

УДК 621.165

doi: 10.20998/2078-774X.2016.08.02

Ю. Н. ГОВОРУЩЕНКО**ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ИТЕРАЦИОННО-РЕКУРСИВНЫЙ АЛГОРИТМ МОДЕЛИРОВАНИЯ ТЕРМОГИДРАВЛИЧЕСКИХ СИСТЕМ**

АННОТАЦИЯ Приведено описание объектно-ориентированного итерационно-рекурсивного алгоритма моделирования термогидравлических систем, предназначенного для решения широкого круга практических задач и отличающегося эффективностью, гибкостью, расширяемостью и переносимостью. На некоторых примерах продемонстрированы преимущества объектного подхода к анализу термодинамических циклов различного типа. Описаны особенности программной реализации с использованием алгоритмического языка общего назначения и показана целесообразность интеграции с графической средой для анализа термогидравлических систем.

Ключевые слова: объектный подход, моделирование, термогидравлические системы.

Yu. N. GOVORUSHCHENKO**OBJECT-ORIENTED ITERATIVE-RECURSIVE ALGORITHM FOR THERMAL-HYDRAULIC SYSTEMS SIMULATION**

ABSTRACT The description of the object-oriented iterative-recursive algorithm for thermal-hydraulic systems simulation is presented, which intended for wide range of practical problems and characterized by effectiveness, flexibility, extensibility and portability. Some examples of advantages of the object-oriented approach (encapsulation, inheritance, composition etc.) to the analysis of thermodynamic cycles are demonstrated, in particular, the possibility of the turbomachine's detailed flow path aerodynamic calculation in common with the cycle's simulation is justified. The features of the software implementation using a general purpose algorithmic language are described and the expediency of integration of the algorithm with a graphical environment for the thermal-hydraulic systems analysis is shown.

Key words: object approach, simulation, thermal-hydraulic systems.

Введение

Современные программные комплексы моделирования термодинамических циклов (тепловых схем, термогидравлических систем) основываются на объектно-ориентированном подходе [1, 2]. Последний в сочетании с традиционным использованием моделей базовых компонентов [3, 4] как объектов, а также алгоритмов решения балансовых уравнений предполагает наличие специализированного языка для поддержки:

– *инкапсуляции*, т.е. представления компонента в виде совокупности данных и присущих ему действий – методов;

– *наследования* для организации описания компонентов в виде иерархии;

– *композиции*, т.е. создания новых типов компонентов из набора существующих;

– *делегирования (агрегирования)* в смысле переназначения компоненту определённой функциональности другого компонента.

Специализированный язык для расчёта циклов обычно базируется на одном из алгоритмических языков общего назначения, способном поддерживать перечисленные выше особенности. Более простой базовый язык является предпочтительным, поскольку его компилятор (или интерпретатор) становятся составной частью программного комплекса расчёта циклов. Объектно-ориентированный подход предъявляет специальные требования также и к алгоритму расчёта, главными из которых является открытость для

добавления новых элементов и изменения функциональности уже существующих в соответствии с перечисленными выше требованиями, а также гибкость в организации различных вариантов моделирования рассматриваемого объекта (проектирование и анализ, оптимизация параметров и проч.).

В данной статье предлагается один из возможных сценариев разработки программного комплекса расчёта термодинамических циклов, предназначенного для решения широкого круга практических задач.

Цель работы

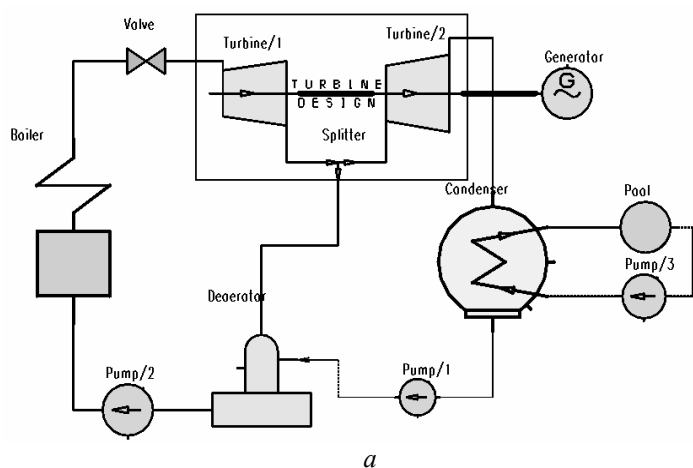
Целью статьи является изложение основного элемента системы – предлагаемого *алгоритма расчёта* на базе языка процедурного описания тепловых схем.

Алгоритм расчета

Для понимания работы алгоритма рассмотрим пример (рис. 1). На рис. 1а изображена структура парового цикла, включающая компоненты, подписанные и пронумерованные на схеме. Типам элементов в системе присвоены мнемонические имена, например, *turb*, *cond*, *pump* и т.д. Методы расчёта компонентов, имеют те же имена, но с подчеркиванием в конце (*turb_*, *cond_*, *pump_*, ...). Компоненты снабжены пронумерованными узлами (полюсами, портами) к которым подсоединяются

связи, объединяющие элементы в схему, и физически моделирующие потоки вещества и энергии. Процедура описания связей обычно называется *link_* и имеет два параметра, обозначающих выходной и входной узлы смежных элементов. Функция *lsplit_* является методом специального элемента – сплиттера, разделяющего поток на два в заданной пропорции. Как видно из кода, описывающего данную схему в режиме проектирования, когда расходы пара через турбины считаются заданными (рис. 1б), программа состоит, в основном, из пар вызовов процедур описания связи и расчёта элемента, в которую она входит. Для начала программирования цикл разрывается на выходе одного из элементов (в данном случае клапана за котлом) и составляется цепочка вызовов по ходу основного потока до тех пор, пока не будет достигнут начальный элемент.

Как видно, код программы, хотя и выглядит описательным, одновременно задаёт и топологию



схемы и последовательность выполнения расчётов (в отличие от языка *DNA* [5]). Будучи дополненной данными по параметрам элементов, этой информации оказывается достаточно для выполнения теплового расчёта в соответствии с алгоритмом, реализованным в ядре системы. Сам же алгоритм оказывается очень простым, поскольку заключается в итерационном выполнении кода (рис. 1б) до тех пор, пока не достигается с заданной точностью баланс схемы. Опыт расчётов показывает, что количество необходимых итераций на единицу больше числа регенеративных отборов и в данном случае равно двум. В простых схемах (без дополнительных циклов) для расчёта достаточно одной итерации, если только не требуется уточнение параметров элементов, например, КПД турбин. Заметим, что в соответствии с алгоритмом программы [6] для расчёта этой схемы требуется решать уравнение с 10 неизвестными.

```
link_valve[0].I2, turb[0].I1); turb_(0);
lsplit_(turb[0].I2, turb[1].I1,
deaer[0].I1, 93./113.); turb_(1);
turb[0].lex.G = turb[0].I2.G - turb[1].I1.G;
link_(turb[1].I2, cond[0].I1); cond_(0);
link_(cond[0].I4, pool[0].I1); pool_(0);
link_(pool[0].I2, pump[2].I1); pump_(2);
link_(pump[2].I2, cond[0].I3);
link_(cond[0].I2, pump[0].I1); pump_(0);
link_(pump[0].I2, deaer[0].I2); deaer_(0);
link_(deaer[0].I3, pump[1].I1); pump_(1);
link_(pump[1].I2, boiler[0].I1); boiler_(0);
link_(boiler[0].I2, valve[0].I1); valve_(0);
elgen[0].N = shaft_power(0); elgen_(0);
elgen[0].S = pump[2].I2.G * pump[2].L;
elgen[0].P -= elgen[0].S;
```

Рис. 1 – Регенеративный цикл:

а – упрощённая схема цикла; б – программный код для его расчёта

Если расходы турбин не заданы, а находятся в процессе расчёта (скажем, по формуле Стодоль-Флюгеля или с помощью характеристик) алгоритм несколько усложняется, т.к. сегмент схемы, содержащий турбинную цепочку, должен быть рассчитан путём решения нелинейной системы уравнений для определения давлений между отсеками. Для этого должна быть составлена дополнительная функция для расчёта невязок. Для данного примера эта функция имеет стандартный вид (рис. 2) (программа на рис. 1б при этом изменяется незначительно). В случае наличия промежуточных элементов (клапанов, перегревателей, сепараторов и пр.) этот код удлиняется, но его составление по-прежнему является достаточно тривиальным. Текст программы может составляться вручную или генерироваться автоматически на основании определённых правил разбора структуры схемы, представленной в графическом виде.

Преимущества объектного подхода

Приём, заключающийся в выделении сегмента схемы для её более детального расчёта может быть обобщён и на другие часто встречающиеся комбинации элементов: регенераторы с охладителями пара и конденсата, котлы-утилизаторы, сепараторы-пароперегреватели, насосы с турбоприводом и другое оборудование. Расчёт выделенной подсхемы проводится обычно с помощью того же стандартного итерационного алгоритма. Это даёт основание называть предлагаемый метод расчёта термодинамических циклов *итерационно-рекурсивным*.

Рассмотрим в качестве иллюстрации возможные схемы котла-утилизатора (рис. 3):

- в виде простого теплообменника (рис. 3а);
- в виде схемы с двумя теплообменниками и испарителем (рис. 3б);

```

_TURB_
    link_ (valve [0].I2, turb [0].I1);
    for (i=0; i <= turbnum-1; i++)
    {
        if (turb [i].ishaft != turb [0].ishaft) continue;
        turb_ (i);
        turb_split(i);
    }
    for (i=0; i < turbnum-1; i++)// невязки по расходам
    {
        turb_Gex (i);
        dg = (turb[i+1].I1.G - (turb[i].I2.G - turb[i].Iex.G))/turb[0].gpt0.G;
        dG += dg*dg;
    }
_END_TURB_

```

Рис. 2 – Дополнительный код для расчета цепочки турбин в случае прямой задачи

- представление варианта рис. 3б одним элементом (рис. 3в);
- с двумя теплообменниками и более детальной моделью испарителя (рис. 3г);
- представление варианта рис. 3г одним элементом (рис. 3д).

Варианты рис. 3в и рис. 3д являются типичными случаями применения *объектной композиции*. С практической точки зрения такой подход позволяет заметно сократить затраты времени на составление тепловых схем и их расчёты.

Примером может служить комбинированный газопаровой цикл, изображённый на рис. 4, программа расчёта которого укладывается в шесть строк кода.

Методы расчёта могут быть переписаны с учётом особенностей функционирования включающих их компонентов в различных условиях. В данной реализации методы организованы как указатели на функции («виртуальные» функции) для расчёта балансов расхода и энергии или моделирования экономичности элемента. Изначально каждый компонент содержит указатели на базовые методы, реализованные в ядре программы.

Например, если теплообменник по умолчанию работает как регенератор для газотурбинной установки, метод расчёта использует задание коэффициента регенерации. Для расчёта другой разновидности – теплообменника для испарителя котла-утилизатора исходная модель не подходит, поскольку температуры теплоносителя по одной из веток на входе и выходе одинаковы. Решение достигается переписыванием метода таким образом, чтобы в качестве условия задавалась степень сухости пара на входе при *наследовании* всех данных, описывающих исходный теплообменник. Указатель на пользовательскую функцию расчёта присваивается при инициализации, а её код включается в программу расчёта и компилируется вместе с остальными модулями, необходимыми для

расчёта конкретного цикла. По мере решения различных задач дописанные процедуры накапливаются и могут повторно использоваться в новых проектах т.е., программный комплекс расчёта циклов является ещё и *расширяемым*.

Другой пример демонстрирует интересные возможности сочетания объектного подхода с итерационным методом расчёта тепловых схем, реализацию которых сложно себе представить при традиционных методах моделирования [5, 6]. Речь идёт о включении полноценных алгоритмов аэродинамического расчёта проточных частей турбомашин, входящих в цикл, с целью более обоснованной оценки экономичности, массогабаритных и стоимостных параметров установки.

В режиме проектирования цикла при расчёте турбомашин можно воспользоваться алгоритмами предварительного проектирования [7, 8], которые заключаются в поиске оптимальной конфигурации проточной части турбин или компрессоров на основании одномерного аэродинамического расчёта. Более того, применяя алгоритм экспресс-профилирования лопаток, уже на самых ранних стадиях проектирования можно составить достаточно детальное представление об облике турбин и компрессоров чтобы начать конструкторскую проработку агрегата и технико-экономическую оценку полученного варианта.

Когда уже на этапе анализа циклов, известны геометрические параметры турбомашин, возможны и более сложные постановки задач расчёта проточной части, вплоть до трёхмерных *CFD*-расчётов. Такие постановки становятся реальными с точки зрения трудозатрат проектировщика благодаря предлагаемому итерационному алгоритму расчёта, который требует лишь одного расчёта турбомашин на итерацию, а общее количество последовательных приближений предельно мало.

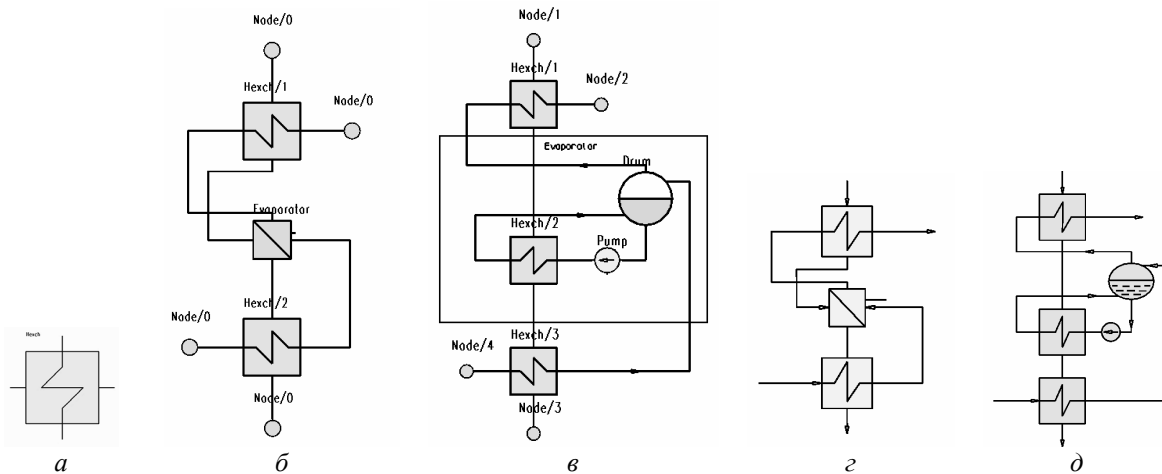


Рис. 3 – Способы представления модели котла-утилизатора:
 а – простой теплообменник; б – схема с испарителем; в – то же одним элементом;
 г – с детальной моделью испарителя; д – то же одним элементом

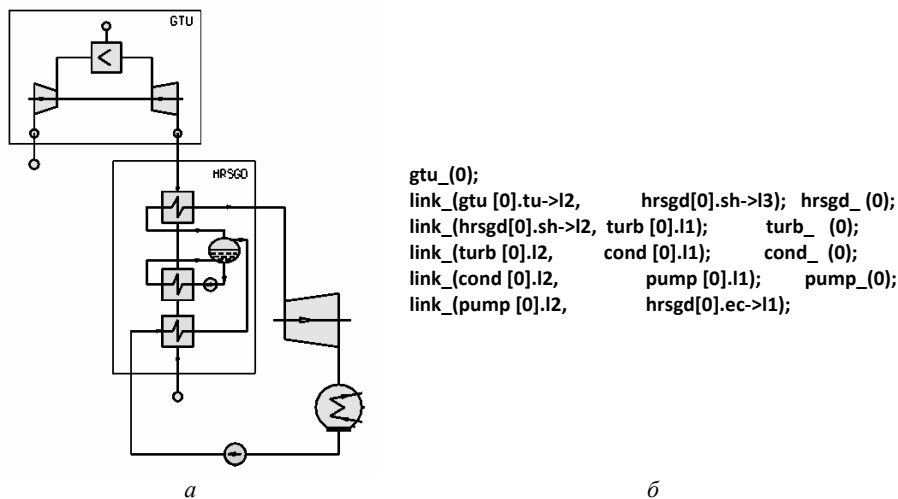


Рис. 4 – Комбинированный цикл с представлением ГТУ и котла-утилизатора в агрегированном виде: а – схема цикла; б – программный код для его расчёта

Может быть переписана также стандартная процедура, по умолчанию выполняющая итерационный цикл расчёта и вывод результатов вычислений. Это позволяет проводить многовариантные расчёты, оптимизацию по выбранным параметрам и критериям, организовывать нестандартный вывод информации с минимальными затратами на программирование. Такая гибкость и открытость в принципе не возможна в программах с традиционной архитектурой.

Программная реализация

Программный комплекс для расчёта тепловых схем реализован на базе свободного компилятора языка C [9], отличающегося малыми размерами, рекордной скоростью компиляции, переносимостью, поддержкой стандарта ISO C99 и дру-

гими полезными свойствами. С точки зрения пользователя программирование упрощается за счёт добавления процедур и макросов, ориентированных на расчёты термодинамических циклов.

Ядро системы содержит базовые модели большинства типов элементов, которые встречаются в паровых и газотурбинных циклах для авиационного и наземного применения [3, 4]. Поддерживается набор встроенных моделей рабочих тел, включая водяной пар (совместимый со стандартом IAPWS-IF97), продукты сгорания (стандартные или генерируемые в соответствии с составом используемого топлива) и др., а также возможность подключения обширной библиотеки свойств NIST REFPROP [10].

Запуск пользовательских программ сопровождается компиляцией текущей программы, её объединением с ядром системы, оформленным в

виде статической библиотеки, и внешними динамическими библиотеками (например, свойств рабочих тел, аэродинамического расчёта турбомашин и т.п.). После этого выполняется запуск программы с командной строки или из графического интерфейса пользователя. Результатом работы программы являются текстовая (листинг) и графическая (например, диаграммы цикла) информация.

В случае использования графической среды для анализа термодинамических систем в ней рисуется схема, составляется (или опционально генерируется) текст программы расчёта, вводятся исходные данные, выполняются расчёты и проводится пост-процессинг с занесением результатов расчёта в структуру данных схемы. Результаты расчётов могут отображаться непосредственно на схеме, в текстовых протоколах а также в виде графиков и диаграмм. Готовый проект сохраняет всю имеющуюся информацию по схеме, включая и программу расчёта, и может быть повторно использован для решения различных задач.

Тестирование программы проводилось путём детального сравнения с результатами расчётов по программам [5, 6], а также с данными, приведёнными в различных литературных источниках. Набор тестовых примеров в настоящее время содержит варианты расчёта отдельных элементов, фрагментов циклов и полных схем различного типа, включая паровые, газотурбинные (стационарного и авиационного применения), комбинированные и перспективные на проектировочном и нерасчётных режимах работы. Рисунки к статье подготовлены с использованием разработанной автором интерактивной среды для моделирования термодинамических систем, особенности реализации которой будут представлены в отдельной публикации.

Выводы

Представленный новый алгоритм расчёта термодинамических циклов, основанный на объектно-ориентированном подходе, сочетает такие качества как простота реализации, эффективность, универсальность, гибкость в использовании, расширяемость и переносимость.

Накопленный опыт в решении конкретных задач моделирования с использованием предлагаемых инструментов является предметом отдельного, более детального рассмотрения.

В частности, хотелось бы более подробно остановиться на способах решения задач, связанных с оптимизацией проектных параметров тепловых схем, их энергетическим и термоэкономическим анализом. Особого рассмотрения заслуживают проблемы расчета тепловых схем на частичных режимах работы: паровых турбин с учетом особенностей системы парораспределения и газовых, принимая во внимание влияние потоков

охлаждающего воздуха на экономичность проточной части турбин.

Список литературы

- 1 **Lytle, J. K.** The Numerical Propulsion System Simulation: An Advanced Engineering Tool for Airbreathing Engines [Text] / **J. K. Lytle** // ISABE 2001-1216. – 2001.
- 2 **Alexiou, A.** 2006, Gas Turbine Engine Performance Model Applications Using an Object-Oriented Simulation Tool [Text] / **A. Alexiou, K. Mathioudakis** // *ASME Turbo Expo 2006: Power for Land, Sea, and Air*, Barcelona, Spain, 8–11 May 2006. – ASME Paper No. GT-2006-90339. – P. 109–116.
- 3 **Рыжкин, В. Я.** Тепловые электрические станции [Текст] : учебник для вузов / Под ред. **В. Я. Гиршвельда**. – 3-е изд., перераб. и доп. – М. : Энергоатомиздат, 1987. – 328 с.
- 4 Теория и расчет воздушно-реактивных двигателей [Текст] / Под ред. **С. М. Шляхтенко** [Текст] : учеб. для вузов. – 2-е изд., перераб и доп. – М. : Машиностроение, 1987. – 568 с.
- 5 **Elmegaard, B.** The Engineer's "DNA by Example". Edition 4. Department of Mechanical Engineering Technical University of Denmark. 2003.
- 6 Cycle-Tempo. Technical Notes. A program for thermodynamic modeling and optimization of energy conversion systems. Delft University of Technology, 2007.
- 7 **Говорущенко, Ю. Н.** Автоматизированное предварительное проектирование проточной части многоступенчатых паровых турбин [Текст] / **Ю. Н. Говорущенко, Г. Л. Романов, Е. Э. Скибина** // Теплоэнергетика. – 1991. – № 6. – 3 с.
- 8 **Govorushchenko, J.** A Uniform Approach to Conceptual Design of Axial Turbine/Compressor Flow Path / **J. Govorushchenko, L. Moroz, P. Pagur** // The Future of Gas Turbine Technology. 3rd International Conference. 11-12 October 2006, Brussels, Belgium – 11 p.
- 9 **Tiny C Compiler** News 2009 [Online]. – Режим доступа: <http://bellard.org/tcc/>. – 15.01.2016.
- 10 **NIST Standard Reference Database 23.** NIST Reference Fluid Thermodynamic and Transport Properties Database (REFPROP): Version 9.1 [Online]. – Режим доступа: <http://www.nist.gov/srd/nist23.cfm>. – 10.12.2015.

Bibliography (transliterated)

- 1 **Lytle, J. K.** (2001), "The Numerical Propulsion System Simulation: An Advanced Engineering Tool for Airbreathing Engines", ISABE 2001-1216.
- 2 **Alexiou, A. and Mathioudakis, K.** (2006), "Gas Turbine Engine Performance Model Applications Using an Object-Oriented Simulation Tool", *ASME Turbo Expo 2006: Power for Land, Sea, and Air*, Barcelona, Spain, 8–11 May 2006, ASME Paper No. GT-2006-90339, pp. 109–116.
- 3 **Ryzhkin, V. J.** (1987), *Teplovyje jelektricheskie stancii* [Thermal power plants], 3th ed., Jenergoatomizdat, Moscow, Russia.
- 4 **Shljahntenko, S. M.** (1987), *Teorija i raschet vozdušno-reaktivnyh dvigatelej* [Theory and design of airbreathing engines], 2nd ed., Mashinostroenie, Moscow, Russia.

- 5 **Elmegaard, B.** (2003), The Engineer's "DNA by Example". Edition 4. Department of Mechanical Engineering Technical University of Denmark.
- 6 **Cycle-Tempo** (2007). Technical Notes. A program for thermodynamic modeling and optimization of energy conversion systems. Delft University of Technology.
- 7 **Govorushchenko, J. N., Romanov, G. L. and Skibina, E. E.** (1991), "Computer-aided preliminary design of the steam turbine's flow path", *Теплоэнергетика*, vol. 6, 3 p.
- 8 **Moroz, L., Govorushchenko, J. and Pagur, P.** (2006), "A Uniform Approach to Conceptual Design of Axial Turbine/Compressor Flow Path", *The Future of Gas Turbine Technology. 3rd International Conference*, Brussels, Belgium, 11–12 October 2006. – 11 p.
- 9 **Tiny C Compiler** (2009), News [Online], available at: <http://bellard.org/tcc/> (Accessed 15 January 2016).
- 10 **NIST Standard Reference Database 23** (2015), NIST Reference Fluid Thermodynamic and Transport Properties Database (REFPROP): Version 9.1 [Online], available at: <http://www.nist.gov/srd/nist23.cfm> (Accessed 10 December 2015).

Сведения об авторах (About authors)

Говорущенко Юрий Николаевич – кандидат технических наук, старший научный сотрудник кафедры турбиностроения, Национальный технический университет «Харьковский политехнический институт», г. Харьков, Украина; e-mail: jng1952@hotmail.com.

Govorushchenko Yuriy Nikolayevich – Candidate of Technical Sciences, Senior researcher of Turbine Projection Chair named after prof. Makovski V. M., National Technical university "Kharkov Polytechnic Institute", Kharkov, Ukraine.

Пожалуйста ссылаетесь на эту статью следующим образом:

Говорущенко, Ю. Н. Объектно-ориентированный итерационно-рекурсивный алгоритм моделирования термодинамических систем [Текст] / **Ю. Н. Говорущенко** // Вісник НТУ «ХПІ». Серія: Енергетичні та теплотехнічні процеси й устаткування. – Харків : НТУ «ХПІ», 2016. – № 8(1180). – С. 16–21. – Бібліогр.: 10 назв. – ISSN 2078-774X. – doi: 10.20998/2078-774X.2016.08.02.

Please cite this article as:

Govorushchenko, Yu. N. (2016), "Object-oriented Iterative-recursive Algorithm for Thermal-hydraulic Systems Simulation", *Bulletin of NTU "KhPI". Series: Power and heat engineering processes and equipment*, no. 8(1180), pp. 16–21, ISSN 2078-774X, doi: 10.20998/2078-774X.2016.08.02.

Будь ласка посилайтесь на цю статтю наступним чином:

Говорущенко, Ю. М. Объектно-ориентированный итерационно-рекурсивный алгоритм моделирования термодинамических систем [Текст] / **Ю. М. Говорущенко** // Вісник НТУ «ХПІ». Серія: Енергетичні та теплотехнічні процеси й устаткування. – Харків : НТУ «ХПІ», 2016. – № 8(1180). – С. 16–21. – Бібліогр.: 10 назв. – ISSN 2078-774X. – doi: 10.20998/2078-774X.2016.08.02.

АНОТАЦІЯ Приведений опис об'єктно-орієнтованого ітераційно-рекурсивного алгоритму моделювання термодинамічних систем, пристосованого для вирішення широкого кола практичних задач, який відрізняється ефективністю, гнучкістю, розширюваністю та переносимістю. На деяких прикладах продемонстровані переваги об'єктного підходу до аналізу термодинамічних циклів. Описані особливості програмної реалізації з використанням алгоритмічної мови загального призначення і показана корисність інтеграції з графічним середовищем для аналізу термодинамічних систем.

Ключові слова: об'єктний підхід, моделювання, термодинамічні системи.

Поступила (received) 25.01.2016