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

The **Complex** type is a complex 128-bit number. It has the output format **a±bi**, where **a** is the real part, **b** is the imaginary part. You can also specify a complex number in the form **(a,b):**

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

Type **Array** is an array (vector string) of real numbers. In the program below, the variable **p** with dimension 4 is assigned values to the array elements equal to 1,2,3,4:

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

Type **boolean** is a binary 1-bit value. In the program below, the **bool_value** variable is assigned the value **0** or can be assigned the value **1**:

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

Type **string** – a string of characters. In the program above, the **str_value** variable is assigned the value of the string of characters **"Laboratory work No. 1"**:

```
var str_value:string="Laboratory work No.1";
```

In SimInTech programs, the **time** system variable is available, which is equal to the model time.

Assignment for the lab report

Complicated level

Independently consider other types of variables used in SimInTech.

Report

Include a description of the types of variables considered and examples of their use in the laboratory report.

10. Computing mathematical expressions containing operators and functions is the main goal of any system designed for numerical calculations.

An *operator* is a special notation for a specific operation on data (*operands*).

Functions are objects with unique names that perform certain transformations of their arguments and at the same time necessarily *return* the results to the place of the call.

If the operations have equal priority, they will be performed sequentially, from left to right, but first the expressions enclosed in parentheses «( )» are evaluated, after which they are treated as operands. Operations with a higher priority will be performed first, regardless of their location.

Table 1 shows the designations of SimInTech arithmetic, integer, and logical operators [1], their purpose, and application examples.

Table 1

Designaion	**Appointment**	**Application example**
+	Addition of expression 1 (15) and expression 2 (10)	$15+10=25$
	Piecemeal addition of two vectors	$[3,4]+[2,2]=[5,6]$
	Piecemeal addition of two and matrice	$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}+\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}=\begin{bmatrix}6 & 8\\10 & 12\end{bmatrix}$
	For strings, addition corresponds to concatenation	“Sim”+”InTech”=“SimInTech”
–	Subtraction from an expression 1 (15) expressions 2 (10)	$15-10=5$
	Piecemeal subtraction of two vectors	$[3,4]-[2,2]=[1,2]$
	Piecemeal subtraction of two and matrices	$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}-\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}=\begin{bmatrix}-4 & -4\\-4 & -4\end{bmatrix}$
*	Multiplier Multiplications 1 (15) for a multiplier of 2 (10)	$15*10=150$
	The vectors are multiplied element by element	$[3,4]*[2,2]=[6,8]$
	Matrix multiplication	$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}*\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}=\begin{bmatrix}19 & 22\\43 & 50\end{bmatrix}$
/	Dividing the divisible (15) by the divisor (10)	$15/10$
	Element-wise division of vectors	$[3,4]/[2,2]=[1.5,2]$
	For matrices, multiplication by the inverse matrix is performed	$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}*\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}^{-1}=\begin{bmatrix}3 & -2\\2 & -1\end{bmatrix}$
.*	Element-wise matrix multiplication	$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}.*\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}=\begin{bmatrix}5 & 12\\21 & 32\end{bmatrix}$
./	Element-wise division of matrices	$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}./\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}=\begin{bmatrix}19 & 22\\43 & 50\end{bmatrix}$

Designaion	Appointment	Application example
^	Raising the first expression to the power of the second. Vectors are processed piecemeal	$5\,\hat{}\,2 = 25$
.^	Exponentiation of a matrix element-wise	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .\hat{}\, \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 64 \\ 2187 & 65536 \end{bmatrix}$
!	Calculating the factorial of an integer	$5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$
..	Calculating the interval of integers from a to b	$1..5$ denotes $[1,2,3,4,5]$
or	Bitwise logical operation OR	$x1 = \text{true or } x2 = \text{true}$
and	Bitwise logical operation AND	$x1 = \text{true and } x2 = \text{true}$
div	Integer division	$5 \text{ div } 2 \to \text{returns } 2$
mod	The remainder of the integer division	$5 \text{ mod } 2 \to 1$
>	The operation IS BIGGER. The operation returns one if operand 1 is greater than operand 2	$5 > 3 \to 1$ $5 > 10 \to 0$
<	The operation is SMALLER. The operation returns one if operand 1 is less than operand 2	$5 < 15 \to 1$ $15 < 5 \to 0$
>=	The operation IS GREATER THAN OR EQUAL TO. The operation returns one if operand 1 is greater than or equal to operand 2	$5 >= 5 \to 1$ $15 >= 5 \to 1$ $3 >= 5 \to 0$
<=	The operation IS LESS THAN OR EQUAL TO. The operation returns one if operand 1 is less than or equal to operand 2	$5 <= 15 \to 1$ $5 <= 5 \to 1$ $5 <= 3 \to 0$
=	The operation is EQUAL. The operation returns one if operand 1 is equal to operand 2	$3 + 2 = 5 \to 1$ $3 + 2 = 6 \to 0$
<>	The operation IS NOT EQUAL. The operation returns one if operand 1 is not equal to operand 2	$3 + 2 <> 6 \to 1$ $3 + 2 <> 5 \to 0$

Assignment for the lab report

Basic level

To create a new project based on the calculation scheme of the solver **Control system diagram**. Write a program in the **Script window** of the page to calculate the value of X in accordance with the specified numeric expression for your option number N:

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

Report

Insert the assignment statement into the laboratory report, the text of the received program in the **Script page** window, as well as the results of its execution (the value of X) in the **View Variable Valu**es window.

11. SimInTech programs can use *functions* or *procedures* that, depending on the parameter values, return different results to the place where they are called.

The format for defining functions (procedures):

“name of the function or procedure" {(“parameter 1”{,“parameter 2”}) }

A comma, semicolon, or space can be used to separate function parameters. The function may have no parameters.

When referring to a function, the value of its parameter (*argument*) is indicated in parentheses.

Table 2 shows the designations of some standard SimInTech functions [2], their purpose and application examples.

Table 2

Designaion	**Appointment**	**Application example**
abs	The function of obtaining the modulus of a real or complex number or vector	abs(-5) → (returns) 5 abs(3+4i) → 5 abs([-3,-4]) →[3, 4]
exp	The function of calculating the exponent of a real or complex number	exp(2) → 7.389051
lg	The function of calculating the exponent of a real or complex number	lg(100) → 2
ln	The function of calculating the natural logarithm of a real or complex number	ln(5) → 1.6094379

Designaion	Appointment	Application example
max	The function of calculating the maximum value from two values or from the values of a vector	max(1,2) $\rightarrow$ 2 max([1,2,3,4,5]) $\rightarrow$ 5
min	The function of calculating the minimum value from two values or from the values of a vector	min(1,2) $\rightarrow$ 1 min([1,2,3,4,5]) $\rightarrow$ 1
root	The function of calculating the root of an arbitrary power of a real or complex number	root(2,9) $\rightarrow$ 3, where 2 is the degree of the root; 9 is the input value.
sign	The function determines the sign of the number	sign(-50) $\rightarrow$ -1 (the sign of a negative number) sign(50) $\rightarrow$ 1 (the sign of a positive number)
round	The function rounds a real number to an integer	round(2.4) $\rightarrow$ 2
sqrt	The function of calculating the root of a square real or complex number	sqrt(25) $\rightarrow$ 5
trunc	The function separates the integer part of a real number from the fractional part	trunc(13.7) $\rightarrow$ 13

rand	The function of generating uniform noise in the range of values from 0 to 1	var t:array = [0,0];
		t[1]=rand; t[2]= rand;
		The values of the elements of the array t will be filled with random numbers.

Table 3 shows the designations of the SimInTech trigonometric functions [1], their purpose and application examples.

Note that the arguments of the trigonometric functions and the angles returned by the inverse trigonometric functions in SimInTech are set in radians.

Table 3

Designaion	**Appointment**	**Application example**
arccos	The function of calculating the arccosine of a real or complex number. The value of the real argument x must be limited by the condition $-1 \leq x \leq 1$	arccos(0.5) → 1.0471976
arcsin	The function of calculating the arcsin of a real or complex number. The value of the real argument x must be limited by the condition $-1 \leq x \leq 1$	arcsin(0.5) → 0.52359878
arctg	The function of calculating the arctangent of a real or complex number	arctg(1) → 0.78539816
atan2	The function of calculating the arctangent of a point, taking into account the quadrant. Format: atan2(y,x), where y is the ordinate coordinate (a real number or an array of real numbers); x – coordinate along the abs-cis axis (real number or array of real numbers)	atan2(1,1) → 0.78539816 atan2(1,-1) → 2.3561945 atan2(-1,-1) → -2.3561945 atan2(-1,1) → -0.78539816
sin	The function of calculating the sine of a real or complex number	sin(pi/6) → 0.5
cos	The function of calculating the co-sine of a real or complex number	cos(pi/3) → 0.5
tg	The function of calculating the tangent of a real or complex number	tg(pi/4) → 1
ctg	The function of calculating the cotangent of a real or complex number	ctg(pi/4) → 1

Table 4 shows the designations of SimInTech functions for working with complex numbers [1] using the example of the complex number $\underline{Z}$=3+i4.

Table 4

Designaion	Appointment	Application example
arg	A function for obtaining the argument of a complex number z, which returns the angle (in radians) formed on the complex plane by a vector defined by a complex number with a real axis	arg(3+4i) → 0.92729522
complex	The function of obtaining a complex number	complex(3,4) → 3+4i
conj	The function of obtaining a complex conjugate number	conj(z) → 3 - 4i
imag	The function of obtaining the imaginary part of a complex number	imag(z) → 4
real	The function of obtaining the real part of a complex numbera	real(z) → 3
sgr	The function of obtaining the complex square root of a real number	sgr(25) → 5+0i

Assignment for the lab report

Basic level

Write a program in SimInTech that sets a complex number $\underline{Z} = N + i\cdot(N + 5)$ for its variant number N.

Using SimInTech functions, calculate the real and imaginary parts of a given complex number, as well as its modulus and argument

Report

Insert the assignment statement into the laboratory report, the text of the received program in the **Script page** window, as well as the results of its execution in the **View Variable Values** window.

Basic level

Data $x = 1{,}5$; $y = 2$; $z = 3$. Calculate a and b from Table 5 for your task option in **double** format

Report

Insert the assignment statement into the laboratory report, the text of the received program in the **Script page** window, as well as the results of its execution (values a and b) in the **View Variable Values** window.

Table 5

Option number	a	b
1	$a = \dfrac{z + y/(x^2+4)}{e^{-x}/x^2+4}$	$b = x + y/(\ z + 1/6)$
2	$a = \dfrac{3{,}5 + e^{y-1}}{1 + x^2\lvert y - \mathrm{tg}\ z\rvert}$	$b = \dfrac{(y-x)^2}{x+2} + \dfrac{\lvert y-x\rvert^3}{3}$
3	$a = \dfrac{\sqrt{\lvert x-1\rvert} - \sqrt[3]{\lvert z\rvert}}{1 + \frac{x^2}{2{,}5} + \frac{y^2}{4}}$	$b = \dfrac{1 + \cos\ \frac{y}{2}}{\frac{x^2}{2} + \sin^2 z}$
4	$a = \dfrac{z + x}{y^2 + \lvert y + x^3/1{,}3\rvert}$	$b = 1 + \mathrm{tg}\dfrac{z}{2x+2y}$
5	$a = \dfrac{\cos z + 1}{\sqrt{(1 + x \cdot y)}}$	$b = x^y + \dfrac{z^2}{3 + z^2/5}$
6	$a = \dfrac{1{,}5 + \sin^2 z}{\lvert x - 2x/(1 + x^2y^2)\ \rvert}$	$b = \cos^2\dfrac{y}{z} + \sin^2 x$
7	$a = \left\lvert\dfrac{1{,}5y}{z + x^2/4}\right\rvert$	$b = x - \dfrac{x^2}{3!} + \dfrac{x^5}{5!}$
8	$a = \dfrac{\sin(x^2 - 2y + z)}{2{+}2{,}6x^y}$	$b = \cos^2\left(x^2 + \dfrac{y}{z}\right)$
9	$a = \dfrac{\cos(\ x - \pi/2)}{0{,}5 + sin^2\ x}$	$b = \dfrac{5y^2}{3z^2 + 7}$
10	$a = \dfrac{3{,}5 + \mathrm{tg}(x^2 + y)}{\lvert x - 4x/(1 + xy^2)\rvert}$	$b = \sin^2\left(\mathrm{arctg}\dfrac{1}{z+3}\right)$
11	$a = \dfrac{2{,}6 + \mathrm{tg}(x - y)}{\lvert x - 2x/(x^2 + y^2)\rvert}$	$b = \mathrm{tg}^2\dfrac{z + 2y}{2x}$
12	$a = \ln\left\lvert\dfrac{y + x^2/4}{5z}\right\rvert$	$b = 5{,}5 + \dfrac{x}{y(\mathrm{tg\ z} + 1)}$
13	$a = \dfrac{z + y/(x^2+4)}{e^{-x-2}/(x^2+4)}$	$b = \dfrac{(y-x)^2}{2} + \dfrac{\lvert y-x\rvert^3}{3}$
14	$a = \dfrac{3{,}5 + e^{y-1}}{1 + x^2\lvert y - \mathrm{tg}\ z\rvert}$	$b = x/y(\mathrm{tg\ z} + 1/6)$
15	$a = \dfrac{3 + \sin^3(\ x^2 + y)}{2 + \lvert x - x/(1 + x^2y^2\rvert}$	$b = 1 + \dfrac{z^2}{3 + z^2/5}$

Option number	a	b
16	$a = \dfrac{\sqrt{\lvert x-1\rvert} - \sqrt[3]{\lvert y\rvert}}{1{,}5 + x^2 + y^2}$	$b = \cos^2\left(\mathrm{arctg}\dfrac{1}{z+1}\right)$
17	$a = \dfrac{\cos(x - \pi/2)}{0{,}5 + \sin^2 x}$	$b = x + \dfrac{x^2}{3+z} + \dfrac{x^2}{5+z}$
18	$a = \dfrac{3{,}3 + y^2 + (x^2+2)}{e^{-0{,}5} + 1/(x^2+4)}$	$b = \cos^2\left(x^2 + \dfrac{y}{1+z}\right)$
19	$a = y + \dfrac{3{,}5x}{y^2 - \sqrt{\frac{x^2}{2y+x^2}}}$	$b = 2 + \dfrac{y^2}{3 + \frac{z^2}{1+x}} - 3x$
20	$a = \dfrac{2\cos(z-1)}{1/2 + \sin^2 z}$	$b = \sqrt{\sin^3 \dfrac{1}{x+z}}$
21	$a = \dfrac{z + y/(x^2+4)}{e^{-x}/x^2 + 4}$	$b = \sin^2\left(\mathrm{arctg}\dfrac{1}{z+3}\right)$
22	$a = \dfrac{2{,}6 + \mathrm{tg}(x-y)}{\lvert x - 2x/(x^2+y^2)\rvert}$	$b = 5{,}5 + \dfrac{x}{y(\mathrm{tg}\, z + 1)}$
23	$a = \dfrac{\sqrt{\lvert x-1\rvert} - \sqrt[3]{\lvert z\rvert}}{1 + \frac{x^2}{2{,}5} + \frac{y^2}{4}}$	$b = 5{,}5 + \dfrac{x}{y(\mathrm{tg}\, z + 1)}$
24	$a = \dfrac{z + y/(x^2+4)}{e^{-x-2}/(x^2+4)}$	$b = x/y(\mathrm{tg}\, z + 1/6)$
25	$a = \dfrac{\cos z + 2}{\sqrt{(1 + x \cdot y)}}$	$b = \dfrac{(y-x)^2}{2} + \dfrac{\lvert y-x\rvert^3}{3}$
26	$a = \dfrac{\sqrt{\lvert x-1\rvert} - \sqrt[3]{\lvert y\rvert}}{1{,}5 + x^2 + y^2}$	$b = \cos^2\left(\mathrm{arctg}\dfrac{1}{z+1}\right)$
27	$a = \left\lvert\dfrac{1{,}5y}{z + x^2/4}\right\rvert$	$b = 1 + \mathrm{tg}\dfrac{z}{2x+2y}$
28	$a = \dfrac{\cos(x - \pi/2)}{0{,}5 + \sin^2 x}$	$b = x + \dfrac{x^2}{3+z} + \dfrac{x^2}{5+z}$
29	$a = \dfrac{\cos(x - \pi/2)}{0{,}5 + \sin^2 x}$	$b = \cos^2\dfrac{y}{z} + \sin^2 x$
30	$a = \dfrac{2\cos(z-1)}{1/2 + \sin^2 z}$	$b = \dfrac{(y-x)^2}{2} + \dfrac{\lvert y-x\rvert^3}{3}$

Option number	a	b
31	$a=\dfrac{z+y/(x^2+4)}{e^{-x}/x^2+4}$	$b=\dfrac{(y+x)^2}{2}+\dfrac{\lvert y+x\rvert^3}{3}$
32	$a=\dfrac{\sqrt{\lvert x+1\rvert}-\sqrt[3]{\lvert z\rvert}}{1+\frac{x^2}{5}+\frac{y^2}{4}}$	$b=x+y/(\ z+1/6)$
33	$a=\dfrac{3{,}5+e^{y+1}}{5+x^2\lvert y+\operatorname{tg} z\rvert}$	$b=\dfrac{1+\cos\ \frac{y}{2}}{1+\sin^2 z}$
34	$a=\dfrac{z+3x}{y^2+\lvert y+x^3/1{,}3\rvert}$	$b=1+\operatorname{tg}\dfrac{z}{x+y}$
35	$a=\dfrac{cos(z+3)}{2+\sqrt{(1+x/y)}}$	$b=x^2+\dfrac{z^2}{3+z^2}$
36	$a=\dfrac{1{,}5-\sin^2 z}{\lvert x-2x/(1+y^2)\ \rvert}$	$b=\cos^2 z+\sin^2 x$
37	$a=\dfrac{\sin(x^2+y)}{\lvert x-4x/(xz^2)\rvert}$	$b=\sin^2\left(\dfrac{x+1}{z+3}\right)$
38	$a=\left\lvert\dfrac{x+2y}{z+x^2/4}\right\rvert$	$b=x-\dfrac{x^2}{3!}+\dfrac{x^3}{4!}$
39	$a=\dfrac{\sin(x^2-z)}{2+6y}$	$b=\cos^2\left(x+\dfrac{y}{z}\right)$
40	$a=\dfrac{\cos(\ x+\pi/3)}{5-\sin^2 x}$	$b=\dfrac{5y^2}{3z-x}$

Control questions

1. How can you write a script in the SimInTech system?
2. What is the structure of SimInTech programs?
3. How can you enter comments?
4. What are the rules for setting identifiers?
5. What rules should be followed when writing declarations in the SimInTech language?
6. How do the properties of the **Programming Language** block change?

7. How can you run a written program?
8. Where are the program execution results displayed?
9. What is the format for specifying numeric constants in the SimInTech language?
10. What is the format for specifying algebraic variables in the SimInTech language?
11. What formats for specifying algebraic variables in the SimInTech language do you know?
12. What arithmetic operators are used in the SimInTech language?
13. What is the format for specifying functions used in the SimInTech language?
14. What standard functions are used in the SimInTech language?
15. What functions are used to work with complex numbers in the SimInTech language?

Laboratory work № 2
Branching and cyclic programs in SimInTech

The purpose of the work: familiarization with operators designed for creating branching and cyclic programs in the SimInTech system.

Work assignment

1. Run the SimInTech program, create a new project based on the computational scheme of the solver **Control system diagram**. Add the **Programming Language** block to the project window, open its dialog box and write a program in it that searches for the maximum value of the values of two variables:

```
output a1=11, a2=2;     //initialized 2 outputs
                       //variables of the real type
if a1>a2   then am=a1  else am=a2;      //storing
//the maximum value of a1 and a2 in the am variable
output am   //output variable of real type
```

Save the project as a file in a folder as directed by the teacher under the name **lw_2_1** with the default extension **prt**. To do this, select the **File → Save Project As...** command in the project's schematic window.

Run the program by clicking on the **Start** button on the Toolbar of the project window. Add communication lines from the output ports of the **Programming Language** block that appear. Run the program again for execution. Figure 1 shows the results of this program.

The values of the two variables a1, a2 and the resulting maximum am value are output to the output ports of the **Programming Language** block.

This program uses the conditional transition operator **if**, which is used to organize branching algorithms and has the following format:

if <condition or logical expression>
    then <the operator when the condition is met>
    {**else** <operator if the condition is not met>};

If the condition is true, the statement will be executed after the **then** keyword, and after it is executed, the program switches to the next command line. If the condition is false, then the operator will be executed after the **else** keyword and switch to the next command line. However, you can also set difficult conditions.

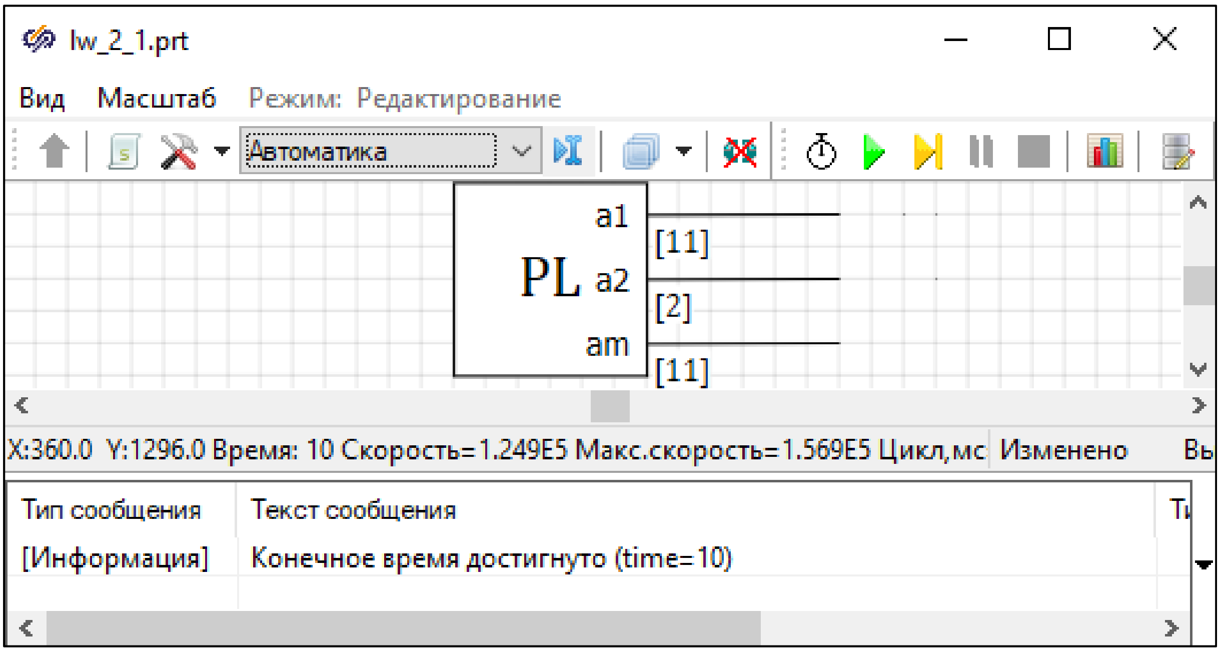


Figure 1

Edit the text of the program by setting the value a2=20 and check if it is working correctly. Close the project window.

2. Create a new project based on the calculation scheme of the solver **Control system diagram**. Add the **Programming Language** block to the project window and write a program in its dialog box in which the condition after the **if** keyword is set as a logical expression and operator brackets are used in each branch:

```
 var x1:boolean=1, x2:boolean=1;
 //initialization of binary type variables
var y, u; //variables of the real type
if x1=true and x2=true then
//a condition in the form of a logical and
    begin
    y=2;u=5;
    end
            else
    begin
    y=0;u=0;
    end;
output y,u;
```

After keywords, it is possible to use operator brackets **begin ... end,** which can be nested, combining several operators into one:

begin
<operation 1>;
...
<operation n>;
end;

Save the project in your folder as a file named **lw_2_2** with the **prt** extension and run the program for execution.

Add communication lines from the output ports y и u of the **Programming Language** block that appear.

Run the program again for execution.

Figure 2 shows the results of this program for x1=1 and x2=1.

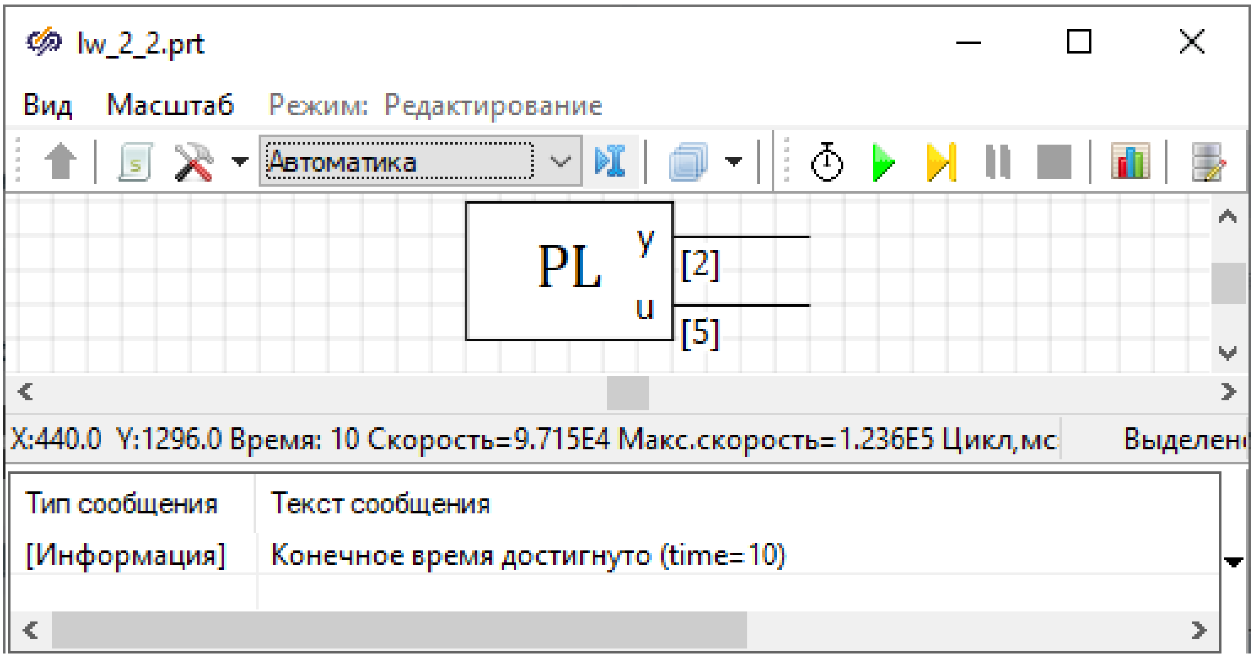


Figure 2

Since the condition after the **if** statement is **true**, the operators between the operator brackets after the **then** keyword are executed.

Edit the text of the program by setting the value of at least one of the variables x1 or x2 to 0 and check that it is working correctly. Close the project window.

Assignment for the lab report

Basic level

Write a program in SimInTech that calculates the value of the step function Z from Table 1 in **double** format for your task option.

Report

Insert the assignment statement, the text of the received program, as well as the results of its execution in the schematic window of the project for all branches of the algorithm into the laboratory report.

Complicated level

Three numbers are given. Arrange them in ascending order for even options or in descending order for odd options.

Report

Insert the assignment statement, the text of the received program, as well as the results of its execution in the schematic window of the project into the laboratory report.

Table 1

Option number	Function
1, 11, 21, 31	$Z = \begin{cases} 10{,}5x^2 + 1, & x \le 0; \\ 5(x+1)^3, & 0 < x < 1; \\ \sqrt{x^2+1}, & x \ge 1 \end{cases}$
2, 12, 22, 32	$Z = \begin{cases} \cos(x-1), & -2\pi < x \le 0; \\ \cos x + 1, & 0 < x < \pi; \\ \sin(x+0{,}5), & \pi \le x \le 2\pi \end{cases}$
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	$Z = \begin{cases} \sin 3x, & -\pi < x \le 0; \\ \cos(x+1), & 0 < x < \pi/2; \\ \sin 2x, & \pi/2 \le x \le \pi \end{cases}$
5, 15, 25, 35	$Z = \begin{cases} 2\sin(x), & x \le 0; \\ -\sin(2x), & 0 < x < \pi; \\ 3\sin(x), & \pi \le x \le 2\pi \end{cases}$

Option number	Function
6, 16, 26, 36	$Z = \begin{cases} \sin(x) \quad , & x \le 0; \\ -\sin(x), & 0 < x < \pi; \\ \sin(x) \quad , & \pi \le x \le 2\pi \end{cases}$
7, 17, 27, 37	$Z = \begin{cases} x^4 - 12 \quad , & x \le -1; \\ (2x+1)^3 \quad , & -1 < x < 1; \\ \sqrt{\dfrac{x+11}{x}}, & x \ge 1 \end{cases}$
8, 18, 28, 38	$Z = \begin{cases} 15x + 1 \quad , & x \le 0; \\ 5(x+1)^2 \quad , & 0 < x < 1; \\ \sqrt{x^3 - 1}, & x \ge 1 \end{cases}$
9, 19, 29, 39	$Z = \begin{cases} 0{,}5x^3 + 1, & x \le -10; \\ (x+1)^2 \quad , & -10 < x < 10; \\ \sqrt{x+10} \quad , & x \ge 10 \end{cases}$
10, 20, 30, 40	$Z = \begin{cases} \cos(x) \quad , & x \le 0; \\ -\cos(x), & 0 < x < \pi; \\ \cos(x) \quad , & \pi \le x \le 2\pi \end{cases}$

3. To get acquainted with the operators designed for the organization of cyclic programs in SimInTech. Recall that cyclic algorithms are called algorithms that contain *cycles* – repeatedly repeated sections of the algorithm.

In SimInTech, the **for** operator is used to organize a cycle with a set number of repetitions, which has the following format:

for (<**name** = <**initial value**>,<**final value**> {,<**step**>})
<loop operator>;

The **name** argument is the name of the loop control variable. At each step, the value of the variable changes by an amount equal to the specified **step**. If no step is specified, then by default it is equal to **one** if the final value is greater than the initial value, or **minus one** if otherwise.

The loop statement is executed until the value of the loop variable is greater than the final value.

To create a new project based on the calculation scheme of the solver **Control system diagram**. Add the **Programming Language** block to the project window and write a program in its dialog box using the **for** operator, which calculates the value of the function $y = \sum_{x=1}^{10}(x * 2)$ for x varying from 1 to 10 in increments of 1:

```
var y;
y=0;
for (x=1,10,1)// cycle condition
    y=y+x*2; // the loop operator
output x,y;
```

Save the project in your folder as a file named **lw_2_3** and run the program for execution. Add communication lines from the output ports of the **Programming Language** block that appear. Run the program again for execution.

Figure 3 shows the results of this program.

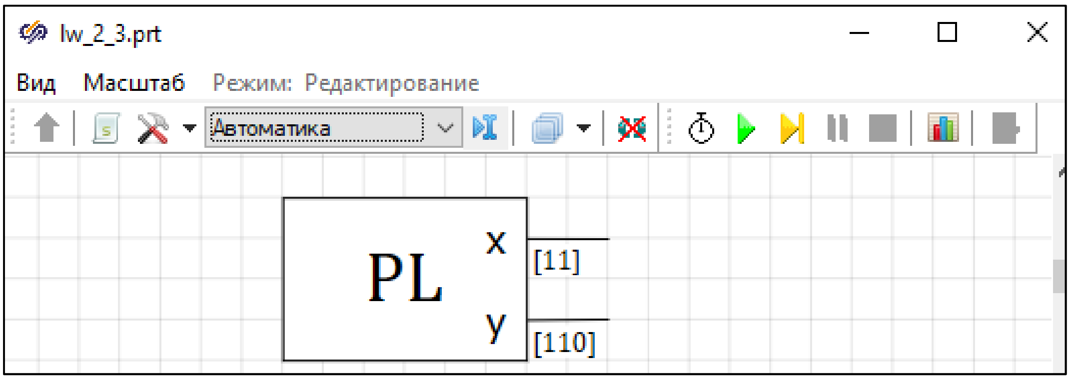


Figure 3

4. In SimInTech, the **while** operator is used to organize a loop with a precondition:

while <loop condition> **do**

<loop operator>;

The loop statement is executed as long as the loop condition is satisfied.

Create a new project based on the calculation scheme of the solver **Control system diagram**.

Add the **Programming Language** block to the project window and write a program in its dialog box using the **while** operator, which calculates the value of the function $y = \sum_{x=1}^{10}(x * 2)$ for x from 1 to 10 in increments of 1:

```
var y, x=1;
while x<=10 do // cycle condition
  begin
  y=y+x*2;  //the loop operator
  x=x+1;    //changing the loop variable
  end;
output x,y;
```

Save the project in your folder as a file named **lw_2_4** and run the program for execution.

Add communication lines from the output ports of the **Programming Language** block that appear.

Run the program again for execution. Make sure that the results of this program match the results of the previous program from point 3.

5. In SimInTech, the repeat operator is used to organize a loop with a postcondition:

```
repeat
    <loop operator>;
until <exit condition>;
```

The loop statement is executed as long as the exit condition of the loop is met.

Create a new project based on the calculation scheme of the solver **Control system diagram**.

Add the **Programming Language** block to the project window and write a program in its dialog box using the **repeat** operator, which calculates the value of the function $y = \sum_{x=1}^{10}(x * 2)$ for x from 1 to 10 in increments of 1:

```
var y;
    x=1; y=0;
repeat
  begin
  y=y+x*2; // the loop operator
  x=x+1;//changing the loop variable
  end;
until x>10 // cycle condition
output x,y;
```

Save the project in your folder as a file named **lw_2_5** and run the program for execution.

Add communication lines from the output ports of the **Programming Language** block that appear.

Run the program again for execution. Make sure that the results of this program match the results of the previous program from point 3.

6. Loops can be nested when it is necessary to repeat a certain sequence of statements inside the body of the loop.

Nested loops cannot intersect: until the inner loop has ended, the outer loop cannot end.

In order to save machine time, all operators that do not depend on the parameters of the inner loop should be transferred to the outer loop.

Let's consider an example of a program using the **for** operator, in which the value of the function $a = \sum_{n=1}^{3}\left(\sqrt{10 + x^n}\right)$ is calculated for the following values x = 1, 2, 3.

Create a new project based on the computational scheme of the solver **Control system diagram**. Add the **Programming Language** block to the project window and write it in its programs dialog box:

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

Save the project in your folder as a file named **lw_2_6** and run the program for execution.

Add communication lines from the output ports of the **Programming Language** block that appear.

Run the program again for execution.

Figure 4 shows the results of this program.

7. Let's consider an example of a program using the **for** operator, in which the value of the function $a = \prod_{n=1}^{3}\sqrt{10 + x^n}$ is calculated for the following values x = 1, 2, 3.

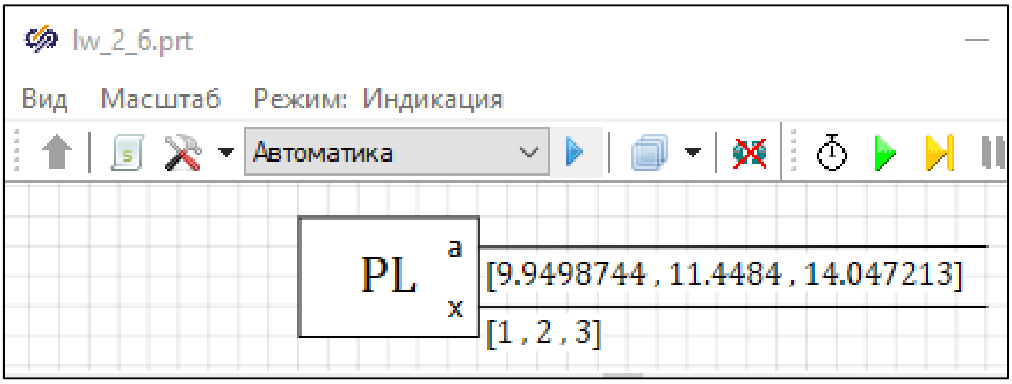


Figure 4

Create a new project based on the computational scheme of the solver **Control system diagram**. Add the **Programming Language** block to the project window and write it in its programs dialog box:

```
output a[3],x[3];
x=[1,2,3];
for (i=1,3,1) //cycle condition
begin
 s=1;
 for (n=1,3,1) //cycle condition
     begin
     a[i]=s*sqrt(10+x[i]^n);
     s=a[i];
     end;
end;
```

Save the project in your folder as a file named **lw_2_7** and run the program for execution.

Add communication lines from the output ports of the **Programming Language** block that appear.

Run the program again for execution.

Figure 5 shows the results of this program.

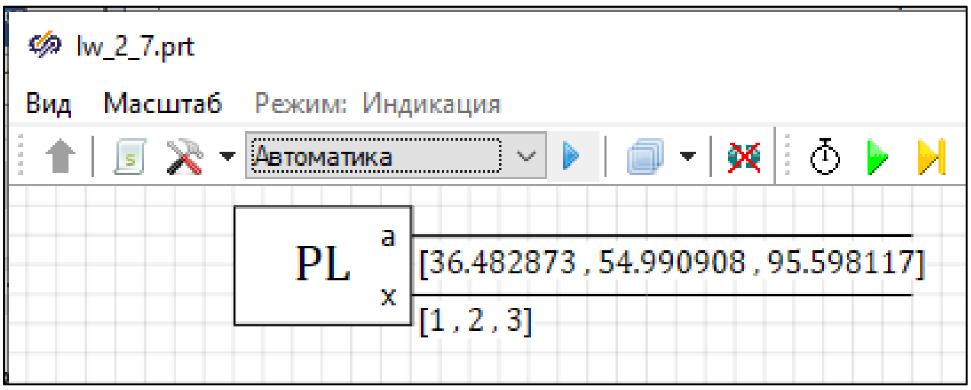


Figure 5

Assignment for the lab report

Basic level

Write a loop program with a for statement to calculate the value of the function A(x) from the Table 2 in double format for three values of x equal to N-1, N, and N+1 (N is the option number).

Complicated level

Write 2 loop programs with the while and repeat statements to calculate the value of the function A(x) from the Table 2 in double format for three values of x equal to N-1, N, and N+1 (N is the option number).

Report

Insert the assignment statement, the text of the received program, as well as the results of its execution in the schematic window of the project into the laboratory report.

Table 2

Option number	**Function**	**Option number**	**Function**
1	$A = \sum_{n=1}^{5} \sqrt[4]{10n + x^4}$	21	$A = \prod_{k=1}^{4} \sin(3k + x)$
2	$A = \prod_{n=1}^{8} (x^2 - n)$	22	$A = \sum_{n=1}^{5} \sqrt{n^2 + 2x}$
3	$A = \prod_{k=1}^{6} \frac{\cos x}{2k}$	23	$A = \prod_{k=1}^{8} \sqrt{(xk + 2)^3}$
4	$A = \sum_{k=1}^{4} (x^4 + 4k)$	24	$A = \sum_{n=1}^{6} \frac{x^n}{n}$
5	$A = \prod_{k=1}^{5} (e^x + kx)$	25	$A = \sum_{k=0}^{5} tg\ (x + k)$
6	$A = \sum_{k=1}^{10} \frac{1 - x}{k}$	26	$A = \sum_{i=1}^{10} \frac{1 + x}{i + 1}$
7	$A = \sum_{i=1}^{8} \frac{1 + x}{i^3}$	27	$A = \prod_{k=1}^{5} \sqrt{(xk + 1)^2}$
8	$A = \prod_{k=1}^{6} \frac{\sin x}{2k + 1}$	28	$A = \sum_{i=1}^{8} \frac{(x)}{i(i + 1)}$
9	$A = \sum_{i=1}^{7} \frac{x}{(2i + 1)^2}$	29	$A = \sum_{i=1}^{6} \frac{(x+2)}{i(i-1)}$

Option number	Function	Option number	Function
10	$A=\sum_{i=1}^{8}\frac{(-1)^i}{(x+1)5}$	30	$A=\sum_{i=1}^{5}\frac{(x)^2}{4^i+5^i}$
11	$A=\prod_{n=1}^{6}(2x^2-n)$	31	$A=\sum_{k=1}^{4}(x^2+2k)$
12	$A=\sum_{n=1}^{6}\frac{x^3}{n}$	32	$A=\sum_{n=1}^{5}\sqrt[4]{n^2+x^2}$
13	$A=\sum_{k=0}^{3}(x+k)^2$	33	$A=\sum_{i=1}^{8}\frac{(i+1)}{(2x+1)}$
14	$A=\prod_{k=1}^{4}\sqrt{(xk+1)}$	34	$A=\prod_{k=1}^{3}(2^x+kx)$
15	$A=\sum_{i=1}^{6}\frac{(-1)^x}{x(i-3)}$	35	$A=\sum_{n=1}^{4}\sqrt{n^4+x}$
16	$A=\prod_{k=1}^{8}\sqrt{(xk+2)^3}$	36	$A=\sum_{n=1}^{5}\sqrt[4]{2n+x^3}$
17	$A=\sum_{n=1}^{5}\sqrt{n^2+2x}$	37	$A=\prod_{k=1}^{6}\frac{\cos x}{2k}$
18	$A=\prod_{k=1}^{5}\sqrt{(xk+1)^2}$	38	$A=\sum_{k=1}^{10}\frac{1-x}{k}$
19	$A=\sum_{k=0}^{5}\mathrm{tg}\ (x+k)$	39	$A=\prod_{k=1}^{6}\frac{\sin x}{2k+1}$
20	$A=\prod_{k=1}^{3}(2^x+kx)$	40	$A=\sum_{i=1}^{8}\frac{(-1)^x}{(x+1)(i+2)}$

Control questions

1 Name the purpose and format of the **if** statement in the SimInTech programming language.

2 Explain the sequence of command execution after executing the operator after the keyword **then**.

3 Give examples of conditions and logical expressions that can be placed after the **if** keyword.

4 What are operator brackets used for? What format do they have?

5 Which operators are used in SimInTech to organize cycles and what is their format?

6 Why and how are nested loops organized in SimInTech?

Laboratory work № 3
Working with arrays in SimInTech environment

The purpose of the work: introduction to algorithms and functions for processing vectors and matrices, as well as basic operations on them.

In mathematics, an array is an ordered set of elements, each of which stores a single value, identified by one or more indexes. A two-dimensional array Xi, j, or matrix is understood to be a rectangular set of numbers characterized by two integers: the number of rows ($i \geq 1$) and columns ($j \geq 1$).

Special cases of matrices are: row vector (consists of one row) and column vector (consists of one column). In SimInTech, the **[ ]** operator is used to pack elements into an array. It has the following format:

"array or matrix"["element or row number"
{, "column number"}]

If the elements are real or integers, then the result is a vector of real numbers. If the elements are vectors of real numbers, the result is a matrix of real numbers [1].

Work assignment

1. Run the SimInTech program, create a new project based on the calculation scheme of the solver **Control system diagram**. Add the **Programming Language** block to the project window, open its dialog box, and write a program in it:

```
output v1[2],  // v1 array of size 2
       m[2,3],  // matrix m 2 rows by 3 columns
       v2[2,1], // vector column v2 consisting of two rows
       v3[1,2]; //vector row v3 consisting of two columns
  v1=[1,2];      m=[[1,2,3];[4,5,6]];
  v2=[[1];[2]];  v3=[[1,2]];
```

The program initializes the output variables of the block in the form of matrices. Then the elements of the matrix rows are assigned values by specifying them inside square brackets separated by commas. For a matrix and a column vector, each row is located inside the outer square brackets and is separated from the adjacent row by a semicolon or a dot.

Save the project in your folder as a file named **lw_3_1** with the default extension **prt** and run the program for execution.

Add communication lines from the output ports of the **Programming Language** block that appear. Run the program again for execution.

Figure 1 shows the results of this program in the **View Variable Values** window, which shows that local variables are stored in the computer's memory, and the values of their elements in mathematics are written as follows:

– one-dimensional array $\mathrm{v1} = [1 \quad 2]$.

– two-dimensional array (matrix) $\mathrm{m} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$.

– column vector $\mathrm{v2} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

– string vector $\mathrm{v3} = [1 \quad 2]$.

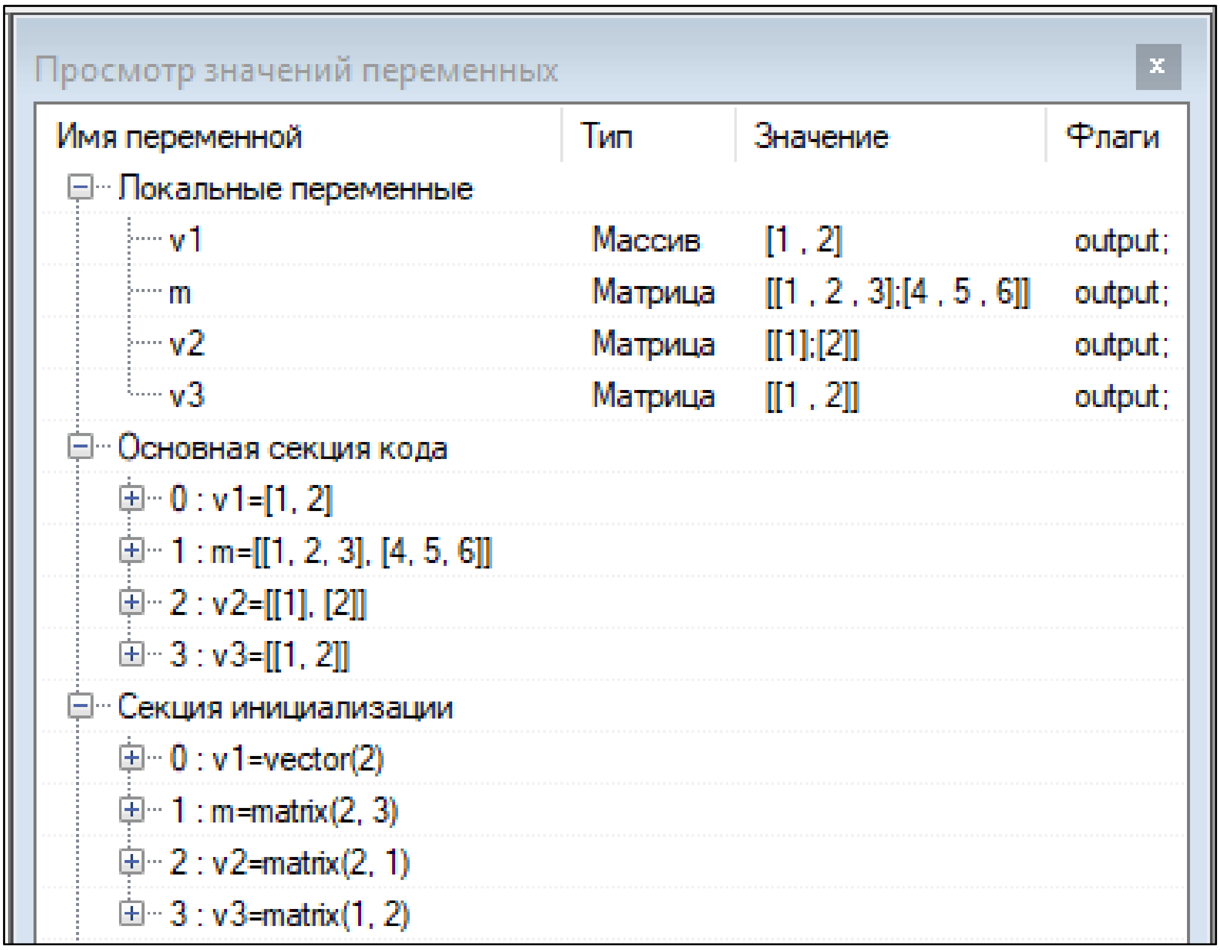


Figure 1

Note that in SimInTech, the matrix elements are displayed in square brackets in one line: first, the elements of the first line are in square brackets, then after the semicolon, the elements of the second line are in square brackets, and so on.

In the Figure 2 shows the values of array elements and matrices that are output in the project window to the corresponding output ports of the **Programming Language** block.

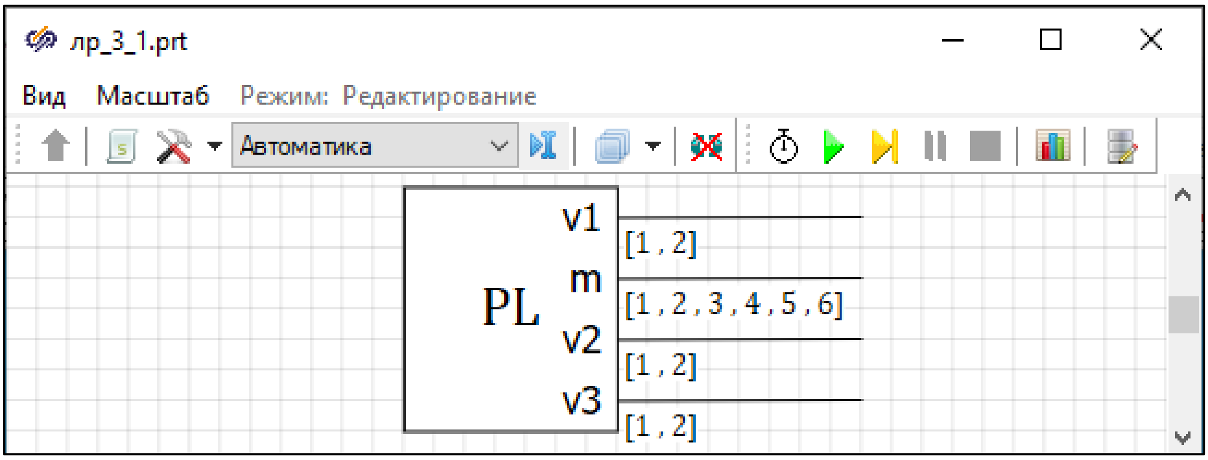


Figure 2

Matrix elements are output to the output ports in square brackets in one line: first the elements of the first line, then after the comma the elements of the second line, and so on.

2. Create a new project in SimInTech based on the calculation scheme of the solver **Control system diagram**. Add the **Programming Language** block to the project window, open its dialog box, and write a program in it:

```
var a[2,2] ;        //matrix A is 2x2 in size
output a1[2],a2[2];//arrays a1 and a2 of size 2
  for (i=1,2,1)     //outer loop condition
    begin
    for (j=1,2,1)   //inner cycle condition
    a[i,j]=trunc(100*rand)-50;
    end;
a1=a[1];//1 row of matrix a is entered into the a1 array
a2=a[2];//2 row of matrix a is entered into the a2 array
```

The program generates a matrix **a** of size 2×2, as well as two one-dimensional arrays **a1** and **a2** of length 2 as output variables of the **Programming Language** block. Then, using two nested loops, the elements of matrix **a** are assigned values generated by the random number generator (**rand** function).

In conclusion, the values of the elements of the first row of matrix **a** are stored in array **a1**, and the second row in array **a2**.

The **rand** function returns a random number of **a** real type in the range from 0 to 1. To get an integer from it in the range from 0 to 100, we use the **trunc** function, which returns an integer part of a real number. When you subtract 50 from it, you get an integer in the range from –50 to 50, which is assigned to the element of the matrix **a[i, j]**.

Note that when accessing a matrix element in SimInTech the **[ ]** operator is used, which has the following format:

"array or matrix" ["element or row number"
{, "column number"}]

The **[ ]** operator returns the value of the element with the specified number. Numbering starts with 1. The return value type corresponds to the type of array elements. If one index is specified for the matrix, then the row vector of the matrix is returned, if two – the value of the corresponding matrix element is returned.

The value returned by the operator can be assigned to another variable. The element separator can be a comma, semicolon, or space.

Save the project in your folder as a file named **lw_3_2** and run the program for execution.

Add communication lines from the output ports of the **Programming Language** block that have appear. Run the program again for execution.

Figure 3 shows the values of local program variables in the **View Variable Values** window (**Просмотр значений переменных**).

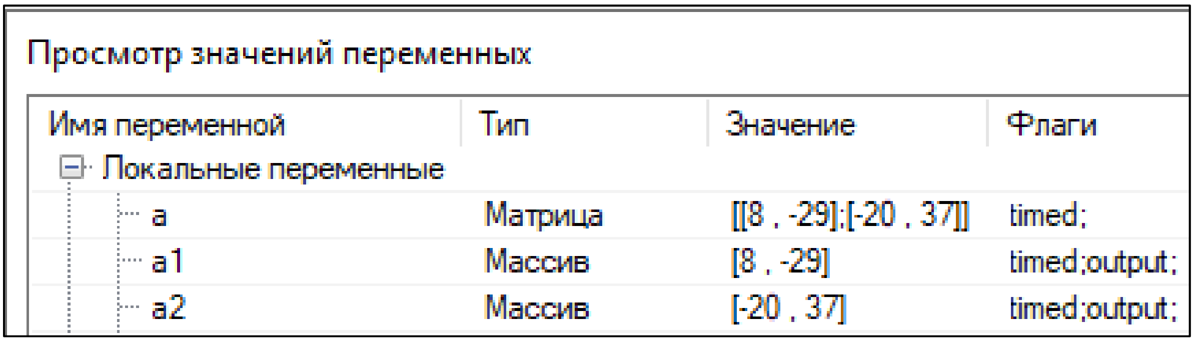


Figure 3

In the Figure 4, the values of array elements **a1** and **a2** are displayed in the project window in the corresponding output ports of the **Programming Language** block.

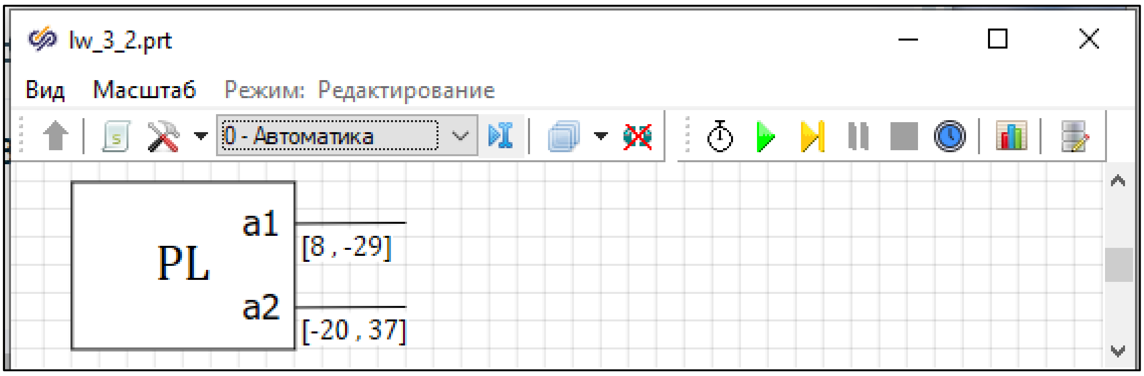


Figure 4

Add two **AutoLabel** primitives next to the output and ports of the **Programming Language** block by selecting the upper link of the LMB and opening the **Properties** window for it. Select the line with the name of the signal on the **Parameters (Параметры)** tab (Figure 5 – **y**) and click the button .

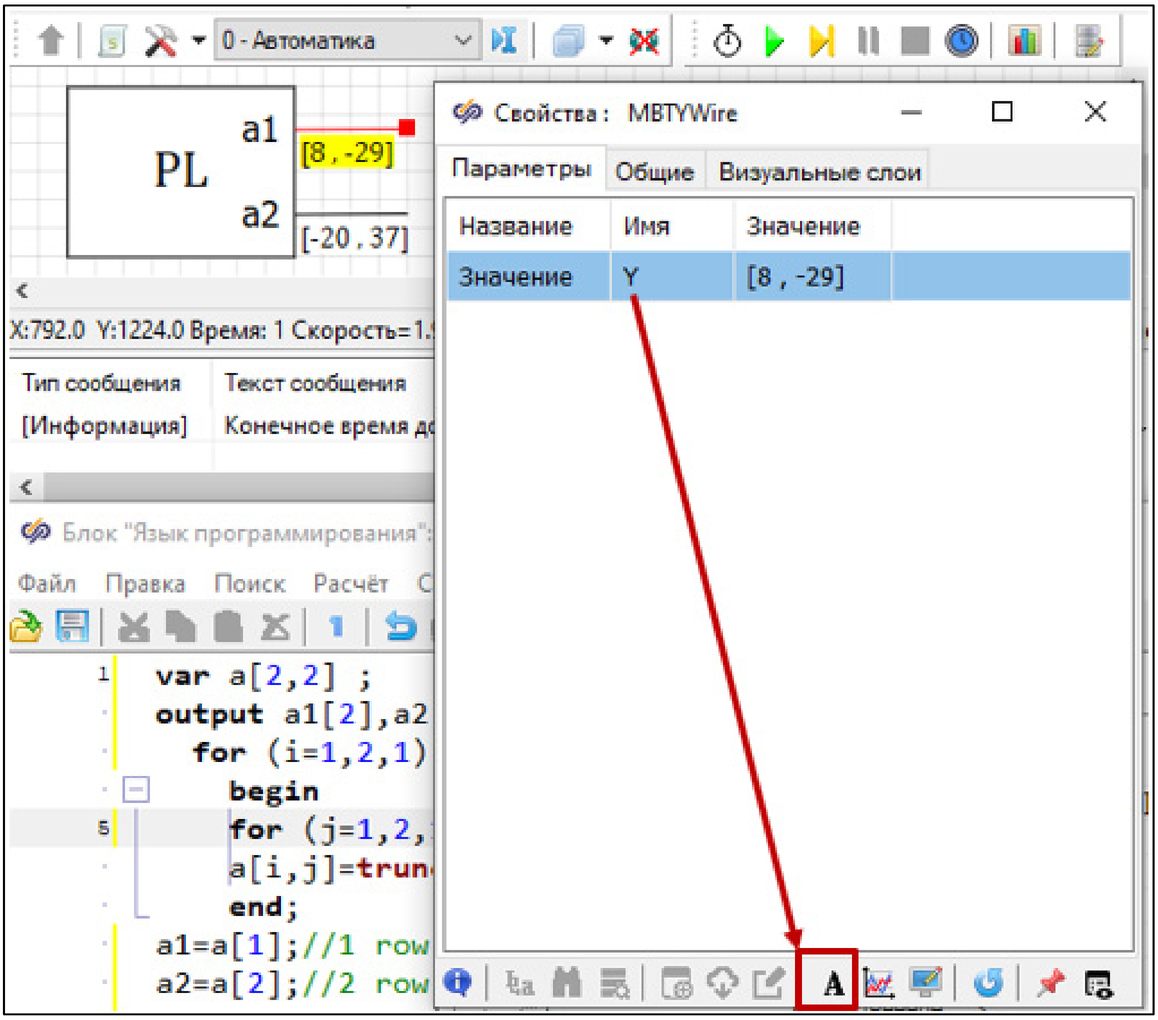


Figure 5

In the **Object Signature** (**Подпись объектов**) window that opens (Figure 6), specify the required **Signature text** (**Текст подписи**) on the **Style** (**Стиль**) panel, which will be displayed before the numeric value of the output parameter.

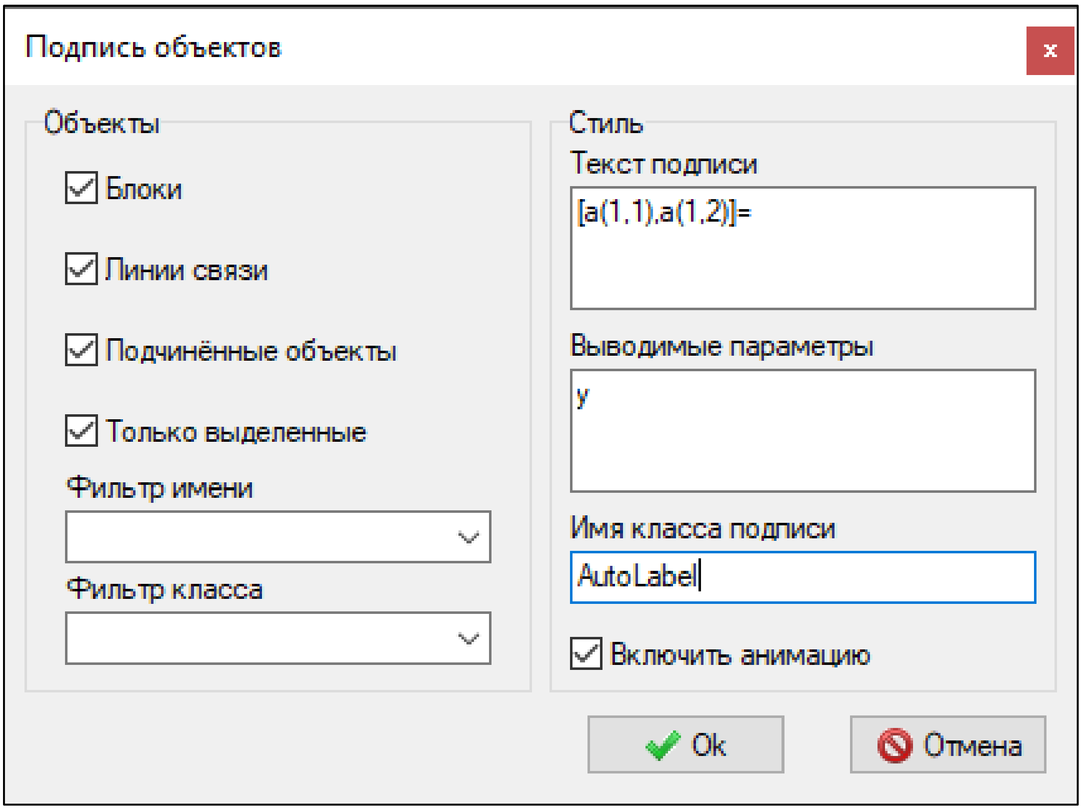


Figure 6

The **Output parameters** (**Выводимые параметры**) field will contain the name of the block parameter (**y**), the value of which will be output in numeric form.

The field **Signature Class Name** (**Имя класса подписи)** contains a unique block class.

In order for the displayed value to change when the parameter value changes, the **Enable animation** (**Включить анимацию**) option must be activated [1].

When you click **Ok** in the schematic window of the project, the **AutoLabel block** appears before the output value of the y parameter (Figure 7).

Similarly, add the **AutoLabel** block to the second section **a2** of the **Programming Language** block. Run the program several times for execution and make sure that various random values appear synchronously on the output ports of the **Programming Language** block and in the corresponding **AutoLabel** blocks.

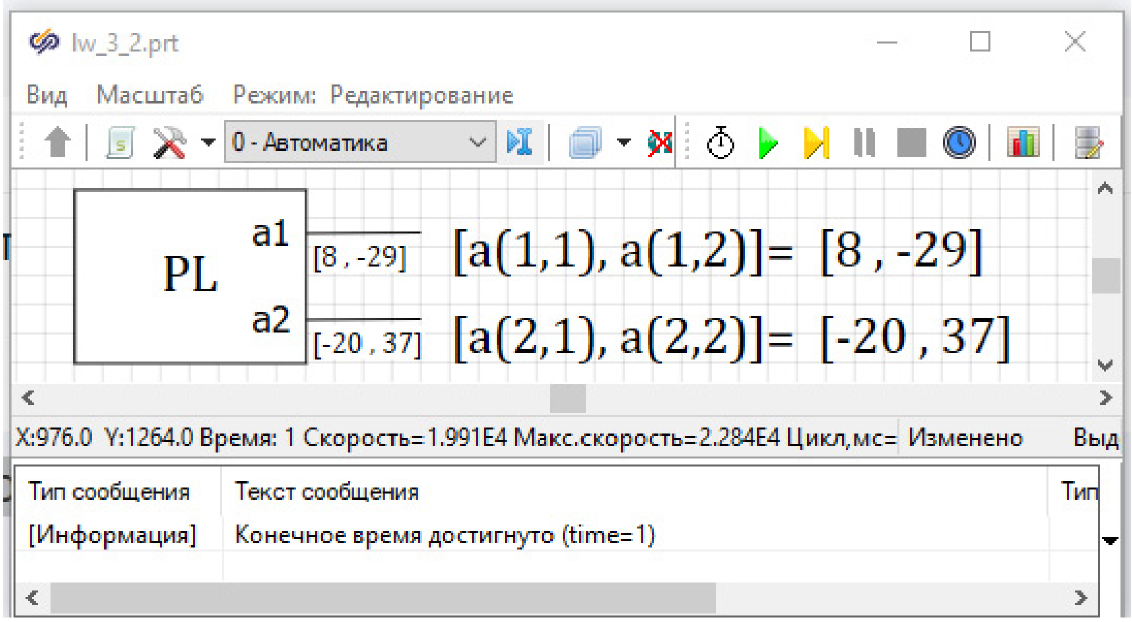


Figure 7

Assignment for the lab report

Basic level

Write a program that generates an A1 matrix with the following element values for its variant number N:

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

Output the obtained values of the elements of the matrix rows to the corresponding output ports of the **Programming Language** block, as well as to the corresponding **AutoLabel** blocks.

Report

Insert the assignment statement, the text of the received program, as well as the results of its execution in the schematic window of the project into the laboratory report.

3. Create a new project in SimInTech based on the calculation scheme of the solver **Control system diagram**.

Click the «Script» button from the **Layers** section on the project editing window panel and write a program in the **Page Script** (**Скрипт страницы**) window in which matrices and vectors are generated using various SimInTech functions:

```
M=matrix(2,3);   //the zero matrix
M1=matrix1(2,3); //the matrix of units
const X = [1, 2, 3];
M2=diag(X);     // the diagonal matrix
M3=eye(3);//all elements of the main diagonal are equal to 1
M4=linspace(0,30,7);//the vector of arithmetic progression
```

The **matrix(n,m)** function returns a matrix of dimension **n** by **m**, in which all elements are zero.

The **matrix1(n,m)** function returns a matrix of dimension **n** by **m**, in which all elements are one.

The **diag(X)** function returns a square matrix with diagonal elements equal to the elements of the vector **X** (diagonal matrix). All other elements of the matrix outside the main diagonal are zero.

The **eye(n)** function returns an **n** by **n** matrix with all diagonal elements equal to one.

The **linspace(xmin,xmax, n)** function returns an arithmetic progression vector from the value of **xmin** to the value of **xmax**. The dimension of the vector is determined by the parameter **n**.

Save the project in your folder as a file named **lw_3_3** and run the program by pressing Run Script .

Figure 8 shows the results of this program in the **View Variable Values window.**

Assignment for the lab report

Basic level

Write a program in the **Script page** window to create a string vector V1:

V1 = (N + 1) (N + 2) ... (N + 10).

Report

Insert the task statement, the text of the received program, and the results of its execution in the **View Variable Values** window in the lab report.

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;

Figure 8

4. Create a new project in SimInTech based on the calculation scheme of the solver **Control system diagram**. Add the **Programming Language** block to the project window, open its dialog box, and write the program in it:

```
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;
```

The program initializes the matrix a of size 3×2 and the real variable **m** as output variables of the **Programming Language** block.

Then the variable **m** is assigned a value that is obviously smaller than the values of the elements of the matrix **a** to be formed later. Since these values in the program can take values in the range from –50 to 50, you can take, for example, **m = –100**.

Then, in the loop, each element of the matrix is compared with the value of the variable **m**, and, if the condition of the **if** operator is fulfilled, then the variable **m** will be assigned the value of this matrix element.

Save the project in your folder as a file named **lw_3_4** and run the program for execution. Add communication lines from the output ports that appear and restart the program.

Figure 9 shows the results of the program, from which it can be seen that the value output to the **m** port is the maximum of all the values of the matrix elements.

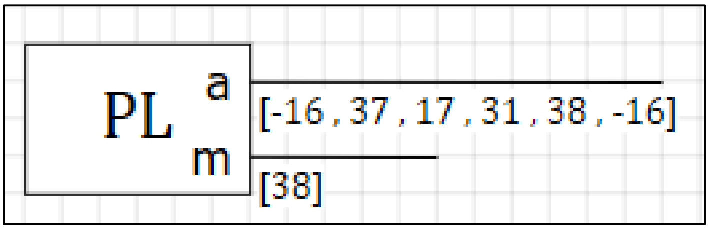


Figure 9

Run the program several times to make sure that it is working correctly.

5. For processing vectors and matrices, as well as for obtaining information about them in SimInTech, there are many functions that can significantly simplify the program. For example, to calculate the maximum value from two values or from the values of a matrix, you can use the **max** function.

Edit a previously written program by adding a real variable **m1** as the output variableой of the **Programming Language** block and assigning it a value equal to **max(a)**:

```
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);
```

Save the project in your folder as a file named **lw_3_5** and run the program for execution. Add communication lines from the output porta **m1** that appears and restart the program.

Figure10 shows the results of the program, from which it can be seen that the values output to the ports **m1** and **m** are the same.

Run the program several times to make sure that it is working correctly.

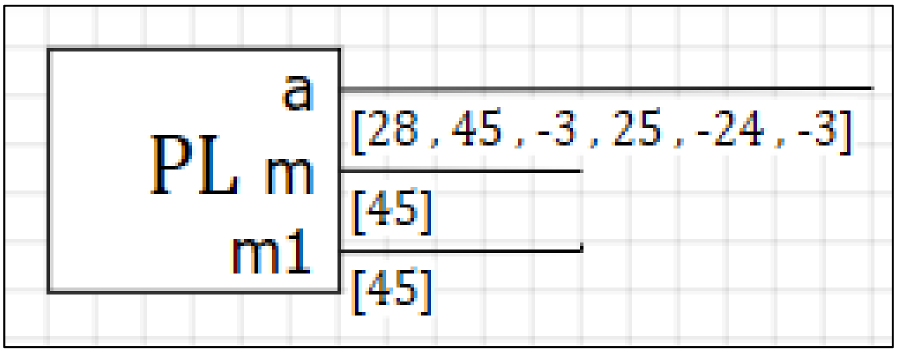


Figure 10

6. Table 1 shows other SimInTech functions designed for processing vectors and matrices, as well as for obtaining information about them [2].

Table 1

Function	Description
transp(M) transp(V)	The function performs a transpose of the input matrix M or vector V
det(M)	The function returns the value of the determinant of the matrix. The matrix M must be square
inv(M)	The function returns a matrix that is the inverse of the matrix M. The matrix M must be square. When the inverse matrix is multiplied by the original matrix, the identity matrix is obtained (all diagonal elements of which are equal to 1)
sort(V)	The function sorts the array in ascending order of elements. For complex vectors, the sorting is done by modules of numbers
invsort(V)	The function sorts the array in descending order of elements. For complex vectors, the sorting is done by modules of numb
mean(V)	The function returns the average value of the elements of the input vector V. The elements of the vector V must be real numbers.
activeelement(V)	The function returns the index of the first nonzero element in the array V. If all the elements are zero, it returns zero. Indexes start with one
rows(M)	Function returns the number of rows of the matrix
cols(M)	Function returns the number of columns of the matrix
cols(V)	Function returns the number of elements of the array V

Create a new project in SimInTech based on the calculation scheme of the solver **General view model scheme** and write a program in the **Page Script** window:

```
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);  //vector string
v=linspace(100,10,10); //array
v2=transp(v);//column vector from row vector
a1_t=transp(a1); //the transposed matrix
a1_inv=inv(a1);  //the inverse matrix
aa=a1*a1_inv;    //the identity matrix
s=sort(v);   //sorting the vector in ascending order
s1=sort(a1); //sorting the matrix in ascending order
vcr=mean(v1);//the average value of the vector elements
```

The program initializes a matrix **a1** of size 3×3, a row vector **v1** and a column vector **v2** of length 10.

The values of **a1** matrix elements are assigned inside the outer square brackets by specifying them inside the square brackets separated by commas for lines separated by semicolons.

The values of the elements of the vector string **v1** and the array **v** are set as an arithmetic progression using the function **linspace**.

The values of the elements of the vector column **v2** are formed using the vector transpose function **transp(v)**, which replaces a row with a column. When the matrix **a1** is transposed, each row is replaced by a column.

Next, in the program, the inverse matrix **a1_inv** is formed for the **a1** matrix using the **inv** function.

Then the variable **aa** is assigned the value of the product of the matrix **a1** and the inverse matrix **a1_inv**, which returns the identity matrix.

The **sort** function sorts the elements of vector **v** and matrix **a1** in ascending order, and the **mean** function rotates the average value of the elements of vector **v1**.

Save the project in your folder as a file named **lw_3_6** and run the program for execution.

Figure 11 shows the results of the program operation in the **View Variable Values** window.

Просмотр значений переменных

Имя переменной	Тип	Значение
Локальные п...		
a1	Матрица	[[-2 , 5 , -3];[-20 , 10 , -1];[2 , -12 , -5]]
v1	Матрица	[[1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10]]
v2	Матрица	[[100];[90];[80];[70];[60];[50];[40];[30];[20];[10]]
v	Массив	[100 , 90 , 80 , 70 , 60 , 50 , 40 , 30 , 20 , 10]
a1_t	Матрица	[[-2 , -20 , 2];[5 , 10 , -12];[-3 , -1 , -5]]
a1_inv	Матрица	[[0.05927342256 , -0.05831739962 , -0.02390057361];[0.09...
aa	Матрица	[[1 , 0 , 5.551115123E-17];[0 , 1 , 2.775557562E-17];[0 , 0 , 1]]
s	Массив	[10 , 20 , 30 , 40 , 50 , 60 , 70 , 80 , 90 , 100]
s1	Массив	[-20 , -12 , -5 , -3 , -2 , -1 , 2 , 5 , 10]
vcr	Вещественное	5.5

Figure 11

Note that for the **a1_inv** matrix, Figure 11 shows only the elements of the first row and one element of the second row.

The values of some elements of the **aa** unit matrix are given taking into account the calculation error in floating-point format.

Assignment for the lab report

Basic level

Write a program from clause 6 for the following arrays for your own version of task N:

– matrices A1:

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

– vector-strings V1, the values of the elements of which are given in the form of arithmetic progression:

$$V1 = (N + 1)\ (N + 2)\ ...\ (N + 10);$$

– a column vector V2, whose values are given as an arithmetic progression:

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

Complicated level

Give examples of using other functions from Table 1 to process arrays for your specific task.

Report

Insert the assignment statement, the text of the received program, and the results of its execution into the laboratory report in the **View Variable Values** window.

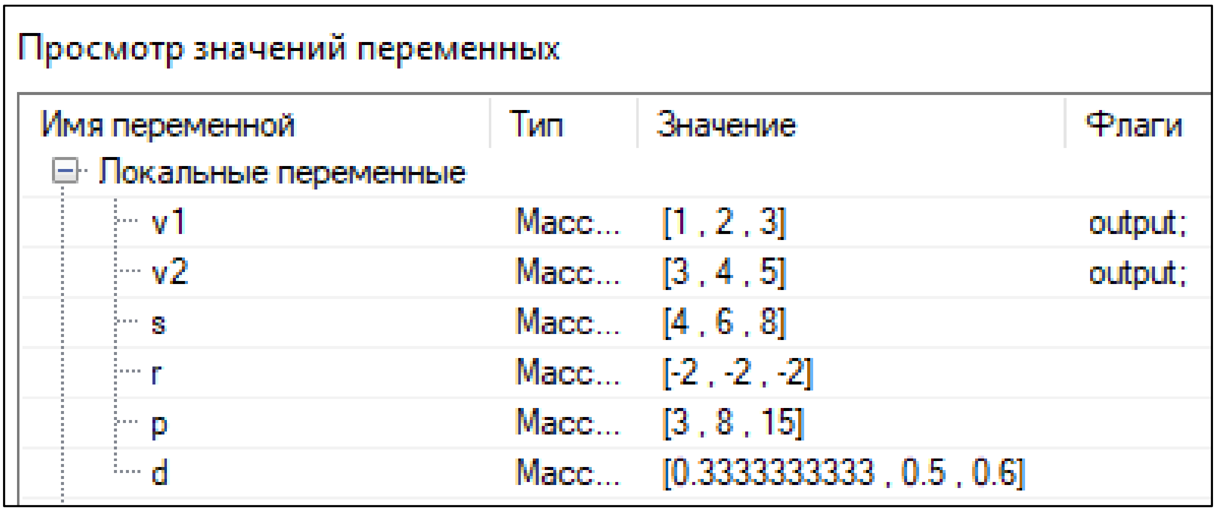
7. In the SimInTech programming language, both traditional operations on vectors and matrices provided for by vector computing in mathematics, as well as their element-by-element transformations, can be performed. For piecemeal addition, subtraction, multiplication, and division of vectors and matrices, they must have the same dimension.

Create a new project in SimInTech based on the calculation scheme of the solver **Control system diagram** and write a program in the **Script page** window:

```
output v1[3],v2[3];
v1=[1,2,3];  v2=[3,4,5];
s=v1+v2; //Piecemeal addition of vectors
r=v1-v2; //Element-by-element vector subtraction
p=v1*v2; //Element-wise vector multiplication
d=v1/v2; //Element-wise division of vectors
```

Save the project in your folder as a file named **lw_3_7** and run the program for execution.

Figure 12 shows the results of the program operation in the **View Variable Values** window.


Просмотр значений переменных

Имя переменной	Тип	Значение	Флаги
Локальные переменные			
v1	Масс...	[1 , 2 , 3]	output;
v2	Масс...	[3 , 4 , 5]	output;
s	Масс...	[4 , 6 , 8]	
r	Масс...	[-2 , -2 , -2]	
p	Масс...	[3 , 8 , 15]	
d	Масс...	[0.3333333333 , 0.5 , 0.6]	

Figure 12

Create a new project in SimInTech based on the calculation scheme of the solver **Control system diagram** and write a program in the **Script page** window:

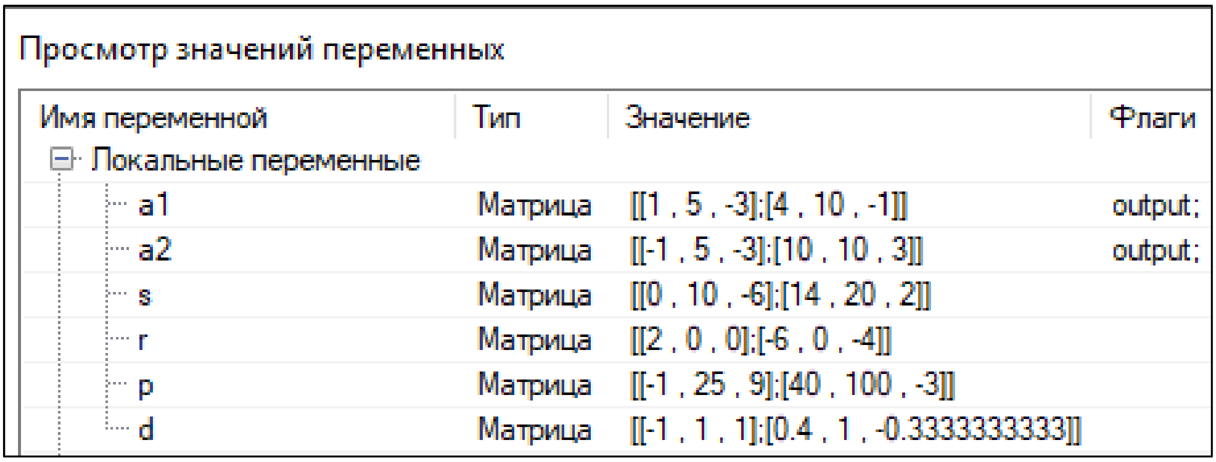
```
output a1[2,3],a2[2,3];
a1=[[1,5,-3];[4,10,-1]];
a2=[[-1,5,-3];[10,10,3]];
s=a1+a2;  //Piecemeal addition of matrices
r=a1-a2;  //Piecemeal matrix subtraction
p=a1.*a2; //Element-wise matrix multiplication
d=a1./a2; //Piecemeal division of matrices
```

Save the project in your folder as a file named **lw_3_8** and run the program for execution.

Figure 13 shows the results of the program operation in the **View Variable Values** window.

For vector matrix multiplication and division, the number of columns of the first matrix must match the number of rows of the second matrix.

Division is performed by multiplying by the inverse matrix.


Просмотр значений переменных

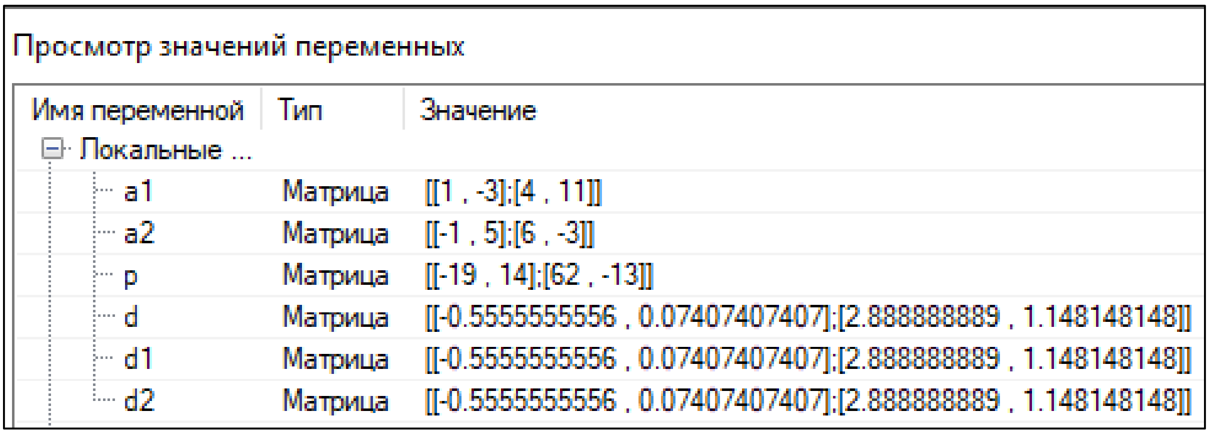
Имя переменной	Тип	Значение	Флаги
Локальные переменные			
a1	Матрица	[[1 , 5 , -3];[4 , 10 , -1]]	output;
a2	Матрица	[[-1 , 5 , -3];[10 , 10 , 3]]	output;
s	Матрица	[[0 , 10 , -6];[14 , 20 , 2]]	
r	Матрица	[[2 , 0 , 0];[-6 , 0 , -4]]	
p	Матрица	[[-1 , 25 , 9];[40 , 100 , -3]]	
d	Матрица	[[-1 , 1 , 1];[0.4 , 1 , -0.3333333333]]	

Figure 13

Create a new project in SimInTech based on the calculation scheme of the solver **Control system diagram** and write a program in the **Script page** window:

```
output a1[2,2],a2[2,2];
a1=[[1,-3];[4,11]];
a2=[[-1,5];[6,-3]];
p=a1*a2;        //matrix multiplication
d=a1/a2;        //dividing the a1 matrix by a2
d1=a1*inv(a2);//dividing the a1 matrix by a2
d2=a1*a2^-1;  //dividing the a1 matrix by a2
```

Save the project in your folder as a file named **lw_3_9** and run the program for execution. Figure 14 shows the results of the program operation in the **View Variable Values** window.


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	[[-0.5555555556 , 0.07407407407];[2.888888889 , 1.148148148]]
d1	Матрица	[[-0.5555555556 , 0.07407407407];[2.888888889 , 1.148148148]]
d2	Матрица	[[-0.5555555556 , 0.07407407407];[2.888888889 , 1.148148148]]

Figure 14

8. Matrix inversion is used to solve a system of linear equations

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

if you specify it in matrix form as

$$\mathrm{AX} = \mathrm{B}, \tag{1}$$

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

System (1) has a unique solution equal to

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

if its principal determinant is nonzero $det(A) \neq 0$.

Assignment for the lab report

Basic level

Write a program for solving a system of linear equations for your own version of task N:

– the main determinant of which is:

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

– the column vector of free terms is:

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

Insert the assignment statement, the text of the received program, and the results of its execution into the laboratory report in the **View Variable Values** window.

Control questions

1. How can the dimensions of arrays be set in SimInTech?
2. How can the values of array elements be set in SimInTech?
3. How are ordered arrays formed in SimInTech?
4. What functions can be used to generate vectors and matrices in SimInTech?
5. What SimInTech functions can be used to process vectors and matrices and obtain information about them?
6. What are the ways to access the elements, rows and columns of a given matrix in the SimInTech system?
7. How can SimInTech perform the traditional operations on vectors and matrices provided for by vector calculation in mathematics, as well as piecemeal transformations of vectors and matrices?
9. How does SimInTech solve a system of linear equations given in matrix form?

Laboratory work № 4
Data visualization in SimInTech

Purpose of the work: to familiarize with the possibilities of data visualization and charting.

Work assignment

1. Run the SimInTech program, create a new project based on the calculation scheme of the solver **Control system diagram**.

Insert the **Graph Y from X** block from the **Data Output** tab into the project window (Figure 1).

Figure 1

Add the **Programming Language** block to the project window, open its dialog box and write a program in it:

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

The program uses the **linspace** function to generate 11 values of the vector **x**, varying from –5 to 5 in increments of 1, and then the values of the function $y=x^2$ are calculated.

The values of the vectors **x** and **y** are output to the corresponding ports of the **Programming Language** block.

Save the project in your folder as a file named **lw_4_1** and run the program for execution.

Connect the output ports of the **Programming Language** block that have appeared with the corresponding input ports of the **Graph Y from X** block.

If you restart the program for execution and press the **Show values on Communication lines** key in the projects window, the corresponding values of the output variables **x** and **y** of the **Programming Language** block will be displayed under the communication lines (see Figure2).

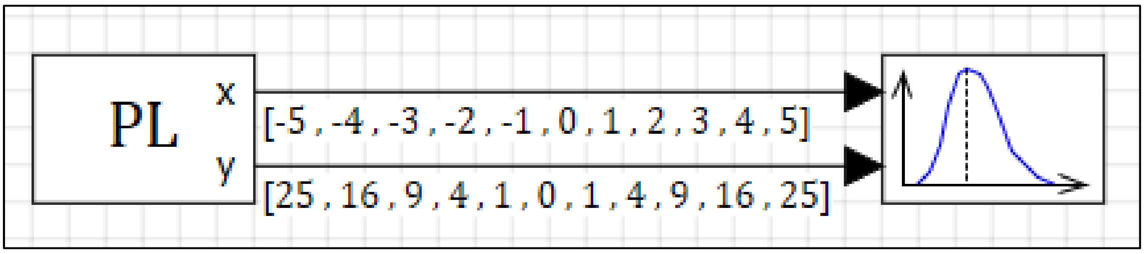


Figure 2

The simulation results can also be seen on the screen of the **Graph Y from X** block if you double-click on it (Figure 3).

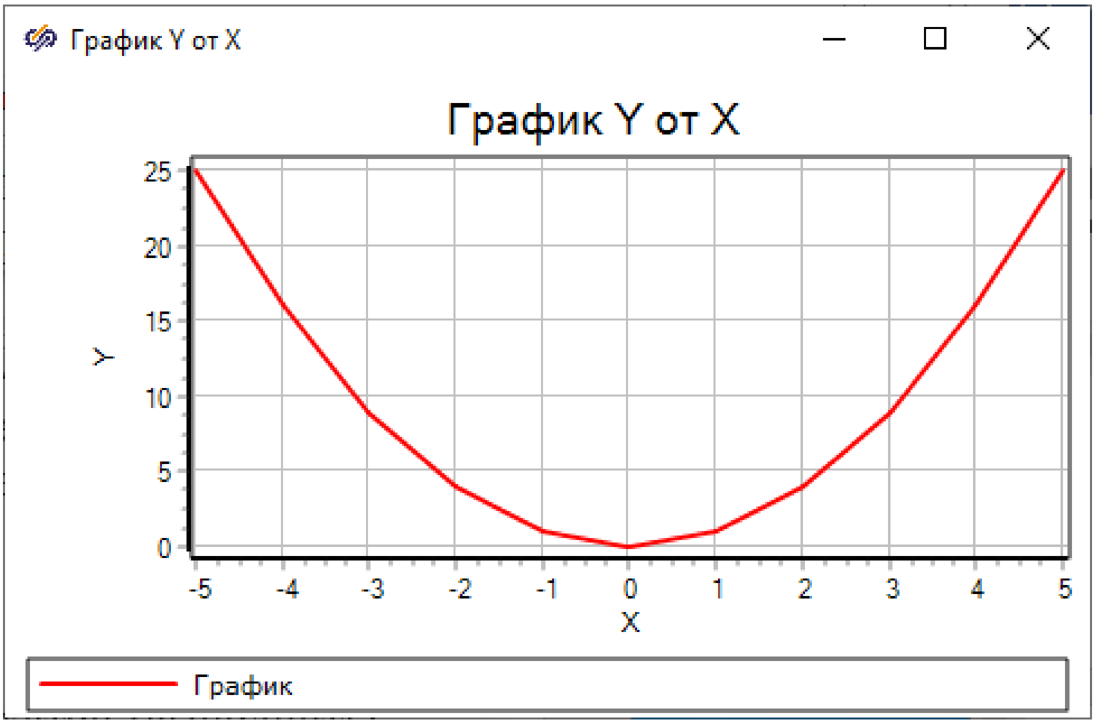


Figure 3

Move the cursor to the central part of the graph window and make a right-click. As a result, a context menu will appear, the **Properties** (**Свойства**) item of which allows you to open a dialog box with the title **Graph Properties** (**Свойства графика**), which sets the settings for its parameters.

Figure 4 shows an example of the **Graph Properties** window with standard default settings, in which the **Line Style** (**Стиль линии**) to **Dashed** (**Штриховая**) and the **Point** (**Точки**) view to **Squares** (**Квадратики**) were selected from the list on the **Graphs and Axes** (**Графики и оси**) tab.

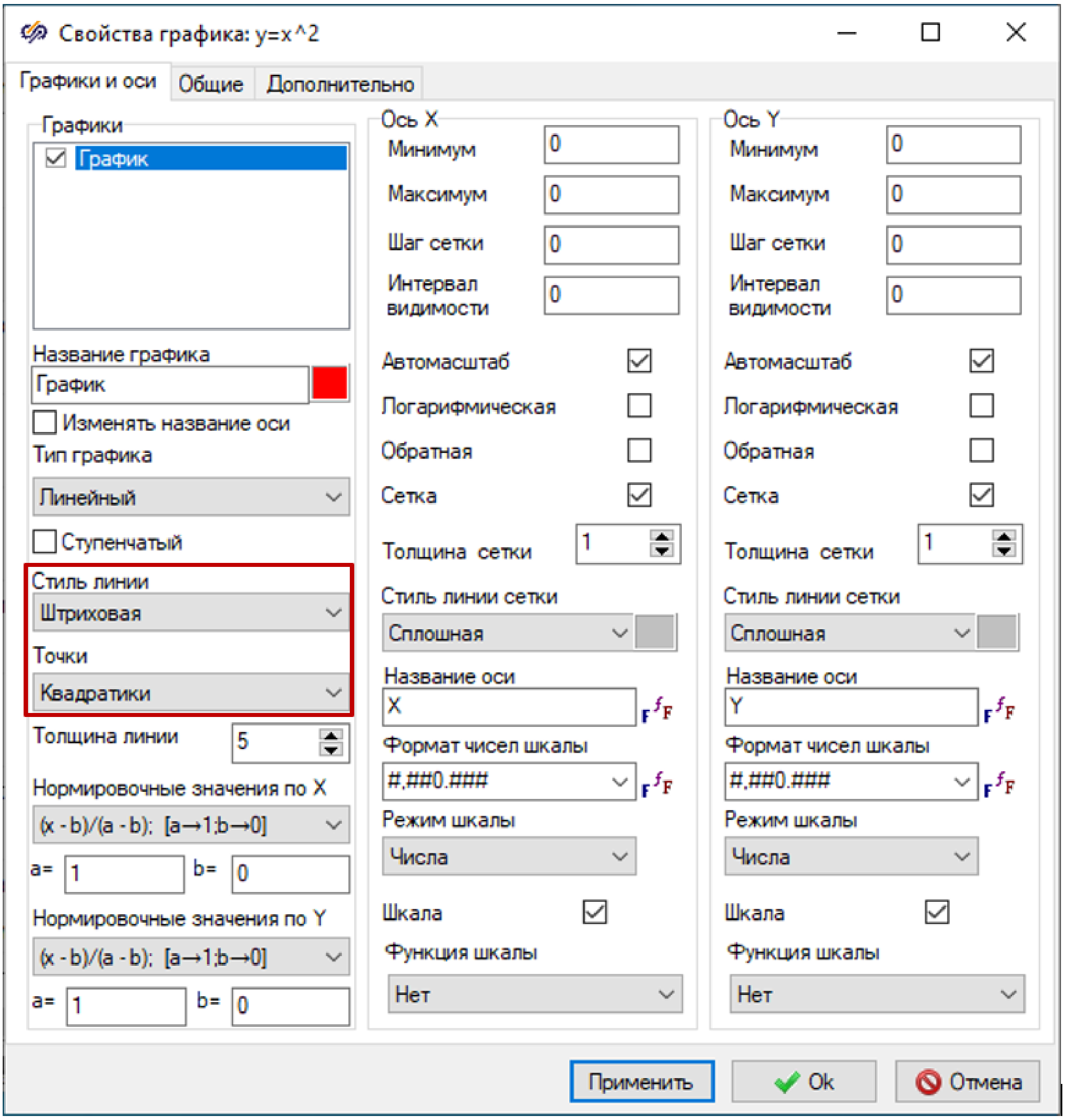


Figure 4

Note that if you uncheck the **Autoscale** (**Автомасштаб**) box, you can display part of the graph in the window by specifying the required **Y** and **X** coordinates in the **Minimum** (**Минимум**) and **Maximum** (**Максимум**) windows.

Figure 5 shows an example of the **Graph Properties** window on the **General** (**Общие**) tab, in which, in order to remove the graph legend, the checkbox in the **Show Legend** (**Показывать легенду**) window was removed.

It is advisable to use the legend if more than one graph is displayed on the screen of the **Graph Y from X** block.

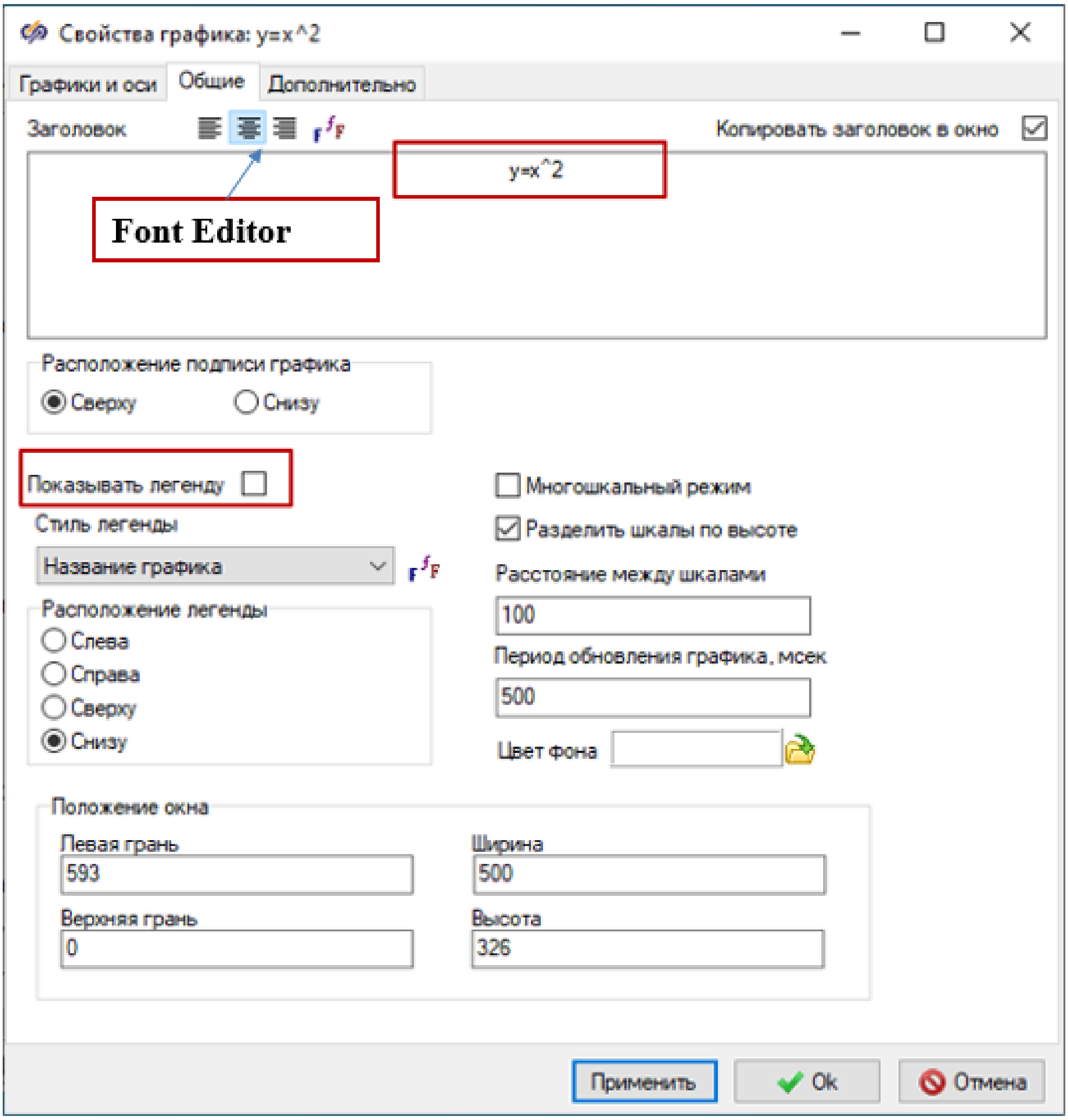


Figure 5

In addition, the **Chart title** (**Заголовок**) has been changed to **y=x^2**.

If necessary, you can change the formatting of the title: alignment options and various font settings in the **Font Editor** window.

All changes made to the settings in окне the **Chart Properties** window will take effect after you click **Apply (Применить)** or **Ok**.

Figure 6 shows, how to change the appearance of the graph in this case.

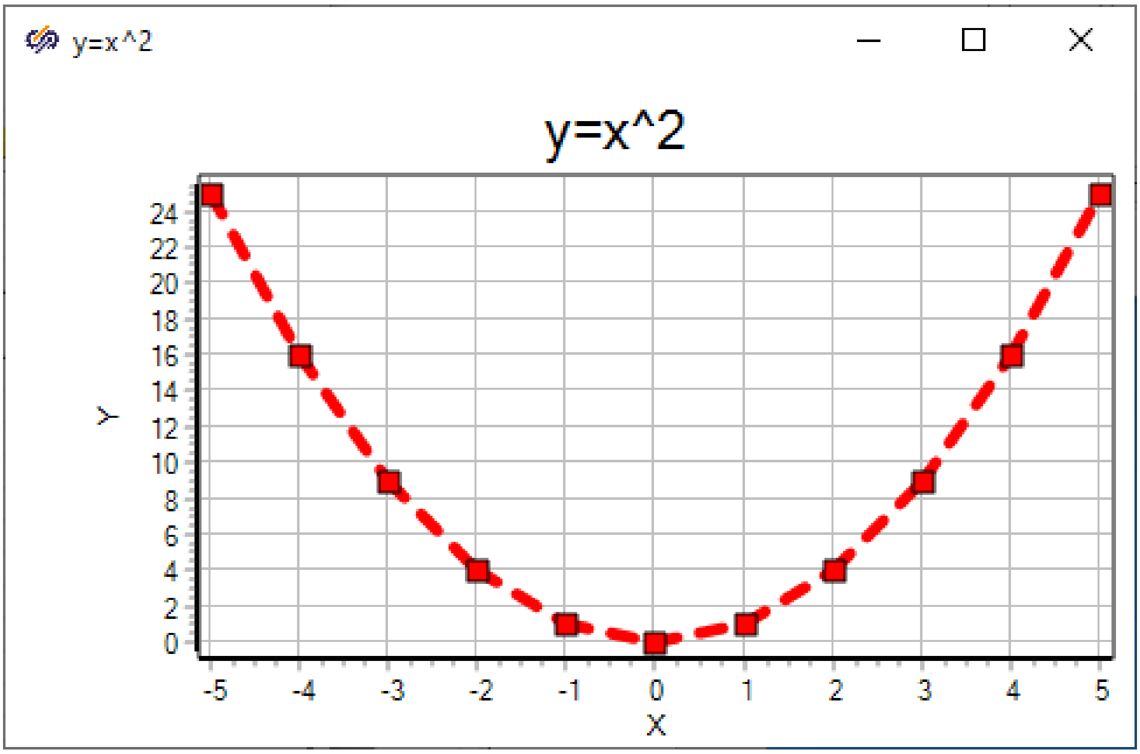


Figure 6

Assignment for the lab report

Basic level

Plot a graph of the function in the specified range along the x-axis and with a specified step in the Cartesian coordinate system for a given assignment option (Table 1). Set the settings in the properties window of the graphical window by analogy with Figures 4–5.

Report

Insert into the laboratory report the wording of the tasks, the text of the programs received, as well as the results of their execution on the screen of the **Graph Y from X** block.

Complicated level

Consider other settings that can be set in the properties window of the graphical window, apply them to your graph, and insert at least three graph variations with different settings into the laboratory report.

Table 1

Option number	Function	Range of change of x	Step change of x
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	$Y = \dfrac{1}{x+4}$	–1 – 3	0,5
5	$y=1 - e^{2x}$	0 – 5	0,1
6	$Y = \dfrac{x+44}{x-23}$	–8 – 8	1
7	$Y = \dfrac{x}{x-31}$	–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	$y = \dfrac{x^3 + x}{\|x\|}$	10 – 50	2
15	$y == \dfrac{x^3 - x^2}{2\|x - 1\|}$	2 – 18	1
16	$Y = \dfrac{2x}{x+100}$	–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
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	$y = 1+2\,e^{2x}$	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
32	$Y = \dfrac{x+44}{x-23}$	–8 – 8	1

Option number	Function	Range of change of x	Step change of x
33	$y = (x^3 - x)$	0 – 200	20
34	$y = (1 + \|x\|)(2 - \|x\|)$	0 – 36	3
35	$y = \dfrac{x^3 - x^2}{2\|x - 1\|}$	2 – 18	1
36	$y = 1 - x^2$	0 – 1	0,01
37	$y = \|x^2 - x + 2\|$	–15 – 15	3
38	$y = \|x^2 + x - 2\|$	–10 – 10	0,2
39	$y = (x^3 - x)$	0 – 30	2
40	$Y = \dfrac{x+44}{x-23}$	–8 – 8	1

2. Create a new project based on the calculation scheme of the solver **Control system diagram**.

Add the **Programming Language** block from the **Dynamic** tab and the block **Graph Y from X** from the **Data Output** tab to the project window.

Write a program in the **Programming Language Block** dialog box:

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

The program plots one period of the trigonometric function y = sin(x/2+30°) at 61 points on the x-axis in the Cartesian coordinate system. The argument of trigonometric functions in SimInTech must be set in radians.

In the program, the argument of a trigonometric function without taking into account the initial phase is formed as a vector of length 61 using the **linspace** function. The initial phase of the harmonic signal **fi** is also formed using the **linspace** function as a vector of length 61.

The values of the **x** and **y** vectors are output to the corresponding ports of the **Programming Language** block.

Save the project in your folder as a file named **lw_4_2** and run the program for execution.

Connect the output ports of the **Programming Language** block that have appeared with the corresponding input ports of the block **Graph Y from X**. Restart the program, double-click the LMB on the block **Graph Y from X**.

Figure 7 shows the simulation results on the screen of the block **Graph Y from X** after changing the title in the **Graph Properties** window and the **Point** view to **Squares.**

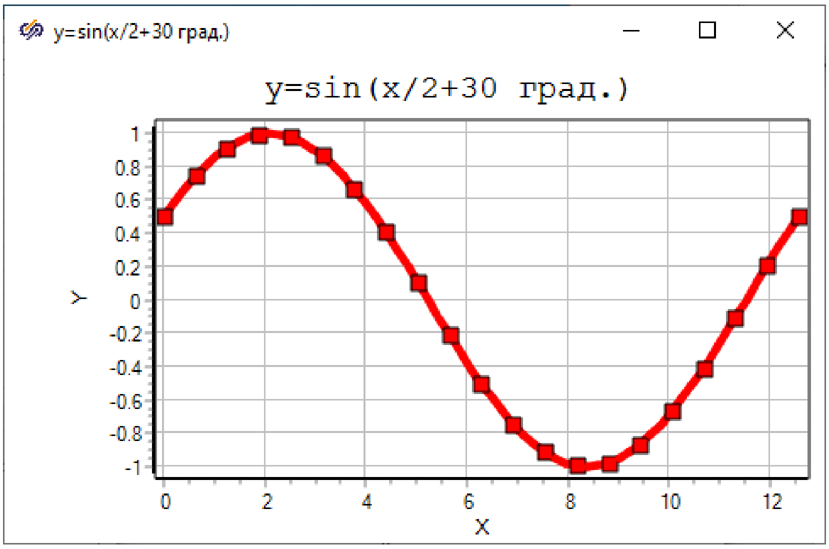


Figure 7

Assignment for the lab report

Basic level

Plot one period of the trigonometric function (Table 2) at 51 points along the x-axis in the Cartesian coordinate system for the specified assignment option. Set the settings in the properties window of the graphical window at your discretion.

Report

Insert into the laboratory report the wording of the tasks, the text of the programs received, as well as the results of their execution on the screen of the **Graph Y from X** block.

Table 2

Option number	Function	Option number	Function
1	$y=\cos(x/2+60°)$	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	$y=\cos(4x-60°)$	23	$y = \sin(3x + 60°)$
4	$y=\cos 2x+\sin x$	24	$y = 2\cos x - \sin x$

Option number	Function	Option number	Function
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	$y = \cos(3x\text{-}45°)$	27	$y = \cos(x - 15°)$
8	$y = 2\sin(x + 90°)$	28	$y = 2\sin x + 2\cos x$
9	$y = \sin(2x – 30°)$	29	$y = 2 + \sin(x)$
10	$y = \cos(4x – 45°)$	30	$y = \sin(x + 45°)$
11	$y = 2\sin(x + 90°)$	31	$y = \cos(5x + 45°)$
12	$y = \sin(3x + 60°)$	32	$y = \cos(x + 60°)$
13	$y = x + \lvert\sin x\rvert$	33	$y = \sin x + \lvert\sin x\rvert$
14	$y = \lvert\cos x\rvert + \cos\lvert x\rvert$	34	$y = \lvert\sin x\rvert + \sin\lvert x\rvert$
15	$y = 2\sin x\lvert\cos x\rvert$	35	$y = 2\cos x\lvert\sin x\rvert$
16	$y = \cos(4x – 60°)$	36	$y = \cos(4x – 60°)$
17	$y = \cos(3x\text{-}30°)$	37	$y = \sin(x + 45°)$
18	$y = \sin(5x + 45°)$	38	$y = \cos(2x + 60°)$
19	$y = \sin(3x + 60°)$	39	$y = \cos(x - 15°)$
20	$y = 2\sin x\lvert\cos x\rvert$	40	$y = \sin(3x – 30°)$

3. Save the project in your folder as a file named **lw_4_3**.

Delete all communication lines, open the properties window of the **Graph Y from X** block and set the value **2** for the **Number of graphs** (**Количество графиков**) property on the **Properties** tab (see Figure 8).

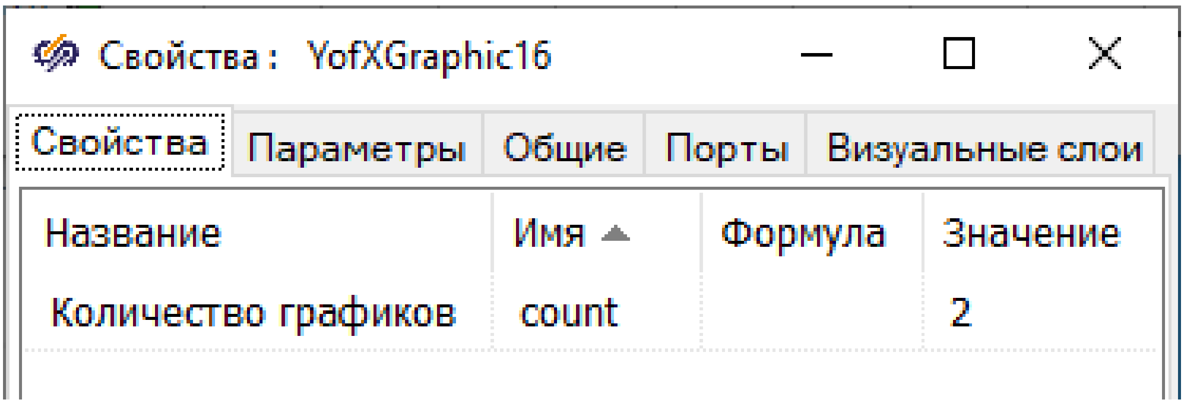


Figure 8

Write a program in the **Programming Language Block** dialog box:

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

Save the changes in the project and run the program for execution. As a result, the blocks will have additional output ports (see Figure 9).

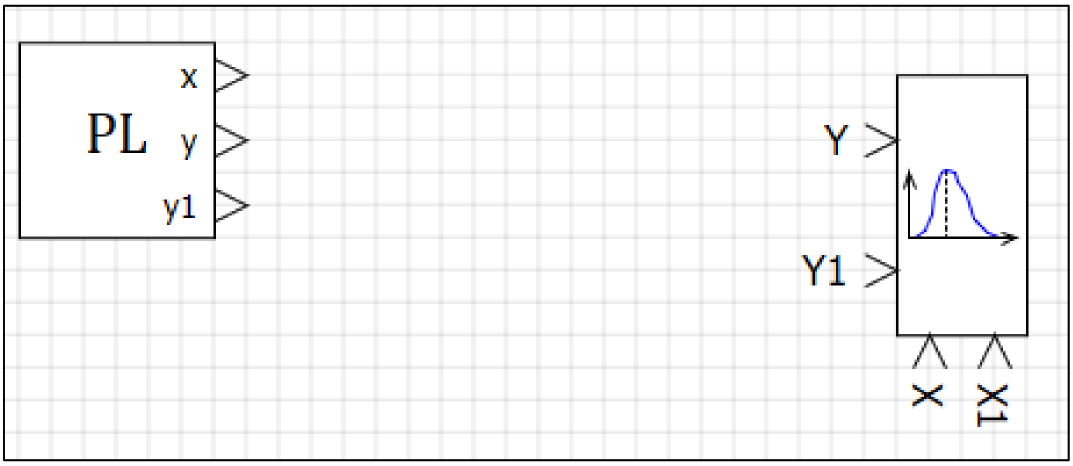


Figure 9

Connect the output ports of the **Programming Language** block that have appeared with the corresponding input ports of the block **Graph Y from X** in accordance with Figure 10.

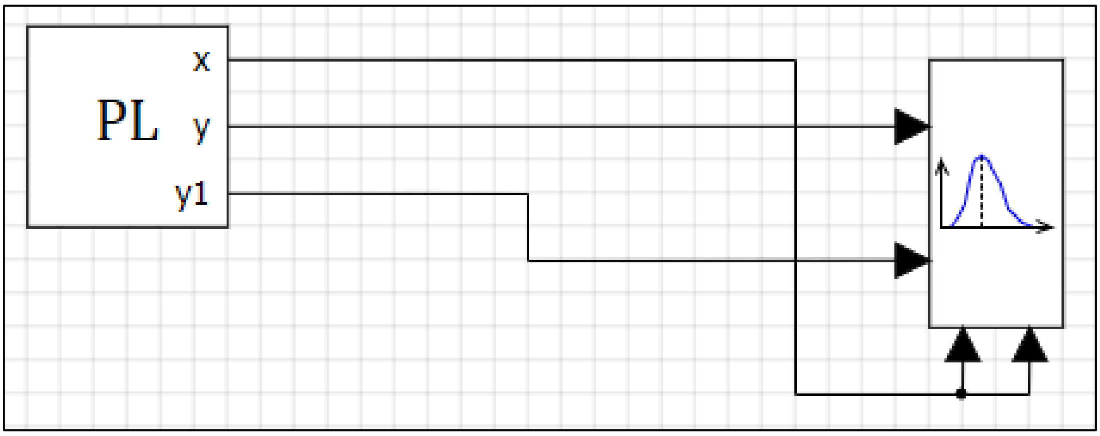


Figure 10

Save the changes in the project and run the program for execution. The simulation results are shown in Figures 11–12.

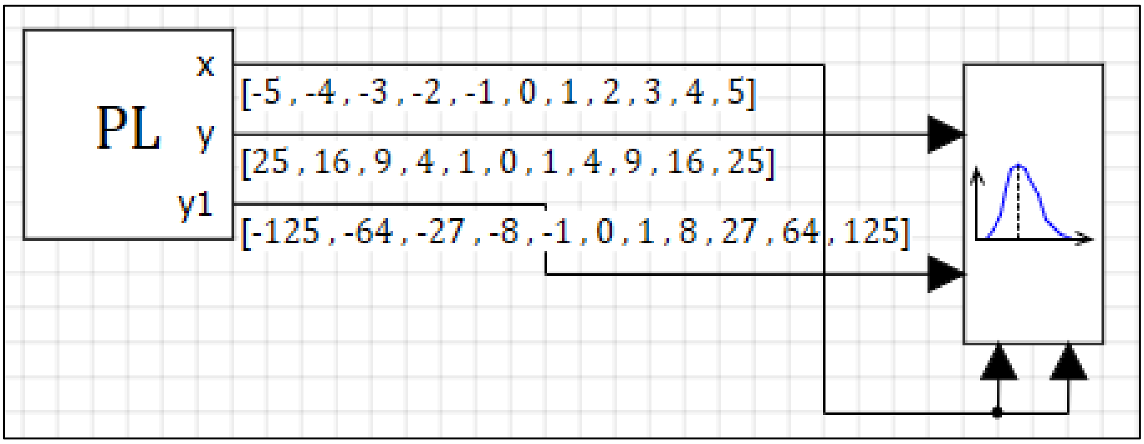


Figure 11

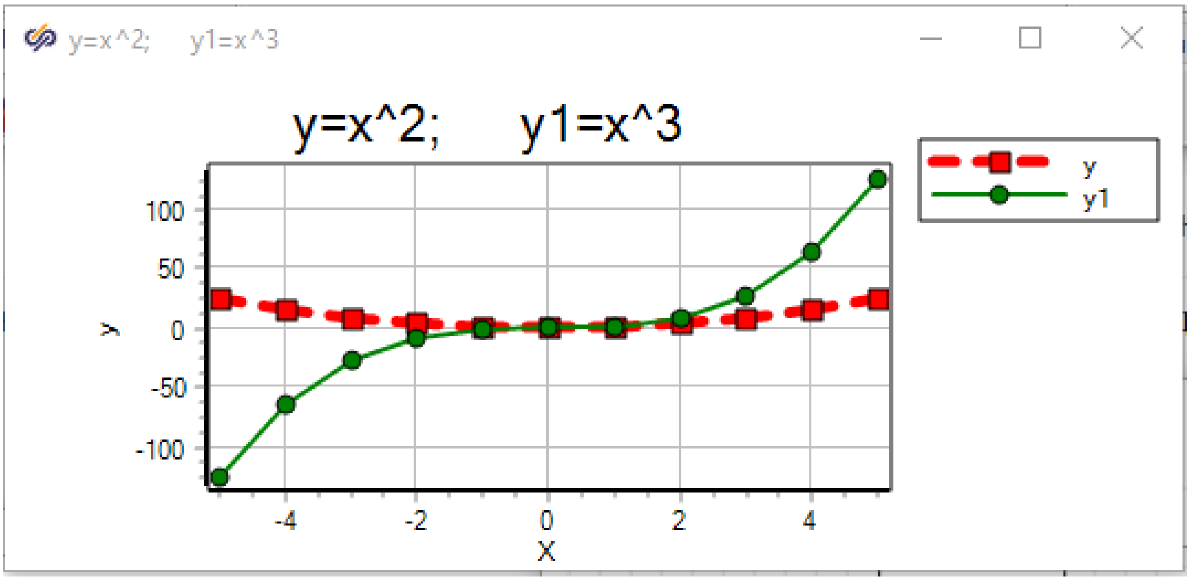


Figure 12

Edit the settings in the **Graph Properties** window in accordance with Figure 12 by changing the name of the graph, the name of the **Y**-axis, and adding a legend.

Assignment for the lab report

Basic level

Construct two graphs in the Cartesian coordinate system for a given assignment option: y=f(x) from Table 1 and y=2·f(x) in the specified range along the x axis and with a specified step.

Report

Insert into the laboratory report the wording of the tasks, the text of the programs received, as well as the results of their execution on the screen of the **Graph Y from X** block.

4. Create a new project based on the calculation scheme of the solver **Control system diagram**.

Add the **Programming Language** block from the **Dynamic** tab and the block **Graph of time** (**Временной график**) from the **Data Output** tab to the project window.

Write a program in the **Programming Language Block** dialog box:

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

In the program, the current vector **i** of zero length is set as the output variable of the **Programming Language** block.

The values 2 and 10 are assigned to the elements of the resistance vector **R**. The values of the vector **R** are included in the expression for the variable $g = \frac{R}{2L}$, which, in turn, is included in the expression for the current

$$i = \frac{E}{L \cdot w} e^{-t \cdot g} \sin(w \cdot t).$$

As a result, the values of two functions are formed in the vector **i** for each moment of time **t**, which is set in the program using the **time** system variable, which returns the model time.

Save the project in your folder as a file named **lw_4_4** and run the program for execution.

Connect the output port of the **Programming Language** block that appears to the corresponding input port of the block **Graph of time**.

Restart the program for execution and view the simulation results on the screen of the **Graph of time** (figure 13) by double-clicking on it.

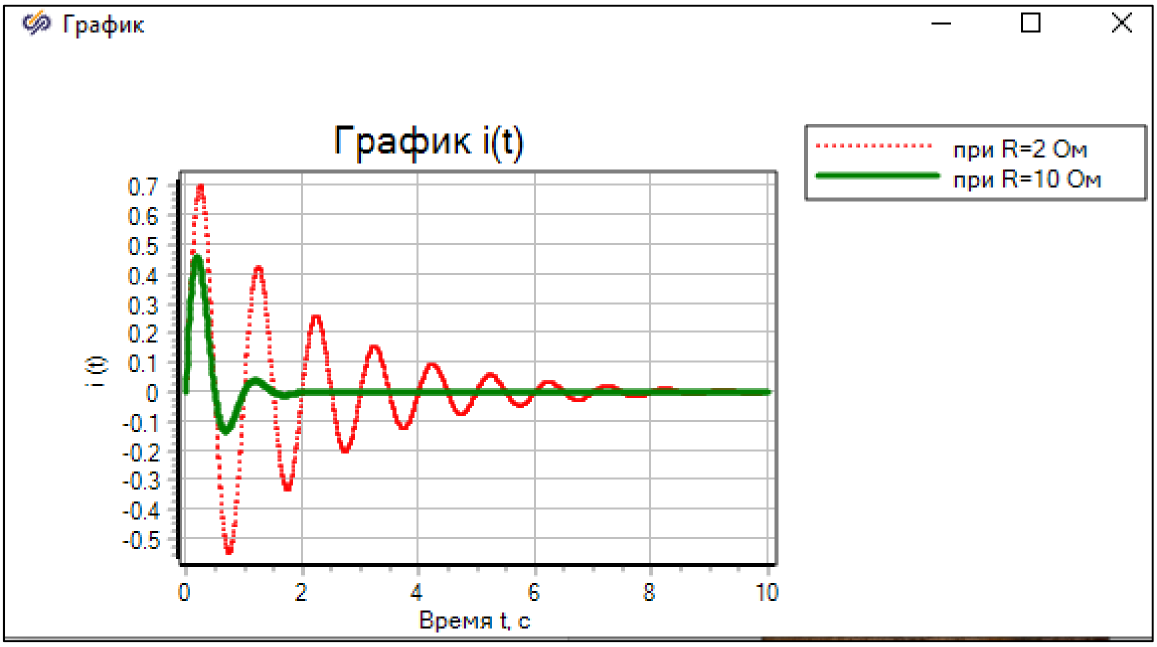


Figure 13

Figure 13 shows the final calculation time of 10 seconds. Figure 13 shows how the transient process will proceed in the RLC circuit at two resistance values of **R**. The higher the **R**, the faster the transient process will fade.

To change the final calculation time, click the **Calculation Parameters**  button of the schematic project window. As a result, the **Project Parameters dialog box** opens (Figure 14). On the **Calculation Parameters** (**Параметры расчета**) tab, you need to set the required values of the calculation parameters.

Figure 15 shows how the simulation results will change if you set the **Final Calculation Time** (**Конечное время расчета**) parameter to 5.

Assignment for the lab report

Basic level

Construct a family of curves in the Cartesian coordinate system for a given assignment option *N*

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

for N, given as a vector string [N, 2N, 3N].

Set the value of the **Final Calculation time** parameter to 5.

Report

Insert into the laboratory report the wording of the tasks, the text of the programs received, as well as the results of their execution on the screen block **Time schedule**.

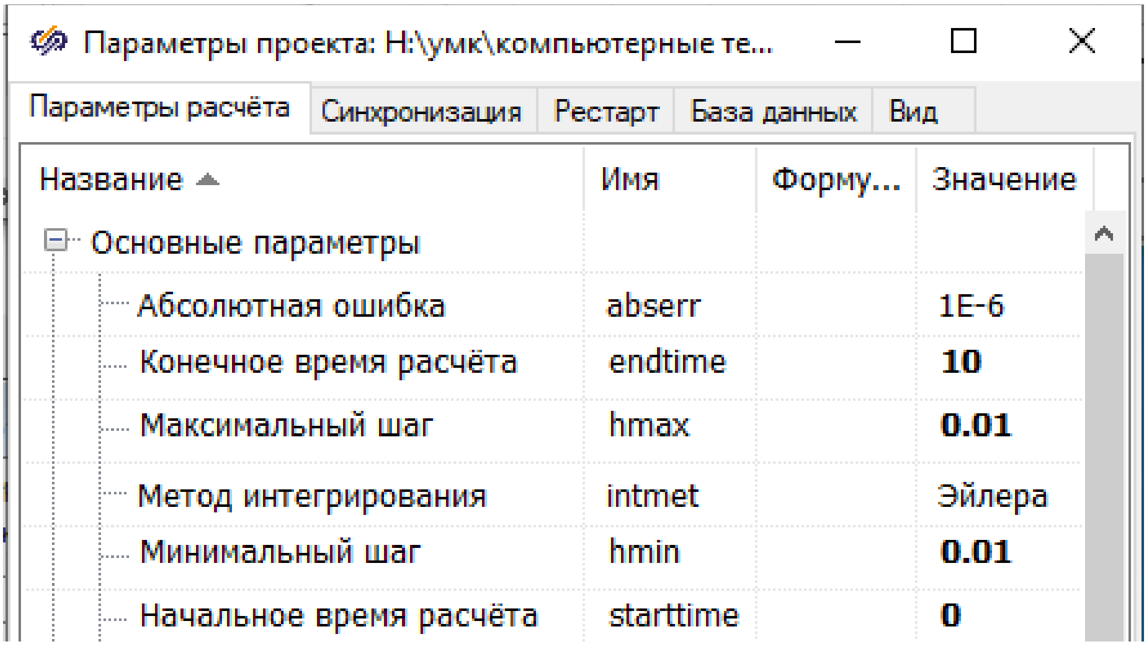


Figure 14

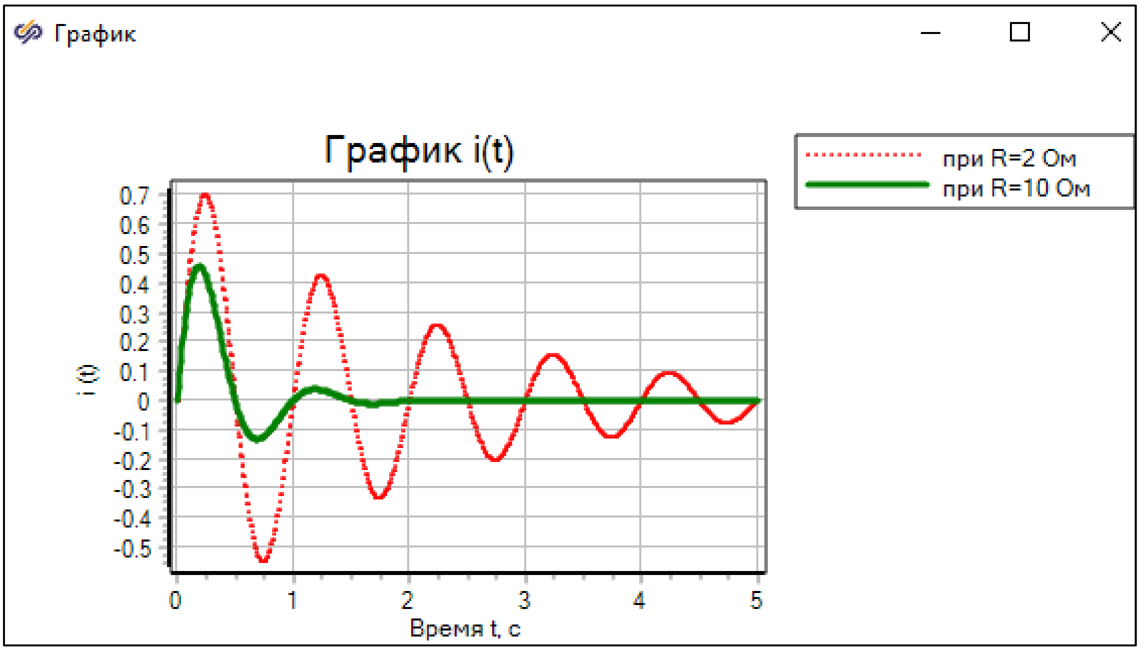


Figure 15

5. Create a new project based on the calculation scheme of the solver **Control system diagram**.

Add the **Programming Language** block from the **Dynamic** tab and the block **Graph Y from X** from the **Data Output** tab to the project window.

Set the value **3** for the **Number of graphs** property for **Graph Y from X**.

Write a program in the **Programming Language Block** dialog box:

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

The program defines as output variables of the **Programming Language** block:

– four vectors: **I1i, I2i** and **I1r, I2r** of dimension 2, which specify the imaginary and real parts of two complex numbers;

– is the dimension of the imaginary and real parts of the third vector **I3i** and **I3r**, equal to two.

In the main part of the program, vectors **I3i** and **I3r** contain the vector sum, respectively, for the imaginary and real parts of a complex number.

The real values of the vectors from the output ports of the **Programming Language** block are fed to the **X** inputs of the **Graph Y from X** block, and form the values of the graph of the function along the abscissa axis.

The imaginary values of the vectors from the output ports of the **Programming Language** block are fed to the **Y** inputs of the **Graph Y from X** block, and form the values of the graph of the function along the ordinate axis.

Save the project in your folder as a file named **lw_4_5** and run the program for execution. As a result, the blocks will have additional output ports.

Connect the output ports of the **Programming Language** block that have appeared with the corresponding input ports of the **Graph Y from X** block in accordance with Figure 16.

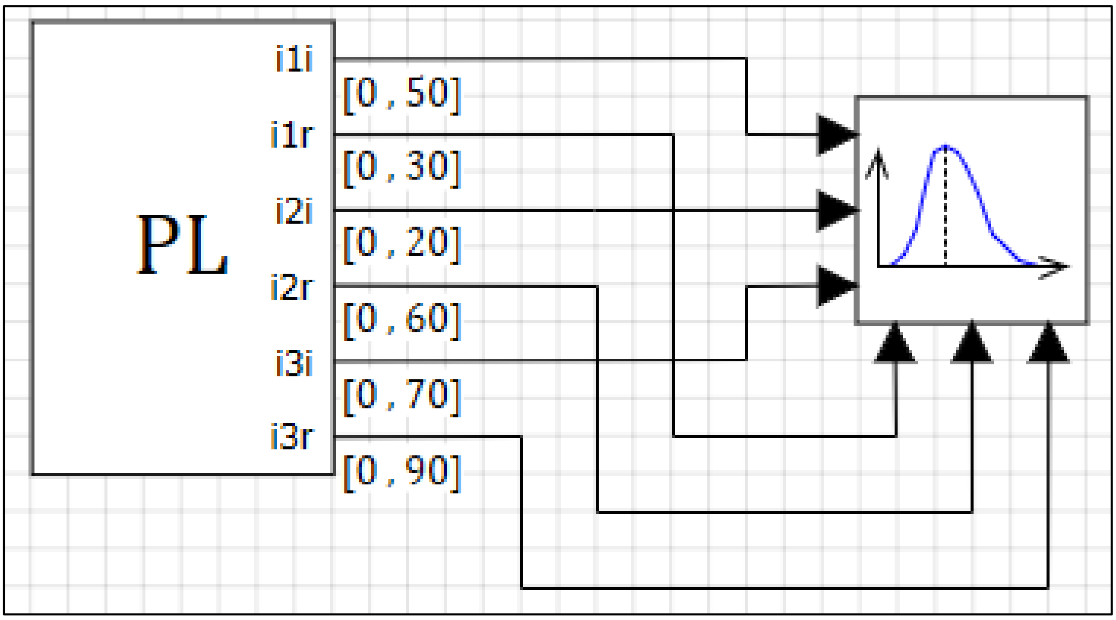


Figure 16

Figure 17 shows the results of the program execution on the screen of the **Graph Y from X** block in the form of two vectors **I1** and **I2**, and the third vector **I3** corresponds to the vector sum **I1 + I2**. In a vector diagram, all vectors go from zero.

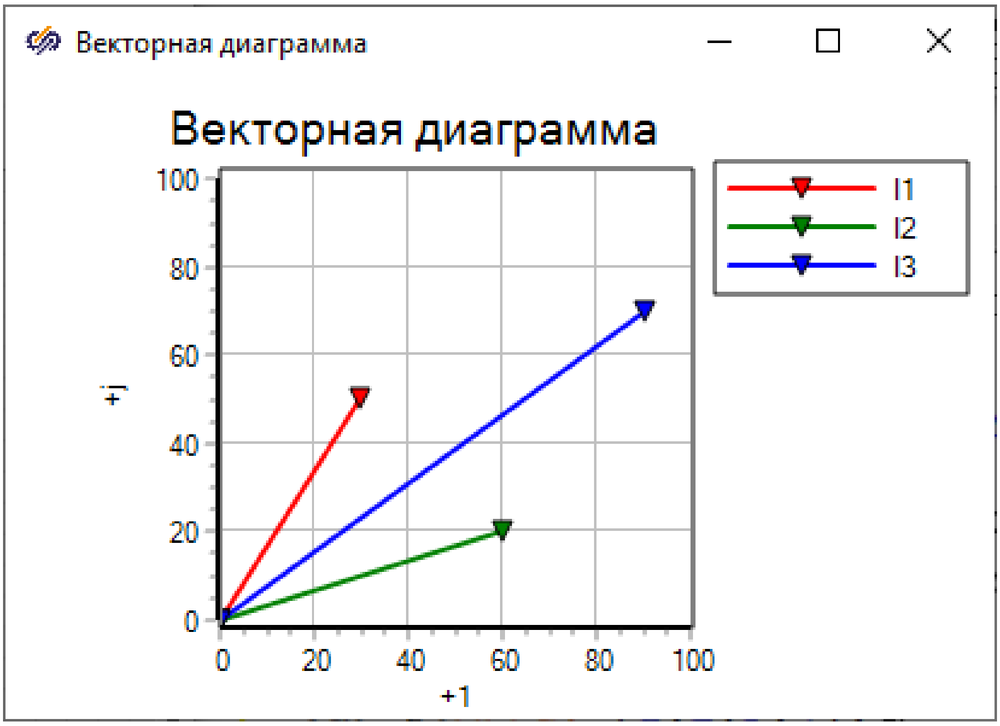


Figure 17

6. Save the previous project from item 5 in your folder as a file named **lw_4_6**.

Write a program in the **Programming Language Block** dialog box:

```
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]];
```

Run the program for execution.

Figure 18 shows the results of program execution on the output ports of the **Programming Language** block.

Figure 19 show the results of the program execution on the screen of the **Graph Y from X** block in the form of a topographic diagram, in which each subsequent vector exits from the end of the previous vector.

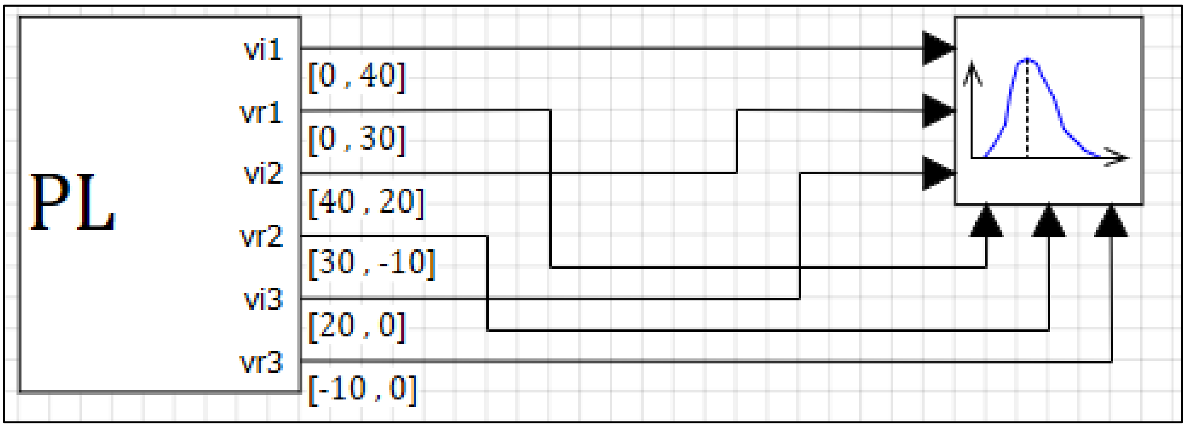


Figure 18

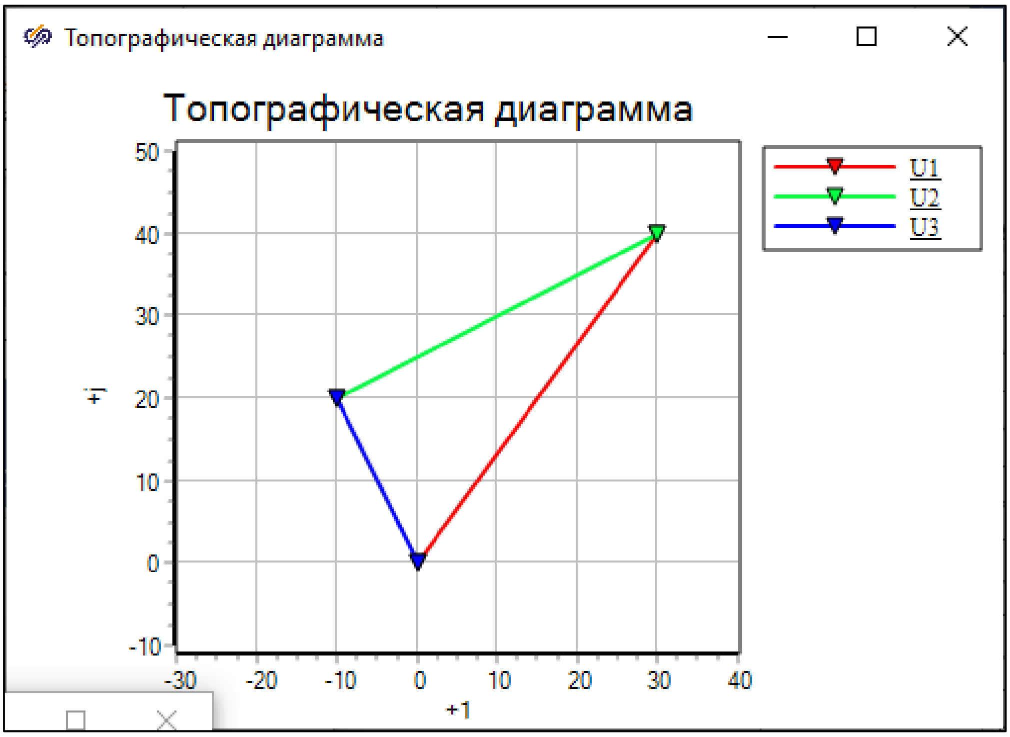


Figure 19

Assignment for the lab report

Basic level

Depending on the option number N, plot the **Graph Y from X** in the Cartesian coordinate system using the block:

– vector diagram for complex numbers

$\underline{I}1 = N + i(N+5)$, $\underline{I}2 = N + iN$; $\underline{I}3 = I1 + I2$;

Complicated level

– a topographic diagram for complex numbers $\underline{U}1 = N + I\,(N+5)$, $\underline{U}2 = (N+10) + iN$; $\underline{U}3 = \underline{U}1 + \underline{U}2$.

Report

Insert into the laboratory report the wording of the tasks, the text of the programs received, as well as the results of their execution on the screen of the **Graph Y from X** block.

Control questions

1. Which blocks are used in SimInTech to build one or more graphs in the same axes?

2. How are the various chart settings set in SimInTech?

3. How can you add a grid of coordinate lines, axis labels, a legend, and a title to graphs?

4. How can you increase the required part of the graph?

5 How can you plot multiple graphs in the same axes using the **Graph Y from X** block?

6. How can you plot multiple graphs in the same axes using the **Graph of time** block?

7. For what purposes can the **time** system variable be used in the program?

8. How can you change the final calculation time?

9. How can vector and topographic diagrams be constructed in SimInTech?

References

1. Website SimInTech.ru. URL: http://simintech.ru/

2. SimInTech Help System. URL: https://help.simintech.ru/#o_sim-intech/browsers.html

3. SimInTech technology modeling, development and creation of ACS. URL: https://miem.hse.ru/data/2018/10/29/1141996123/SimInTech.pdf

Учебное издание

Регеда Владимир Владимирович,
Регеда Ольга Николаевна

Программирование в SimInTech = Programming in SimInTech

На английском языке

Редактор *М. Н. Кузнецова*
Технический редактор *В. А. Гущин*
Компьютерная верстка *В. А. Гущина*

Подписано в печать 07.08.2025.
Формат $60\times84^{1}/_{16}$. Усл. печ. л. 5,11.
Заказ № 332. Тираж 20.

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

ПензГУ