

Автоматизация и управление технологическими процессами и производствами, системы автоматизации проектирования

Научная статья

Статья в открытом доступе

УДК 331.101.1

doi: 10.30987/2658-6436-2024-2-4-11

РОЛЬ СЛОВАРЯ ПОНЯТИЙ В АВТОМАТНОМ ПРОГРАММИРОВАНИИ НА ПРИМЕРЕ РАЗРАБОТКИ СИСТЕМЫ УПРАВЛЕНИЯ АВТОМАТИЗИРОВАННОГО АКВАРИУМА

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. Рассматривается вопрос применения методов технологии автоматного программирования для разработки программного обеспечения автоматизированного аквариума. Подробно разобран технологический процесс функционирования аквариума и обоснован выбор тех или иных аппаратных решений, включенных в автоматизированную систему управления аквариумом. Обоснована необходимость написания системы управления объекта (аквариума) и обоснованы способы её использования. Разобрана структура системы управления, где каждому новому фрагменту кода в данной программе для справки дается пояснение его функционала и возможностей применения, взятое из словаря понятий методики автоматного программирования. Представлен результат в виде программного кода и блок-схемы, которые переводят, сформированные на знаниях о технологическом устройстве аквариума, правила работы данной системы в язык логики. Такой способ разработки программного обеспечения является «машинным» переводом четко составленного человеком текста, являясь попыткой показать, что процесс написания программного кода представляет собой процесс перевода текста с одного языка на другой, а также то, что этот способ программирования прост в освоении и удобен в эксплуатации. При этом первый язык понятен человеку, а второй язык адаптирован для компьютера. Для устойчивой работы этой методики перевода необходим словарь, который позволяет пользователю формировать задание для компьютера.

Ключевые слова: автоматизируемый аквариум; умный контроль; switch-технология; автоматное программирование; система управления; индустрия 4.0

Для цитирования: Гумунюк П.В., Шурпо А.Н. Роль словаря понятий в автоматном программировании на примере разработки системы управления автоматизированного аквариума // Автоматизация и моделирование в проектировании и управлении. 2024. №2 (24). С. 4-11. doi: 10.30987/2658-6436-2024-2-4-11.

Original article

Open Access Article

THE ROLE OF THE CONCEPT DICTIONARY IN AUTOMA-BASED PROGRAMMING ON THE EXAMPLE OF DEVELOPING AN AUTOMATED AQUARIUM CONTROL SYSTEM

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 paper considers the issue of using automa-based programming technology methods for developing software for an automated aquarium. The technological process of operating the aquarium is analyzed in detail and the choice of certain hardware solutions included in the automated aquarium control system is substantiated. The need to write a control system for the object (aquarium) is justified and the methods of its use are proved. The control system structure is analyzed, where each new piece of code in this program is given an explanation of its functionality and application possibilities for reference, taken from the concept dictionary of automa-based programming techniques. The result is presented in the form of a program code and block diagrams that translate the rules of this system operation into the language of logic, based on the knowledge about the technological structure of the aquarium. This software development method is a “machine” translation of a clearly written human text, being an attempt to show that the process of writing a program code is a process of translating a text from one language into another, and also that this method of programming is easy to learn and convenient in operation. In this case, the first language is understandable to humans, and the second language is adapted for computers. For a stable operation of this translation technique, a dictionary is needed that allows the user to create a task for the computer.*

Keywords: automated aquarium; smart control; switch technology; automa-based programming; control system; industry 4.0.

For citation: Gumunyuk P.V., Shurpo A.N. The Role of the Concept Dictionary in Automa-Based Programming on the Example of Developing an Automated Aquarium Control System. Automation and modeling in design and management, 2024, no. 2 (24). pp. 4-11. doi: 10.30987/2658-6436-2024-2-4-11.

Введение

Автоматизированный аквариум может быть использован как для разведения рыбы и других, обитающих в воде существ (раков, крабов, креветок, устриц и им подобных), с целью получения икры, молодняка или рыбной продукции в рыболовных хозяйствах, так и для применения в домашних условиях [1].

Воссоздание климата в искусственной водной среде является трудоемким процессом, требующим слаженной работы определенного оборудования. Сложность обеспечения необходимых условий зачастую зависит от чувствительности будущих обитателей, что влияет на их дальнейшее комфортное и здоровое существование и развитие.

Материалы, модели, эксперименты и методы

Среди технических средств, обеспечивающих нормальные условия для роста обитателей аквариума, можно выделить следующие:

- фильтр очищает аквариумную воду от продуктов жизнедеятельности и подлежит замене через определенные промежутки времени;
- датчик температуры контролирует температуру воды;
- первый насос – снабжает воду кислородом для дыхания обитающих в воде существ;
- второй насос – обеспечивает циркуляцию воды.

Среди технологических параметров воды, которые необходимо постоянно контролировать выделим основные – это температура, насыщение кислородом и её очистка от отходов [2].

Технологическая схема аквариума представляет собой резервуар с системой труб для обеспечения циркуляции воды с помощью насоса. Труба, забирающая воду из аквариума, имеет систему фильтров для очистки воды от имеющихся в ней твердых частиц и вредных химических соединений, образующихся в результате жизнедеятельности обитателей резервуара с водой. Фильтрация необходима ранее всех остальных процессов во избежание засорения приборов, находящихся далее по технологической цепи. Далее расположен клапан для набора воды, который открывается при сигнале поплавкового датчика уровня в аквариуме о недостаточном количестве воды. Следующим звеном является водонагревательный элемент для поддержания температуры воды в комфортном для обитателей аквариума диапазоне. Самым важным агрегатом в технологической цепи является центробежный насос, который обеспечивает циркуляцию водной массы во всем аквариуме.

Для данной схемы выбран менее мощный центробежный насос по сравнению с поршневым. Поршневые насосы имеют свойство прорывать трубы при образовании засоров в трубах от накопления твердых частиц в жидкости. Из-за наличия в воде аквариума различных органических твердых остатков возможно образование засоров в трубах, несмотря на наличие фильтра в самом начале циркуляционного узла. При образовании засора центробеж-

ный насос просто заглохнет и вода перестанет циркулировать, в то время как поршневой насос продолжит работать, что приведет к нарастанию давления в трубах и к поломке системы очистки, насыщения кислородом и нагрева воды. Поэтому менее мощный центробежный насос вызовет меньшую проблему [3].

Наличие насоса помогает растворять в воде кислород, необходимый для дыхания живущих в резервуаре существ. Как известно, при нагревании воды, кислород вытесняется из нее, но в вопросах данной задачи потери его будут невелики из-за слишком малого нагрева воды.

Ознакомившись с техническими средствами, регулирующими параметры системы, можно выделить устойчивые состояния для обеспечения её работы, которые необходимы для умного контроля этой автоматизированной системы.

Рассмотрим все состояния системы [4].

1. Загрязнился фильтр.
2. Перестал циркулировать насос.
3. Упала температура – включить нагрев.
4. Повысилась температура – выключить нагрев.
5. Добрать извне воды.
6. Слив воды.

Рассмотрим роль каждого состояния в технологическом процессе.

1. Датчики засорения фильтра в данной системе не предусмотрены из-за технологической сложности их создания. Определить, что срок использования фильтра подошел к концу можно при помощи таймера, который будет вести отчет от момента его установки, до указанного производителем времени его работы. Если же приобретать фильтры с индикаторами загрязненности то, во-первых, такие фильтры стоят гораздо дороже обычных, а, во-вторых, такой индикатор будет сложно включить в автоматизированную систему, а наблюдать за его состоянием придется визуальным образом, поэтому данный вариант не подходит из-за большой цены и малой пользы [5].

2. При нормальной работе насос циркулирует постоянно. Остановка насоса допустима только для замены фильтра, во всех прочих случаях – это поломка автоматизированной системы.

3. Резервуар аквариума оснащен температурным датчиком, который реагирует на изменение температуры и включает или выключает нагревательный элемент для поддержания требуемых условий.

4. При повышении температуры необходимо выключить нагрев, так как нет никаких источников, способных извне нагреть воду. Единственным способом изменения температуры воды является нагревательный элемент.

5. Аквариум оснащен поплавковым датчиком, который сигнализирует об изменении уровня жидкости в резервуаре и далее по технологической цепи перед центробежным насосом установлен кран для набора новой и слива старой воды при ее сильном загрязнении.

6. Слив осуществляется по показанию таймера, так же как и замена фильтра. Со временем в воде будут накапливаться вещества, которые нельзя отфильтровать. Эти вещества являются остатками жизнедеятельности аквариумных обитателей, а также сторонние продукты, которые попадают в аквариум извне и наносят вред рыбам и другой фауне в емкости. Слив части воды с вредными веществами и набор новой воды помогает снизить концентрацию опасных химических и органических соединений в ней и позволяет комфортно существовать обитателям аквариума [6].

На рис. 1. представлена схема данного аквариума.

Для корректного составления программы необходимо изучить требуемый технологический процесс и попытаться разделить его этапы на так называемые устойчивые состояния. Под устойчивым состоянием подразумевается поведение системы, которое длится определенный промежуток времени и характеризуется определенным значением переменных этой системы (физических, программных, аппаратно-реализуемых различными техническими устройствами).

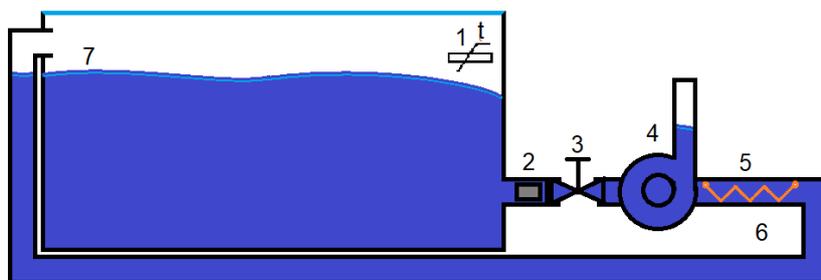


Рис. 1. Схематическое изображение автоматизированного аквариума

1 – температурный датчик; 2 – фильтр; 3 – клапан; 4 – центробежный насос; 5 – нагревательный элемент;
6 – система труб; 7 – резервуар с водой

Fig. 1. Schematic illustration of an automated aquarium

1 – temperature sensor; 2 – filter; 3 – valve; 4 – centrifugal pump; 5 – heating element; 6 – pipe system; 7 – water tank

Ядро программы будет иметь оператора switch (второе название метода автоматного программирования – это свитч-технология. Оператор switch применяется в любой программе, написанной данным методом. В языках, не имеющих оператора switch, применяется конструкция if, if/else).

Оператор switch необходим, так как только он способен осуществлять перебор состояний системы.

Состояние обозначается следующим образом. Вверху идет комментарий с номером состояния и его названием. Второй строкой идет обозначение автоматов, которые инициализируются в теле данного состояния. Последней строкой обозначается выполнение определенных внутри данного состояния действий, переменных и указания к их выполнению. Схематическое изображение устойчивого состояния представлено на рис. 2.

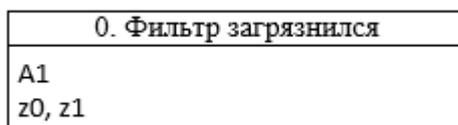


Рис. 2. Схематическое изображение устойчивого состояния

0 – начальное (первое) состояние систем; A1 – обращение к автомату, который следит за временем; z0, z1 – выполнение команды центробежным насосом, обозначается как Filter = false, pumpState = true

Fig. 2. Steady State Schematic Representation

0 – initial (first) state of systems; A1 – access to the machine that keeps track of time; z0, z1 – execution of a command by a centrifugal pump, designated as Filter = false, pumpState = true

Переход между состояниями графически обозначается стрелками. На стрелке указывается значение e вверху и z внизу; e обозначает переменную, инициализация которой приведет к переходу в другое состояние. Эта переменная исполняет роль ключа, дающего разрешение на переход. Символ z обозначает то же самое, что он обозначает внутри состояния.

В переходе между состояниями при необходимости может происходить выполнение определенных действий, переменных и указания к их выполнению, в состоянии это происходило безусловно, а здесь во время перехода при требуемой инициализации переменной e . Схематическое изображение перехода между состояниями представлено на рис. 3.

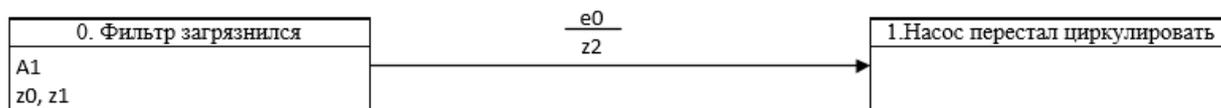


Рис. 3. Схематическое изображение перехода между состояниями

Fig. 3. Schematic representation of the transition between states

Где $e1$ – условие перехода во второе состояние, $z2$ – действие на переход в другое состояние или действие, выполняемое при переходе в другое состояние, обозначается как:

If (pumpState = true) AkvariumState = 1.

Переход между различными состояниями представляет собой фрагмент case конструкции switch. Как известно, переключение между состояниями в операторе switch может про-

исходить неограниченное количество раз и количество элементов case может быть так же неограниченным. Такие принципы идеально подходят устойчивым состояниям, которые получаются путем перевода формального языка правил исследуемой системы в свод законов ее функционирования, именуемые как состояния [7].

Словарь понятий представляет собой набор правил, описывающих принципы составления графов переходов, методики автоматного программирования. Более подробно с ним можно ознакомиться в репозитории.

Словарь понятий отражает одну из моделей взаимодействия методики автоматного программирования с существующей базой информации по технологическому объекту, для которого применяется технология автоматного программирования.

Определимся с переменными системы управления аквариума:

```
Int AkvariumState; // Состояние аквариума.
```

```
bool Filter = true; // Состояние фильтра, где true – фильтр чист, а false – следует заменить фильтр (срабатывает по таймеру).
```

```
bool pumpState = true; // Состояние насоса равно true при нормальной работе, значение false свидетельствует об аварии. Единственным случаем, когда насос может отключаться это намеренная остановка для смены фильтра и очистки аквариума. Насос снабжен энкодером, который фиксирует скорость его вращения. Эта скорость всегда одинакова, поэтому датчик необходим самый примитивный, свидетельствующий только о том, что скорость вращения равна скорости, указанной в технических характеристиках изделия. Если энкодер указывает на отсутствие вращения лопастей насоса, то это свидетельствует о его поломке.
```

```
float tempAkvarium; // Температура аквариума, если она становится меньше значения  $X$  (выбирается индивидуально для каждого вида аквариумных обитателей) понижение или повышение данного значения на величину  $Y$  свидетельствует о необходимости включить или выключить нагревательный элемент.
```

```
float waterLevel; // Уровень воды в аквариуме, при снижении установленного уровня  $Z$  на  $W$  единиц, подается сигнал на кран набора воды. Также слив и набор воды происходит периодически по истечению интервала  $T$ . Сначала происходит слив  $Q$  единиц, затем через промежуток  $t$  набор новой воды равный слитому количеству.
```

```
bool valveOn = false; // Команда для управления клапаном, где true – команда на открытие, false – на закрытие.
```

```
bool heaterOn = false; // Команда для нагрева воды, где true – команда на включение нагрева, false – на отключение.
```

Код написан на языке C# подробнее с ним и с примером словаря понятий можно ознакомиться в репозитории <https://github.com/pavel-proger/Akvarium>.

Систему целесообразно разделить на два автомата. Первый автомат $A0$ работает с основной логикой системы и имеет разветвленный программный код, который будет приведен ниже. Второй автомат $A1$ необходим для подсчета времени, необходимого для смены фильтра в аквариуме.

```
Timer = New.Timer ();
```

```
Filter = Timer;
```

Фрагмент программного кода представлен ниже:

```
Switch AkvariumState
```

```
Case 0 // Фильтр загрязнился.
```

```
Filter = false;
```

```
pumpState = true;
```

```
If( pumpState = true ) AkvariumState = 1;
```

```
If( tempAkvarium < X ) AkvariumState = 2;
```

```
If( tempAkvarium > X ) AkvariumState = 3;
```

```
If( waterLevel < Z ) AkvariumState = 4;
```

```
If( waterLevel > Z ) AkvariumState = 5;
```

```
Case 1 // Насос перестал циркулировать.
```

```
If( Filter = true ) AkvariumState = 0;
```

```
If( tempAkvarium < X ) AkvariumState = 2;
```

```
If( tempAkvarium > X ) AkvariumState = 3;
```

```

If( waterLevel < Z ) AkvariumState = 4;
If( waterLevel > Z ) AkvariumState = 5;
pumpState = false;
Case 2 // Температура упала включить нагрев.
If( Filter = true ) AkvariumState = 0;
If( pumpState = true ) AkvariumState = 1;
If( tempAkvarium > X ) AkvariumState = 3;
If( waterLevel < Z ) AkvariumState = 4;
If( waterLevel > Z ) AkvariumState = 5;
heaterOn = true;
Case 3 // Температура повысилась выключить нагрев.
If( Filter = true ) AkvariumState = 0;
If( pumpState = true ) AkvariumState = 1;
If( tempAkvarium < X ) AkvariumState = 2;
If( waterLevel < Z ) AkvariumState = 4;
If( waterLevel > Z ) AkvariumState = 5;
heaterOn = false;
Case 4 // Донабор воды из вне.
If( Filter = true ) AkvariumState = 0;
If( pumpState = true ) AkvariumState = 1;
If( tempAkvarium < X ) AkvariumState = 2;
If( tempAkvarium > X ) AkvariumState = 3;
If( waterLevel > Z ) AkvariumState = 5;
valveOn = true;
Case 5 // Слив воды.
If( Filter = true ) AkvariumState = 0;
If( pumpState = true ) AkvariumState = 1;
If( tempAkvarium < X ) AkvariumState = 2;
If( tempAkvarium > X ) AkvariumState = 3;
If( waterLevel < Z ) AkvariumState = 4;
valveOn = false;

```

Программный код в виде блок-схемы представлен на рис. 4.

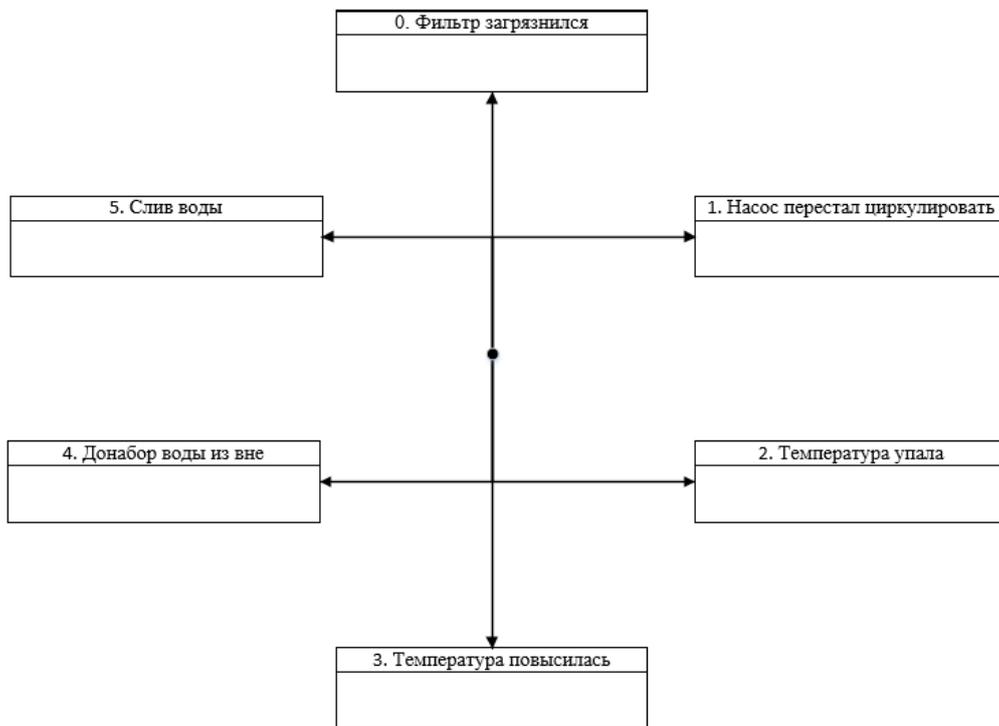


Рис. 4. Представление программного кода в виде алгоритма (блок-схемы)
Fig. 4. Representation of program code in the form of an algorithm (block-diagram)

Результаты

Если взглянуть на полученную схему графов переходов, то можно заметить, что каждое состояние это графическое обозначение состояний case; стрелки между состояниями это взаимосвязи состояний из какого в какое оно может переходить и какие переходят в него. Дроби на стрелках имеют вверху условия и внизу последствия переходов между состояниями. Взяв за основу написание технологического процесса в виде совокупности правил и переведя эти правила в совокупность состояний, можно разработать систему управления, обладая минимальными знаниями в программировании [8].

Таким образом процесс разработки системы управления будет представлять собой конвертацию информации из одного представления в другое. Разработка программного обеспечения на базе такой методики из процесса построения принципиально новой совокупности правил работы уже известной системы, реализуемой последовательностью логических конструкций превращается в преобразование существующей и проверенной информации в язык понятный вычислительным средствам системы. Преобразование, передача и чтение информации были базовыми понятиями развития любой информационной деятельности, но для осуществления обработки информации необходим понятный язык и набор правил для работы с ней. Так как при помощи простого и понятного словаря можно распространить язык и сделать его популярным, а в век информационных технологий именно владение информацией дает человеку как контроль над ситуацией, так и возможность дальнейшего развития [9].

Заключение

Автоматное программирование, как методика, содержит в себе совокупность инструментов для ее функционирования, а именно: компьютерные модели, входящие в структуру этой системы. Словарь понятий это модель, направленная на развитие автоматного программирования путем его популяризации и обучения его азам, реализованная в виде совокупности правил составления графов переходов – графического и символического представления программного кода и логики свитч-технологии. Словарь понятий как компьютерная модель позволяет как расширять методы автоматного программирования, так и популяризовать эти методы [10].

Автоматное программирование, обладая простотой и универсальностью, позволяет, при ее использовании, упростить и ускорить процесс разработки систем управления для различного рода умных устройств. Автоматное программирование, как методика, в первую очередь, рассчитана на производственных специалистов, имея целью уменьшение временных и материальных затрат на разработку систем управления, однако эта методика так же подойдет и любым желающим расширить свои знания в программировании. Создание программных продуктов для умных устройств методами автоматного программирования направлено на развитие Индустрии 4.0. и непрерывно связаного с этим понятием – понятие «Века информационных технологий».

Список источников:

1. Махлин М.Д., Аквариумный сад – М.: Природа, 1998. – 210 с.
2. Барей К.Т., Васильева Н.А. и др. Комнатный аквариум: Под ред. Пешкова М.А. – 3-е изд. – Алмата: Кайнар, 1965. – 180 с.
3. 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.

References:

1. Makhlin M.D. Aquarium Garden. Moscow. Priroda; 1998.
2. Barey KT, Vasilyeva NA, et al, Peshkova MA, editor. Indoor Aquarium. 3rd ed. Alma-Ata: Kainar; 1965.
3. Kanzhelev S.Yu., Shalyto A.A. Conversing Transition Graphs Presented in MSVisio Format Into Source Codes of Programs for Various Programming Languages (MetaAuto Tool). National Research University Institute of Precision Mechanics and Optics. Saint Petersburg: ITMO; 2005.

4. Шалыто А.А. Методы аппаратной и программной реализации алгоритмов / Национальный исследовательский университет институт точной механики и оптики. – Санкт-Петербург: ИТМО, 2005. – 779 с.

5. 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

6. Татарчевский В.А. Switch – технология в задачах логического управления // Программные продукты и системы №5. – 2006. – С. 30-32.

7. Шалыто А.А., Туккель Н.И. Switch-технология – автоматный подход к созданию программного обеспечения «реактивных» систем // Программирование. – №5. – 2001. – С. 45-62.

8. Липкин Е. ИНДУСТРИЯ 4.0: Умные технологии – ключевой элемент в промышленной конкуренции. – М.: ООО «Остек-СМТ», 2017. – 224 с.

9. Шалыто А.А. Алгоритмизация и программирование задач логического управления. – СПбГУ ИТМО, – 1998. – 55 с.

10. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. Учебно-методическое пособие. – СПбГУ ИТМО, – 2007. – 108 с.

4. Shalyto A.A. Methods of Hardware and Software Implementation of Algorithms. National Research University Institute of Precision Mechanics and Optics. Saint Petersburg: ITMO; 2005.

5. 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.

6. Tatarchevsky V.A. Switch – Technology in Logical Control Problems. Software Products and Systems. 2006;5:30-32.

7. Shalyto A.A., Tukkel N.I. Switch Technology. An Automatic Approach to Developing Software for Reactive Systems. Programming. 2001;5:45-62.

8. Lipkin E. INDUSTRY 4.0: Smart Technologies are a Key Element in Industrial Competition. Moscow: Ostek-SMT; 2017.

9. Shalyto A.A. Algorithmization and Programming of Logical Control Problems. ITMO University; 1998.

10. Polikarpova N.I., Shalyto A.A. Automatic Programming. ITMO University; 2007.

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, ORCID ID: 0000-0003-1962-1969

Information about the authors:

Gumunyuk Pavel Vasilievich

postgraduate student of the Institute for Design-Technological Informatics of the Russian Academy of Sciences

Shurpo Alexander Nikolayevich

Candidate of Technical Sciences, Senior Researcher of the Institute for Design-Technological Informatics of the Russian Academy of Sciences, ORCID ID: 0000-0003-1962-1969

**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 09.04.2024; одобрена после рецензирования 25.04.2024; принята к публикации 03.05.2024.

The article was submitted 09.04.2024; approved after reviewing 25.04.2024; accepted for publication 03.05.2024.

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

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