

Автоматизация и управление технологическими процессами и производствами, системы автоматизации проектирования

Научная статья

Статья в открытом доступе

УДК 004.4.242

doi: 10.30987/2658-6436-2023-3-4-12

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ АВТОМАТНОГО ПРОГРАММИРОВАНИЯ ПРИ РАЗРАБОТКЕ ПРИЛОЖЕНИЯ ДЛЯ ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ

Павел Васильевич Гумунюк^{1✉}, Александр Николаевич Шурпо^{2✉}

^{1, 2} Институт конструкторско-технологической информатики Российской академии наук, г. Москва, Россия

¹ pavel05091997@yandex.ru

² a-shurpo@yandex.ru, <http://orcid.org/0000-0003-1962-1969>

Аннотация. В статье показано, что программирование можно освоить даже тем людям, которые ранее не были знакомы с ним. Для этого было разработано приложение, позволяющее новичкам программировать с помощью подсказок. Исходный код приложения размещен в открытом доступе, разработан с помощью технологии автоматного программирования с применением таких инструментов как компьютерная модель «словарь понятий» и явное выделение состояний. Программа будет написана в бесплатной интегрированной среде разработки Sharp Developer на основе языка C# и в тексте статьи будут приведены главные проблемы, которые были выявлены в процессе разработки программы, а также методы их решения, пояснены основные логические решения, принятые для написания кода программы. Помимо основной цели, для которой был написан код данной программы, текст статьи можно рассматривать как руководство по написанию программой логики на основе методов и инструментов автоматного программирования. Данная статья может быть полезна людям, желающим улучшить свои знания в области программирования и помимо этого она направлена на популяризацию технологии автоматного программирования с целью развития данной области разработки программной логики и упрощения ее принципов и методов для широких масс специалистов.

Ключевые слова: инструментальное средство, справочник понятий, программа справочник по программированию, Switch-технология, автомат, автоматное программирование, система управления, индустрия 4.0

Для цитирования: Гумунюк П.В., Шурпо А.Н. Использование технологии автоматного программирования при разработке приложения для программирования с помощью инструментальных средств // Автоматизация и моделирование в проектировании и управлении. 2023. №3 (21). С. 4-12. doi: 10.30987/2658-6436-2023-3-4-12.

Original article

Open Access Article

USING AUTOMATIC PROGRAMMING TECHNOLOGY IN THE APPLICATION DEVELOPMENT FOR PROGRAMMING WITH TOOLS

Pavel V. Gumunyuk^{1✉}, Alexander N. Shurpo^{2✉}

^{1, 2} Institute for Design-Technological Informatics of the Russian Academy of Sciences, Moscow, Russia

¹ pavel05091997@yandex.ru

² a-shurpo@yandex.ru, <http://orcid.org/0000-0003-1962-1969>

Abstract. *The article shows that programming can be mastered even by those who have not been previously familiar with it. For this, an application is developed allowing beginners to program using hints. The source code of the application is available in the public domain, developed with the use of automatic programming technology applying such tools as a computer model “dictionary of concepts” and explicit state selection. The programme will be written in the free Sharp Develop integrated development environment based on the C# language. The main problems identified during the programme development, as well as methods for solving them will be given in the article, the main logical decisions made to write the programme code will be explained. In addition to the main aim for which this programme code is written, the article can be considered as a manual for writing logic by the programme based on the automatic programming methods and tools. This article may be useful to people who want to improve their knowledge in the field of programming, and in addition, it is aimed at popularizing the technology of automatic programming to enlarge this area of programme logic development and simplify its principles and methods for the general public.*

Keywords: tool, concept reference book, programme, programming reference book, Switch technology, automatic machine, automatic programming, control system, industry 4.0

For citation: Gumunyuk P.V., Shurpo A.N. Using Automatic Programming Technology in the Application Development for Programming with Tools. Automation and modeling in design and management, 2023, no. 3 (21). pp. 4-12. doi: 10.30987/2658-6436-2023-3-4-12.

Введение

В современном мире постоянно растёт доля умных устройств, которые, хотя и удобны при использовании, одновременно становятся всё более сложными технически. Это обстоятельство требует увеличения количества специалистов, способных программировать данные устройства. Однако для успешного программирования требуется необходимая степень квалификации специалистов, занимающихся написанием программного кода. Switch-технология позволяет сгенерировать программный код для любого умного устройства, а единственным условием для осуществления такой операции является знание разработчиком логики данного устройства. Помимо знания принципов работы системы необходимы хотя бы минимальные знания в области программирования. Поскольку в реалиях современного мира высококвалифицированные программисты слишком загружены, процесс генерации программного кода и составление графов переходов может быть передан в руки специалистов, разбирающихся в работе умных устройств. Таким специалистам, чтобы разобраться в автоматном программировании, должна помочь программа «справочник по программированию». Для оперативного освоения нового материала и улучшения качества полученных знаний в программировании целесообразно использовать различные инструменты, а также сервисы и методики, помогающие в этом обучении [1]. Программа «справочник по программированию» может быть написана с применением технологии автоматного программирования. Для построения логики поможет компьютерная модель словаря понятий рассматриваемая в «Преобразование графов переходов, представленных в формате MSVisio, в исходные коды программ для различных языков программирования (инструментальное средство MetaAuto)» [2]. В результате выполнения данной задачи достигаются две цели, а именно: рассмотрен пример создания программы с применением автоматного программирования и создана программа, цель которой помощь начинающим программистам.

Материалы, модели, эксперименты и методы

Для написания программы по изучению программирования необходимо сначала определить требования для будущей программы, которая должна содержать следующие элементы:

– ползунок выбора языка программирования. Рассматриваются такие варианты как: C#; JavaScript; Python; ST; FBD; LD, а также язык гипертекстовой разметки HTML и примеры каскадной таблицы стилей CSS;

– ползунок выбора действия. Включает в себя следующие действия: объявления переменных и их инициализация; операторы условия; операторы цикла; объявление массивов; объявление двумерных массивов; создание классов; объектов и их инициализация; наследо-

вание классов; создание функций; создание структур; создание перечислений; работа с базами данных;

– экран считывания шаблона. Желательно сделать наглядную подсветку для распознавания различных фрагментов текста, причём текст должен иметь возможность копироваться в буфер обмена;

– перечень примеров по языкам стандартов МЭК 61131-3, который должен быть включён в ползунок выбора языка программирования, такой как FBD и LD; ползунок выбора действия нестандартный;

– перечень примеров по веб-вёрстке. Должен быть включен в ползунок выбора языка программирования, такой как HTML и CSS; ползунок выбора действий нестандартный;

– индикатор отображения версии программы;

– программа должна быть написана с применением Switch-технологии [3].

Справочная программа уместно дополнит компьютерную модель «словарь понятий», применяемую для автоматного программирования. Switch-технология является еще одним названием технологии автоматного программирования. Это связано с тем, что данная методика использует оператор множественного выбора switch для выбора устойчивого состояния, в коде ниже можно убедиться в этом [4].

Рассмотрим интерфейс программы, который должен встречать пользователя при запуске приложения. Программа состоит из окна, в котором имеется одно текстовое поле с возможностью вывода и копирования информации в буфер обмена. Кроме этого, программа имеет два ползунка: выбора языка программирования и выбора требуемого действия, а также кнопку, которая необходима для вывода нужной информации, при её нажатии (рис. 1).

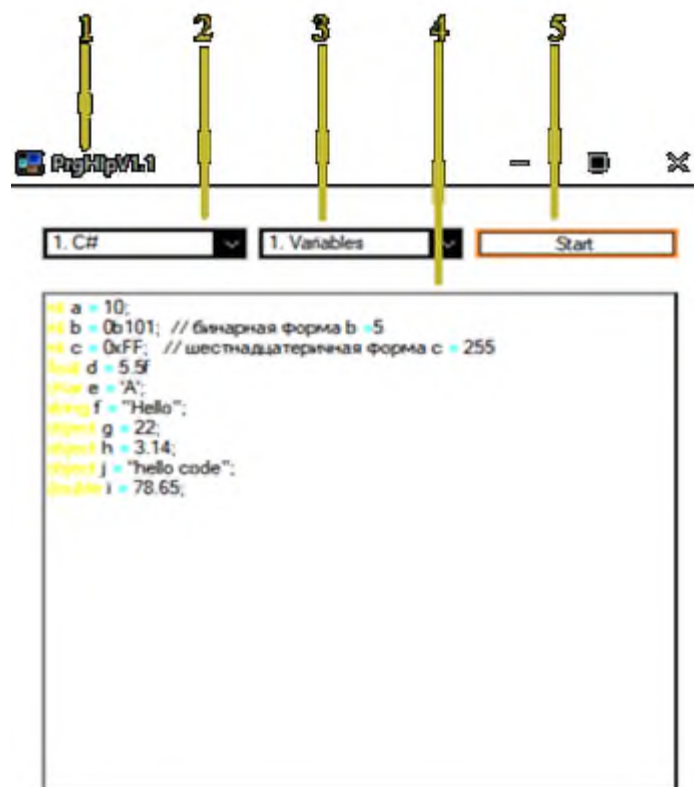


Рис. 1. Окно программы «Справочника по программированию»

1 – версия программы; 2 – ползунок выбора языка; 3 – ползунок выбора действия;

4 – текстовое поле; 5 – кнопка пуска программы

Fig. 1. «Programming helper» window

1 – program version; 2 – language selection slider; 3 – action selection slider; 4 – text field; 5 – program start button

Программа работает следующим образом. Пользователь выбирает язык, по которому он хочет получить справку, используя ползунок 2, затем действие этого языка утверждается ползунком 3 и далее этот выбор подтверждается кнопкой 5 после чего на экране 4 выводится

пример из базы данных. Пользователь может сначала выбрать действие ползунком 3, а затем язык ползунком 2, так как это для программы не очень важно. Важно то, что после подтверждения своего действия кнопкой 5 на текстовом поле 4 появляется новая справочная информация, а старая – удаляется, если поле до этого было заполнено текстом. Весь текст на экране 4 доступен для копирования в буфер обмена. При закрытии программы текстовое поле очищается, справки в программе предназначены только для уточнения синтаксиса и снабжены комментариями с пояснениями наиболее трудных моментов. Например, пользователю необходимо выяснить, как объявляются классы на языке PHP. Для этого первым ползунком выбирается пункт «4. PHP», а вторым ползунком – пункт «6. Classes», а кнопка 5 при нажатии «Start» отображает результат на табло.

Программа написана на языке C#, с использованием среды Sharp Develop (рис. 2).

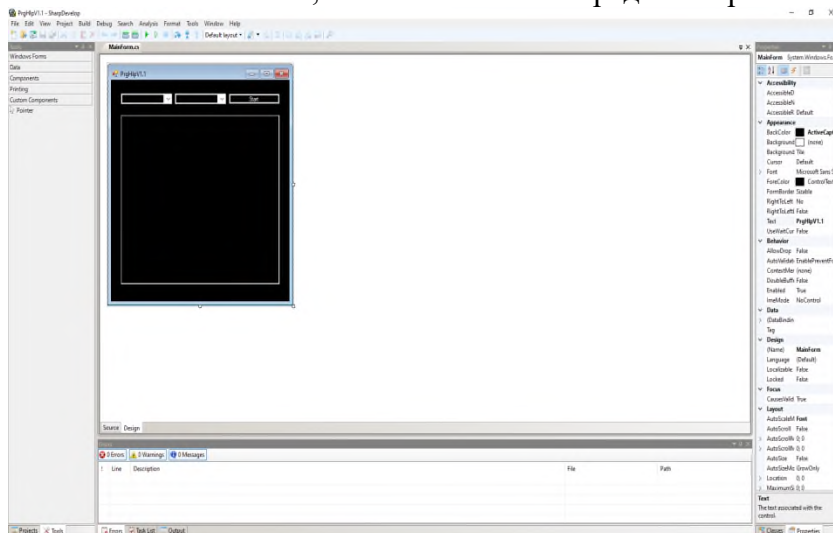


Рис. 2. Среда разработки Sharp Develop
Fig.2. Sharp develop IDE

В коде программы не будет показана логика всех состояний, так как она приблизительно однородная. С программой можно ознакомиться в репозитории.

Сначала идет объявление всех библиотек, требуемых в программе.

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
namespace PrgHlp_v1
{
    /// <summary>
    /// </summary>
    public partial class MainForm : Form
    {
        public MainForm()
        {
```

После объявления основного класса программы идет инициализация основных компонентов

```
        InitializeComponent();
        //comboBoxLang.SelectedIndexChanged += comboBoxLang_SelectedIndexChanged;
        comboBoxLang.SelectedIndex=0;
        comboBoxTitle.SelectedIndex=0;
    }
    string selectedState1;
    string selectedState2;
```

Для реализации подсветки текста необходим метод `BigPrint`, который в свою очередь будет состоять из нескольких ключевых фрагментов.

```
void BigPrint (string strBig)
{
    richTextBox.SelectedText = richTextBox.SelectedText + strBig;
    Данная строка будет отвечать за добавление текста обрабатываемого в методе с текстом, указанным в текстовом поле через элемент richTextBox.
    string[] RedCategory = { "for", "foreach", "using", "switch", "new", "public", "private", "protected", "static", "while", "do",
    "=", "<=", ">=", "!=", "+=", "-=", "=+", "++", "--",
    "case ", "break;", "default;", "+", "-", "!", "if", "else", "else if", "else\n{ ", "<", ">" };
    for (int i = 0; i < RedCategory.Length; i++)
    {
        int x = strBig.IndexOf(RedCategory[i]);
        int lenght001 = RedCategory[i].Length;
        if(x!=-1)
        {
            int pos001 = 0;
            while((pos001 = strBig.IndexOf(RedCategory[i], pos001)) != -1)
            {
                richTextBox.Select(pos001,lenght001);
                richTextBox.SelectionColor = Color.Red;
                pos001++;
            } } }
    Созданный массив RedCategory типа string позволяет добавлять в него все те ключевые слова, которые требуется подсветить красным цветом. Далее идет цикл позволяющий обработать все элементы массива. Следующей строкой происходит поиск индекса в тексте, где начинается данное слово [5]. Следующий шаг позволит определить длину искомого слова. И если индекс искомого слова не равен -1, то тогда с помощью цикла закрашиваются все искомые слова столько раз, сколько раз они встречаются в тексте [6].
    string[] BlueCategory = { "int", "namespace", "class", "void", "Random", "random", "Next", "WriteLine", "float", "string", "char", "object", "double", "Length", "ReadLine" };
    for (int i = 0; i < BlueCategory.Length; i++)
    {
        int x = strBig.IndexOf(BlueCategory[i]);
        int lenght002 = BlueCategory[i].Length;
        if(x!=-1)
        {
            int pos002 = 0;
            while((pos002 = strBig.IndexOf(BlueCategory[i], pos002)) != -1)
            {
                richTextBox.Select(pos002,lenght002);
                richTextBox.SelectionColor = Color.Blue;
                pos002++;
            } } }
    string[] GoldCategory = { "Main" };
    for (int i = 0; i < GoldCategory.Length; i++)
    {
        int x = strBig.IndexOf(GoldCategory[i]);
        int lenght003 = GoldCategory[i].Length;
        if(x!=-1)
```

```

{
    int pos003 = 0;
    while((pos003 = strBig.IndexOf(GoldCategory[i], pos003)) != -1)
    {
        richTextBox.Select(pos003, length003);
        richTextBox.SelectionColor = Color.Gold;
        pos003++;
    } } }

```

Данный код дублируется несколько раз, затем идет логика поиска требуемых фрагментов текста, которые надо закрасить.

```

void ComboBoxLanguageSelectedIndexChanged(object sender, EventArgs e)
{
    selectedState1 = comboBoxLang.SelectedItem.ToString();
    selectedState1 = selectedState1.Remove(2,11);
}

```

Ограничение на количество символов. Данная строка взята на основании самого длинного названия JavaScript. Данная строка удаляет 12 символов после первого.

```

void ComboBoxTitleSelectedIndexChanged(object sender, EventArgs e)
{
    selectedState2 = comboBoxTitle.SelectedItem.ToString();
    selectedState2 = selectedState2.Remove(2,18);
}

```

Данная строка реализует ползунок по аналогии с первым ползунком выбора языка программирования. Только в этот раз все происходит по аналогии, но с выбором действий.

Затем переменные selectedState1 и selectedState2 инициализированные еще в начале программы объединяются, и далее происходит выбор требуемого значения через оператора switch.

```

void ButtonClick(object sender, EventArgs e)
{
    string selectedFinal = selectedState1 + selectedState2;
    richTextBox.Clear();

```

Основная часть программы происходит после нажатия на кнопку. Первым происходит объединение selectedState1 и selectedState2 в переменную selectedFinal. Затем происходит очистка текстового поля, если на нём остался от предыдущей операции текст. Далее следует оператор множественного выбора, который в качестве аргумента принимает переменную selectedFinal. При выполнении каждого конкретного условия происходит вывод требуемого текста, который выводится через метод BigPrint и, если потребуется, применяются дополнительные методы.

```

    switch ( selectedFinal )
    {
        case "1.1.":
            BigPrint("int a = 10;\nint b = 0b101; // бинарная форма b = 5\nint c = 0xFF; // шестнадцатеричная форма c = 255\nfloat d = 5.5f\nchar e = 'A';\nstring f = \"Hello\";\nobject g = 22;\nobject h = 3.14;\nobject j = \"hello code\";\ndouble i = 78.65;");
            PrintComment(26,24);
            PrintComment(64,37);
            PrintGreen(125,3);
            break;
        case "1.2.":
            BigPrint("int num1 = 8;\nint num2 = 6;\nif(num1 > num2)\n{\n Console.WriteLine($\"Число {num1} больше числа {num2}\");\n}\nelse if (num1 < num2)\n{\n Console.WriteLine($\"Число {num1} меньше числа {num2}\");\n}\nelse\n{\n Console.WriteLine(\"Число num1 равно числу

```

```

num2);\n}\n\n\nConsole.WriteLine("\nНажмите Y или N\");\nstring sel =
Console.ReadLine();\nswitch (sel)\n{\ncase \"Y\":\n Console.WriteLine("\nВы нажали букву
Y\");\n break;\ncase \"N\":\n Console.WriteLine("\nВы нажали букву N\");\n break;\ndefault:\n
Console.WriteLine("\nВы нажали неизвестную букву\");\n break;\n}");
break;
case "1.3.":
BigPrint("for (int i=0; i<=Arr.Length;i++){\\nArr[i] =
random.Next(0,10);\nConsole.WriteLine(Arr[i]+\";\");}");
PrintTextGreen ("for (int i=0; i<=Arr.Length;i++){\\nArr[i] =
random.Next(0,10);\nConsole.WriteLine(Arr[i]+\";\");}");
//PrintGreen(87,3);
break;
} } } }

```

Помимо немаловажных функций, идущих в начале кода программы, основной составляющей является конструкция смены текста, выводимого на экран. Данный код реализован при помощи оператора switch и подразумевает собой различные варианты текста, получаемые при выборе двух условий ползунками, формирующими уникальное состояние [7]. Варианты выбора ползунка языка программирования и ползунка выбора действия представлены в табл. 1.

Таблица 1

Варианты ползунков выбора языка программирования и выбора действия

Table 1

Options for selecting the programming language and selecting the action sliders

Ползунок выбора языка программирования	Ползунок выбора действия
1. C#	1. Variables
2. JavaScript	2. Operators
3. Python	3. Cycles
4. PHP	4. Arrays
5. ST	5. DoubleArrays
6. FBD	6. Classes
7. LD	7. Functions
8. IL	8. Structures
9. Html	9. Enumerations
10. CSS	10. Databases
11. Git	11. SpFunctions
12. SQL	12. SpCommands

Результаты

Особенность данной программы заключается в том, что переход между устойчивыми состояниями осуществляется извне, самим пользователем данной программы, в то время как в автоматном программировании, как правило, такой переход осуществляется внутри системы управления в результате получения данных от измерительных устройств. В случае же данной программы переход к новому состоянию необходимо осуществлять по команде пользователя. В связи с этим концепция непрерывного управления автоматической системой здесь не подходит, потому что в данной системе присутствует элемент случайности, так как пользователь одновременно и управляет системой и решает сколько времени эта система будет находиться в нужном ему состоянии. Таким образом, ядром программы является переключатель между состояниями, а условиями их переключений является команда от пользователя. Число вариантов из 12 языков программирования по 12 действий для этих языков, формирует 144 состояния. Каждое состояние имеет свой текст и формулы, отвечающие за его окрас [8]. Программный код в виде блок-схемы представлен на рис. 3.

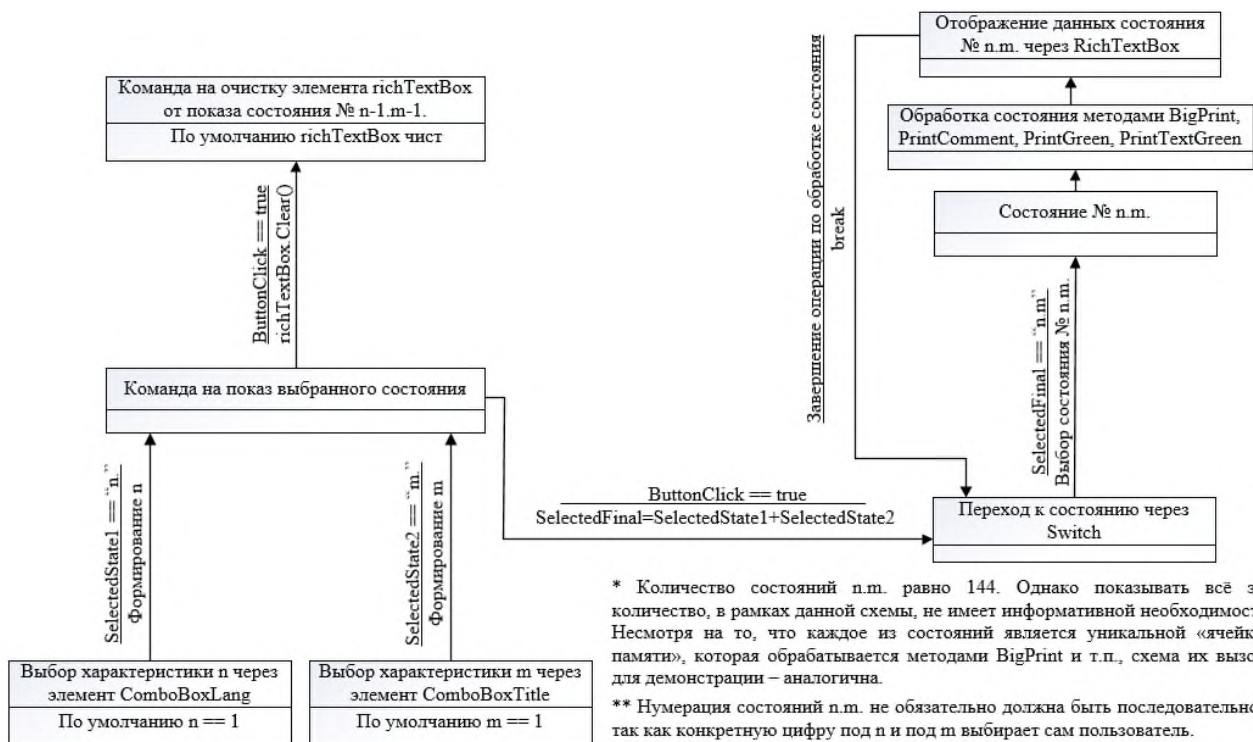


Рис. 3. Представление программного кода в виде алгоритма (блок-схемы)
 Fig. 3. Representation of program code in the form of an algorithm (block-diagram)

С программой можно ознакомиться в репозитории <https://github.com/pavel-proger/PrgHlp>.

Заключение

Значимость данной программы обусловлена возрастающей потребностью в квалифицированных программистах. Умные устройства в своей совокупности и взаимосвязанности образуют умные системы Индустрии 4.0, создание и обслуживание которых требуют быстрого обучения специалистов в своих областях [9]. Методы, используемые в данной программе, демонстрируют актуальность применения Switch-технологии для решения практических задач. При значительной длине кода она остается понятной и структурируемой [10].

Список источников:

1. Минина И.В., Петухова Т.П. О персонификации обучения школьников программированию. – Современные информационные технологии и ИТ-образование. 2018. С. 986-993.
2. Канжелев С.Ю., Шалыто А.А. Преобразование графов переходов, представленных в формате MSVisio, в исходные коды программ для различных языков программирования (инструментальное средство MetaAuto). Национальный исследовательский университет институт точной механики и оптики. – Санкт-Петербург: ИТМО. 2005. 102 с.
3. Шалыто А.А. Методы аппаратной и программной реализации алгоритмов; Национальный исследовательский университет институт точной механики и оптики. Санкт-Петербург: ИТМО. 2005. 779 с.
4. Normand V., Exertier D., “Model-driven systems engineering: SysML & the MDSysE approach at Thales”, in “Model Driven Engineering for distributed real-time embedded systems”, John Wiley & Sons, Sept. 2005; 288. ISBN 9781905209323.

References:

1. Minina I.V., Petukhova T.P. About Personification of Teaching Schoolchildren Programming. Modern Information Technologies and IT-education. 2018:986-993.
2. Kanzhelev S.Yu., Shalyto A.A. The Use of Graph Diagrams and Transition Graphs in MSVisio Format Into Program Source Codes for Various Programming Languages (MetaAuto tool). Saint Petersburg: ITMO University; 2005.
3. Shalyto A.A. Methods of Hardware and Software Implementation of Algorithms. Saint Petersburg: ITMO University; 2005.
4. Normand V, Exertier D. Model-driven Systems Engineering: SysML & the MDSysE Approach at Thales. In: Model Driven Engineering for Distributed Real-Time Embedded Systems. Wiley J. & Sons; Sept. 2005. p. 288.

5. Поликарпова Н.И., Шалыто А.А. Автоматное программирование СПбГУ ИТМО. 2008. 167 с.
6. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th Edition, Wiley, 2015; 304.
7. Татарчевский В.А. Switch – технология в задачах логического управления. Программные продукты и системы №5. 2006. С. 30-32.
8. Шалыто А.А., Туккель Н.И. Switch-технология – автоматный подход к созданию программного обеспечения «реактивных» систем – Программирование. №5. 2001. С. 45-62.
9. Липкин Е. ИНДУСТРИЯ 4.0: Умные технологии – ключевой элемент в промышленной конкуренции. М.: ООО «Остек–СМТ». 2017. 224 с.
10. Шалыто А.А. Алгоритмизация и программирование задач логического управления. СПбГУ ИТМО. 1998. 55 с.

Информация об авторах:

Гумунюк Павел Васильевич

аспирант Института конструкторско-технологической информатики Российской академии наук

Шурпо Александр Николаевич

Кандидат технических наук, старший научный сотрудник Института конструкторско-технологической информатики Российской академии наук

5. Polikarpova N.I., Shalyto A.A. Automated Programming. Saint Petersburg: ITMO University; 2008.
6. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. 4th ed. Wiley; 2015. p. 304.
7. Tatarchevskiy V.A. Switch-Technology in Logical Control Problems. Software and Systems. 2006;5:30-32.
8. Shalyto A.A., Tukkel N.I. Switch Technology: an Automated Approach to Developing Software for Reactive Systems. Programming and Computer Software. 2001;5:45-62.
9. Lipkin E. INDUSTRY 4.0: Smart Technologies are a Key Element in Industrial Competition. Moscow: Ostek-SMT LLC; 2017.
10. Shalyto A.A. Algorithmization and Programming for Logic Control Tasks. Saint Petersburg: ITMO University; 1998.

Information about the authors:

Humunyuk Pavel Vasilievich

Postgraduate Student of the Institute for Design-Technological Informatics of the Russian Academy of Sciences

Shurpo Alexander Nikolaevich

Candidate of Technical Sciences, Senior Research Fellow of the Institute for Design-Technological Informatics of the Russian Academy of Sciences

**Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации.
Contribution of the authors: the authors contributed equally to this article.**

**Авторы заявляют об отсутствии конфликта интересов.
The authors declare no conflicts of interests.**

Статья поступила в редакцию 28.04.2023; одобрена после рецензирования 09.06.2023; принята к публикации 16.06.2023.

The article was submitted 28.04.2023; approved after reviewing 09.06.2023; accepted for publication 16.06.2023.

Рецензент – Горбунов А.Н., кандидат технических наук, доцент, Брянский государственный технический университет.

Reviewer – Gorbunov A.N., Candidate of Technical Sciences, Associate Professor, Bryansk State Technical University.