

УДК 004.422

А.С. Свиридов, В.С. Лазарев

**РАЗРАБОТКА БАЗОВОЙ АБСТРАКЦИИ ДЕЙСТВИЙ ПО ВЫПОЛНЕНИЮ
МАТЕМАТИЧЕСКИХ ОПЕРАЦИЙ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ
PHP***

Базовая абстракция действий, процесс разработки которой описывается в данной статье, представляет собой систему действий, направленных на достижение цели. Единичные действия, входящие в эту систему, состоят из элементов, над которыми выполняется некоторая функция, с помощью некоторого инструмента. В качестве примера базовой абстракции действий была создана базовая абстракция действий по выполнению математических операций. Описан процесс создания данной абстракции на языке программирования PHP. Выбор данного языка сделан в силу легкого для понимания синтаксиса, поддержки объектно-ориентированного программирования, а также web – ориентированности. Параллельно можно сделать вывод о том, насколько данный язык программирования оптимален для создания базовых абстракций. Сама базовая абстракция действий по выполнению математических операций создавалась на базе объектно-ориентированных возможностей языка. По сути, ставилась задача анализа, насколько возможности языка программирования, и объектно-ориентированной технологии подходят для реализации абстракции действий. Анализ первого прототипа реализации действия показал свою несостоятельность из-за невозможности повторного использования данного кода, но подтолкнул к использованию классов. Одного класса для реализации базовой абстракции действия оказалось недостаточно, и было принято решение о разработке классовой структуры, реализующей базовую абстракцию действий по выполнению математических операций в языке программирования PHP. Классовая структура, реализующая базовую абстракцию действий, была реализована согласно требованиям, но с элементами эмпирики. После окончания построения классовой структуры и ее разработки были определены перспективы дальнейших исследований в этом направлении.

Разработка информационных систем; математические операции; абстракция; язык программирования PHP.

A.S. Sviridov, V.S. Lazarev

**DEVELOPMENT OF BASIC ABSTRACTION OF ACTIONS
FOR PERFORMANCE OF MATHEMATICAL OPERATIONS
IN THE PHP PROGRAMMING LANGUAGE**

The basic abstraction of actions which process of development is described in this article represents system of the actions directed on achievement of the purpose. The single actions entering into this system consist of elements over which some function, by means of some tool is carried out. As an example of basic abstraction of actions the basic abstraction of actions for performance of mathematical operations was created. The paper describes the process of creating this abstraction in the programming language PHP. The choice of language is made by virtue of easy-to-understand syntax, support for object-oriented programming, and web - oriented. In parallel, it can be concluded about how this programming language is best for creating basic abstractions. Just basic abstraction of action to implement mathematical operations created on the basis of object-oriented language features. In fact, the task was to analyze how the possibility of programming language and object-oriented technology suitable for the implementation of an abstraction action. The analysis of the first prototype implementation of the actions shown to be ineffective because of the inability to re-use this code but pushed to the use of classes. One class for the im-

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 14-07-00910.

plementation of the basic abstraction action was not enough, and it was decided to develop a class structure that implements the basic abstraction of action to implement mathematical operations in the programming language PHP. Class structure that implements the basic abstraction of action has been implemented according to the requirements, but with elements of empiricism. After finishing the construction of the class structure and its development prospects have been identified for further research in this direction.

Development of information systems; mathematical operations; abstraction; PHP programming language.

Введение. Базовая абстракция действий, процесс разработки которой описывается в данной статье, представляет собой систему действий, направленных на достижение цели [1]. Единичные действия, входящие в эту систему, состоят из элементов, над которыми выполняется некоторая функция, с помощью некоторого инструмента [2–5]. Стоит задача разработки базовой абстракции действий средствами языков программирования высокого уровня.

Современные языки программирования часто делят по типу используемых абстракций на структурные языки (скриптовые) и объектно-ориентированные. Объектно-ориентированные имеют в своем составе абстракции – классы, которые содержат в себе переменные, функции, обладают возможностью наследования. Нужно определить, насколько абстракции в программировании (классы) для реализации базовой абстракции действий. В качестве примера базовой абстракции действий решено создать базовую абстракцию действий по выполнению математических операций. Математические операции лежат в основе вычислительной техники, именно на их базе строятся сложные математические модели, составляющие основу любого программного обеспечения. Немаловажно отметить в то же время простоту и доступность математических операций для понимания широкой аудитории. Теперь нужно рассмотреть, какие действия нужно совершить, чтобы реализовать базовую абстракцию действия по выполнению математических операций.

Базовая абстракция действия для реализации математических операций.

Представим элементы, функции, инструменты по реализации математических операции в виде действий. Для этого заполним табл. 1.

Таблица 1

Базовая абстракция действия для реализации математической операции

Э	Ф	И	Цель
Э1 Переменная a Переменная b	Ф1 Математическая операция	И1 Конструктор класса действия	Реализовать математическую операцию
Значения переменных	Параметр	Правила алгебры Готовые классы элементов, функций	Элементы классов
Задать значение, получить значение	Выполнить	Сопоставление алгебры и классовой структуры	Функции классов
Класс языка программирования	Структура функции языка программирования	Синтаксис языка программирования	Структура классов

Согласно таблице, для достижения цели «Реализовать математическую операцию» нужно совершить действие, в котором элементами будут «Переменная а» и «Переменная б», функцией будет «Математическая операция», а инструментом «Конструктор класса действия».

Для того, чтобы получить элементы «Переменная а» и «Переменная б», нам необходимо над элементами «Значения переменных», выполнить функции «Задать значение» и «Получить значение» с помощью инструмента «Класс языка программирования».

Для того, чтобы получить функцию «Математическая функция», нам необходимо над элементом «Параметр» произвести функцию «Выполнить» с помощью инструмента «Структура функции языка программирования».

Для получения инструмента «Конструктор класса действия» нам необходимо над элементами «Правила алгебры» и «Готовые классы элементов, функций» произвести функцию «Сопоставления алгебры и классовой структуры» с помощью инструмента «Синтаксис языка программирования».

Можно заметить, что собрав воедино элементы «Значения переменных», «Параметр», «Правила алгебры» и «Готовые классы элементов, функций», мы получим «Элементы классов». Аналогично получаем «Функции классов», и «Структуру классов». Можно заметить, собрав воедино «Элементы классов», «Функции классов» и «Структуру классов», получаем также цель, которую нам нужно достигнуть (табл. 1). После описания базовой абстракции можно приступить к обоснованию выбора языка программирования.

Обоснование выбора языка программирования. Первое требование – поддержка возможностей объектно-ориентированного программирования. Как видно из табл. 1, действия по созданию элементов функций и инструментов используют элементы ООП.

За последние годы рынок web – разработки в мире увеличился более чем в 1,5 раза. В мире прослеживается тенденция роста использования web-технологии – это удобно, и имеют прогнозы, что за такими сервисами будущее [6]. Следовательно, язык должен быть web – ориентированным.

Далее нужно учесть субъективный фактор – опыт программирования на языке «С++». Использование языка с похожим синтаксисом позволит сократить затраты времени на обучение.

И напоследок важным требованием является простота настройки среды для написания программ. Иными словами, настройка среды должна быть легкой в установке для пользователя, а сама среда бесплатной, что увеличивает ее доступность для широкой аудитории.

Ориентируясь на данные требования, был выбран язык PHP. Данный язык удовлетворяет всем требованиям:

1. В PHP поддерживается ООП [7, 8]. Объектно-ориентированная технология, встроенная в язык, аналогична объектно-ориентированному программированию в языке C++.
2. PHP – web – ориентированный язык, который может быть встроен в HTML, что обеспечит простоту создания форм представления. Также одним из значительных преимуществ PHP является поддержка широкого круга баз данных. Можно воспользоваться расширением, специфичным для отдельной базы данных, таким как MySQL.
3. Для создания базовой абстракции язык программирования должен быть гибким и динамическим. Именно эти свойства позволяют изменять компоненты базовой абстракции, создавая новые действия. PHP является динамическим [9], что также упрощает создание с его помощью web – приложений.

4. Простота работы и настройки: для реализации проекта вполне достаточно будет использование возможностей Джентльменского набора Web-разработчика («Д.н.в.р», читается «Денвер») – проекта Дмитрия Котерова [10, 11]. Набор сочетает в себе локальный web-сервер (Apache), интерпретатор PHP и сервер баз данных MySQL.

На основании описанных выше причин, мы считаем, что именно этот язык является оптимальным выбором для нашей задачи.

Определение классовой структуры для реализации базовой абстракции.

При разработке данной абстракции решено использовать возможности объектно-ориентированного программирования. Для того, что определить оптимальную классовую структуру, необходимо пройти путь от момента реализации через синтаксис языка программирования до построения классовой структуры, создающей элементы, функции, инструменты (табл. 1) действия по выполнению математической операции. Опишем этапы практического получения оптимальной классовой структуры действия:

1. **Реализация действия через синтаксис языка программирования.** Программа будет содержать три переменные (два числа – входящие и одна переменная результата) с оператором множественного выбора – CASE. Именно такую реализацию предложит программист непосредственно для реализации одной математической операции. Вероятность повторного использования этого кода для создания нового действия отсутствует. То есть, для изменения программы придется ее почти полностью переписать.
2. **Реализация действия как экземпляра класса `myAction`.** Для того, чтобы можно было работать с созданным действием дальше, возможно использовать объектно-ориентированные возможности языка PHP. Теперь программа по реализации действия будет представлять собой один класс с четырьмя функциями, входные данные – также три переменные, оператор множественного выбора – внутри функции класса. Действие создается в виде экземпляра класса. Однако возникает проблема: при создании нескольких действий создадутся соответствующие классы, имеющие в своем составе несколько функций, но использующие лишь одну. Необходимо решить эту проблему.
3. **Реализация действия как класса `myAction`, использующего методы других классов.** Для решения проблемы, сформированной в предыдущем пункте, было решено для переменных, констант и четырех арифметических операций создать отдельные классы. Таким образом, классовая структура программы, полученная на этом этапе (рис. 1), будет состоять из шести классов – два класса для переменных и четыре класса для действий. Оператор множественного выбора определяет, какую операцию необходимо использовать в данный момент.

На рис. 1 можно увидеть, что действие, реализованное через экземпляр `Action` класса `myAction`, использует методы наследников базовых классов элементов `Element` и функций `Function`. Наследниками базового класса `Element` являются классы `Constant` и `Variable`. Программно отличие этих двух классов сводится к отсутствию функции `Change_var()` в классе констант. Это отражает физический смысл переменных и констант. Наследниками базового класса функций `Function` являются классы, реализующие математические операции: умножение (`Multi`), деление (`Divide`), сложение (`Plus`), вычитание (`Minus`). Как можно заметить, функции, реализующие арифметические операции имеют единую классовую структуру. Нужно подумать о том, как объединять действия. Такие объединения будем называть работой [12, 13].

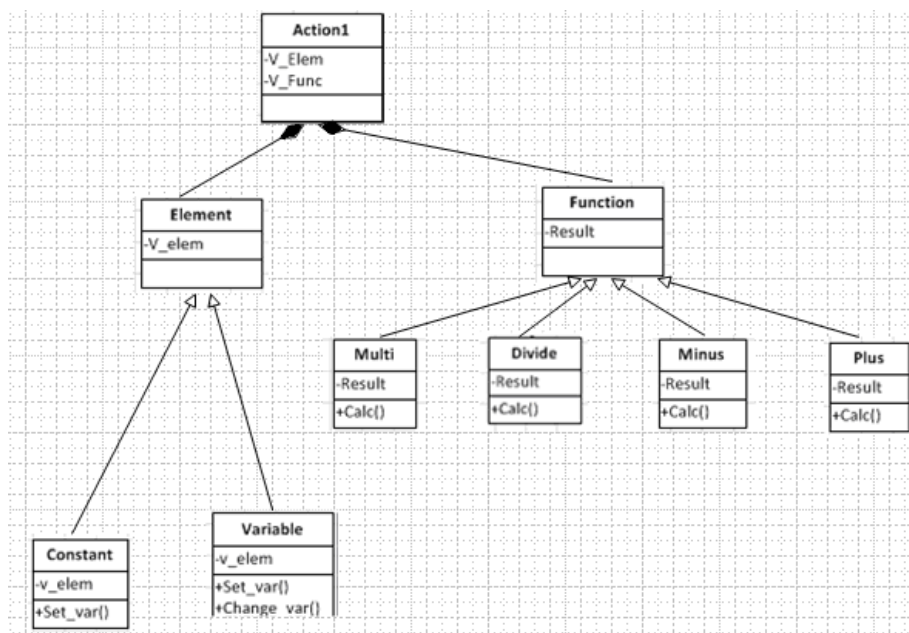


Рис. 1. Классовая структура, реализующее действие в языке PHP

4. **Реализация работы из действий как классовой структуры.** Классовая структура из предыдущего этапа претерпит следующие изменения – будет добавлен базовый класс работы, а базовые классы элементов Element и функций Function станут его наследниками. Таким образом, классовая структура программы реализующей действия, будет включать один (базовый) класс работы, один базовый и два класса элементов, и один базовый и четыре класса функций. Работы создаются как экземпляры базового класса.
5. **Реализация работы из действий как классовой структуры, с абстрактным классом работы и классами работ.** Классовая структура, созданная на предыдущем этапе, привязывает любые изменения в работе к изменениям в базовом классе Work. В свою очередь, так как базовые классы элементов и функций являются наследниками класса Work, любые изменения в последнем отразятся и на них. Было решено создать классы конкретных действий, являющимися наследниками класса Work. Само действие будет создаваться так, как было описано в этапе 3. Например, на рис. 2 показана работа, состоящая из двух действий, одно из которых – Action1 – реализует действие по вычитанию переменной из константы.

Последний вариант является наиболее соответствующим концепции. Однако, необходимо сказать о том, какие ограничения имеет данная абстракция. Действия в ней характеризуются обязательным присутствием:

- ◆ двух экземпляров базового класса элементов;
- ◆ одного экземпляра базового класса функции;
- ◆ отсутствием инструмента в явном виде (используются возможности языка программирования);
- ◆ одного значения результата, в данном случае, одного числа.

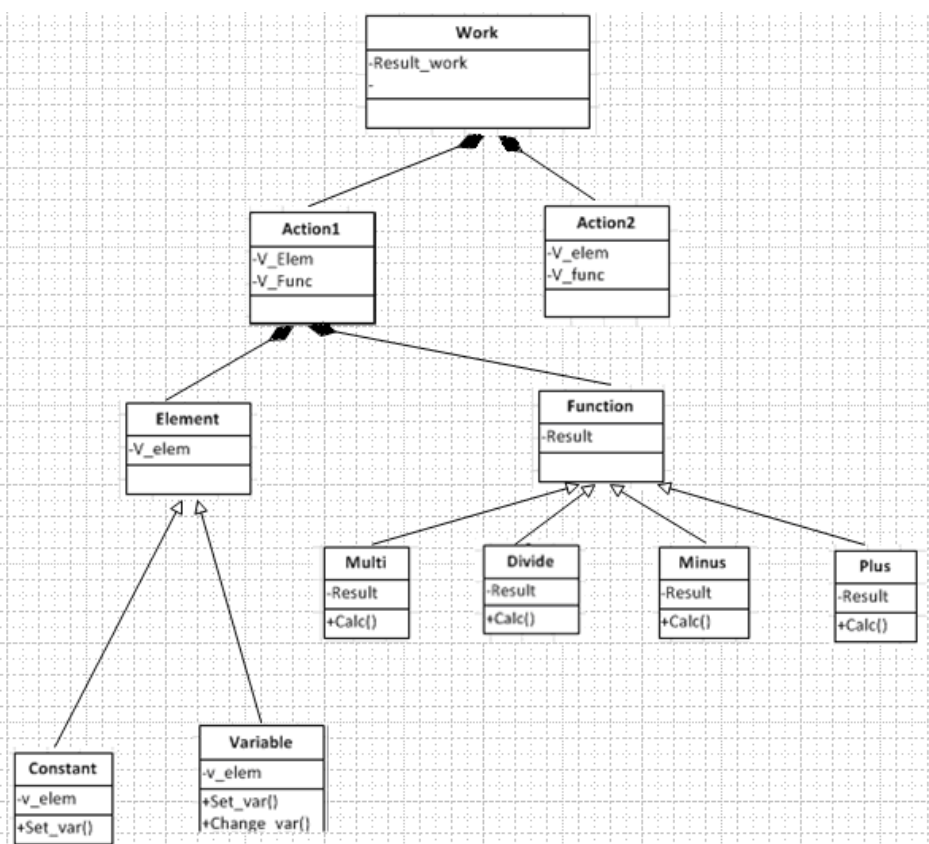


Рис. 2. Классовая структура программы, реализующей работу из действий

Заключение и дальнейшие исследования. В процессе работы была решена поставленная задача – на языке php была создана базовая абстракция действий по выполнению математических операций. Таким образом, с одной стороны можно сделать вывод об адекватности языка для создания базовых абстракций действий по выполнению математических операций. С другой стороны, необходимо отметить недостатки, которые имеют место.

Самый существенный недостаток состоит в том, что созданные классы усложняют структуру программы. Для написания программы, реализующей операцию сложения, придется создавать классы для переменных, констант, отдельно класс для функции, реализующей математическую операцию. Данный факт является обратной стороной представления действий в едином виде, и дает преимущество в виде возможностей настройки действий и их конфигурирования.

Другой недостаток заключается в том, что в классовой структуре инструмент в явном виде не представлен. Иными словами, роль инструмента играет язык программирования и его возможности. Это не критично для действий по выполнению математических операций, рассмотренных в данной статье, однако для более сложной задачи, вполне возможно, потребуется надстройка над языком. Таким образом, одно из направлений дальнейшего исследования заключается в создании более сложных математических работ, предусматривающих:

- ◆ один элемент, как, например, в работе с унарными операциями;

- ◆ создание инструмента для выполнения работ, выходящих за возможности команд языка программирования;
- ◆ два и более значения результата (уравнения второго и более порядка).

Второе направление – создание базовой абстракции для действий, отличных от математики, применимых непосредственно к информационным системам.

Возникает также вопрос хранения данных: спецификой данной проблемы будет являться то, что нужно хранить не только значения результата, и но метод его получения [14, 15].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Лазарев В.С., Rogozov Ю.И., Свиридов А.С.* Построение процесса как результата отношения между действиями // Сборник трудов XI Всероссийской научной конференции молодых ученых аспирантов и студентов “ИТСАиУ-2013”. – Таганрог: Изд-во ЮФУ, 2013. – Т. 2. – С. 20-23.
2. *Лазарев В.С., Свиридов А.С., Rogozov Ю.И.* Методологический подход как преемник объектного подхода в разработке информационных систем // Информатизация и связь. – 2014. – № 2. – С. 85-89.
3. *Лазарев В.С.* Разработка концепции методологической информационной системы // Сборник статей международной научно-практической конференции “Технологии разработки информационных систем”. – Таганрог: Изд-во ЮФУ, 2014. – С. 157-161.
4. *Микита Р.М., Rogozov Ю.И., Свиридов А.С., Стукотий Л.Н.* Концепция построения информационной модели предприятия // Телекоммуникации. – 2004. – № 8. – С. 20-24.
5. *Rogozov Ю.И.* Системный подход к созданию метода разработки информационных объектов на основе метамodelей // Информатизация и связь. – 2011. – № 7. – С. 57-62.
6. Рынок веб-разработки в Рунете: тренды и ожидания. – Интернет-ресурс. – <http://www.ratingruneta.ru/research/webanalytics2011/#2.1>.
7. *Протасевич С.* Объектно-ориентированный РНР: специально для начинающих. – Интернет-источник: http://ruseller.com/lessons.php?rub_id=37&id=1145.
8. *Кузнецов Максим, Симдянов Игорь.* Объектно-ориентированное программирование на РНР. – СПб.: БХВ-Петербург, 2007. – 608 с.
9. Динамические языки программирования. – Интернет-источник: <http://bourabai.kz/alg/dynamic.htm>.
10. *Котеров Д.В.* Джентльменский набор Web-разработчика. – Интернет-источник: <http://www.denwer.ru/>.
11. *Котеров Дмитрий, Костарев Алексей.* РНР. В подлиннике. – СПб.: БХВ-Петербург, 2008. – 1104 с.
12. *Rogozov Ю.И., Свиридов А.С.* Подход к построению информационных систем на основе методологического подхода // Сборник статей Международной научно-практической конференции “Информационные технологии и дидактика в обучении”. Т. 1. – Таганрог: Изд-во ЮФУ, 2014. – С. 3-8.
13. *Rogozov Ю.И.* Подход к определению метасистемы как системы // Труды ИСА РАН. – 2013. – Т. 63, № 4. – С. 92-110.
14. *Кучеров С.А., Rogozov Ю.И., Свиридов А.С.* Метод построения структурно-независимых баз данных с использованием реляционных технологий // Информационные технологии. – 2011. – № 2. – С. 52-59.
15. *Rogozov Ю.И., Свиридов А.С., Кучеров С.А.* Структурно-независимые базы данных для разработки конфигурируемых пользователем информационных систем // Информатизация и связь. – 2013. – № 3. – С. 20-23.

REFERENCES

1. *Lazarev V.S., Rogozov Yu.I., Sviridov A.S.* Postroenie protsesssa kak rezul'tata otnosheniya mezhdru deystviyami [The build process as a result of the relationship between actions], Sbornik trudov XI Vserossiyskoy nauchnoy konferentsii molodykh uchenykh aspirantov i studentov “ITSAiU-2013” [Proceedings of XI all-Russian scientific conference of young scientists and students “Idayu-2013”]. Taganrog: Izd-vo YuFU, 2013. Vol. 2, pp. 20-23.

2. Lazarev V.S., Sviridov A.S., Rogozov Yu.I. Metodologicheskiy podkhod kak preemnik ob"ektnogo podkhoda v razrabotke informatsionnykh sistem [Methodological approach as the successor of object approach in the development of information systems], *Informatizatsiya i svyaz'* [Informatization and Communication], 2014, No. 2, pp. 85-89.
3. Lazarev V.S. Razrabotka kontseptsii metodologicheskoy informatsionnoy sistemy [The methodological development of the concept of information system], *Sbornik statey mezhdunarodnoy nauchno-prakticheskoy konferentsii "Tekhnologii razrabotki informatsionnykh sistem"* [Collection of articles of international scientific-practical conference "Technologies of information systems development"], Taganrog: Izd-vo YuFU, 2014, pp. 157-161.
4. Mikita R.M., Rogozov Yu.I., Sviridov A.S., Stukotiy L.N. Kontseptsiya postroeniya informatsionnoy modeli predpriyatiya [The concept of building information model of the enterprise], *Telekommunikatsii* [Telecommunications], 2004, No. 8, pp. 20-24.
5. Rogozov Yu.I. Sistemnyy podkhod k sozdaniyu metoda razrabotki informatsionnykh ob"ektov na osnove metamodeley [A systematic approach to the creation of the method of development of information objects based on metamodels], *Informatizatsiya i svyaz'* [Informatization and Communication], 2011, No. 7, pp. 57-62.
6. Rynok veb-razrabotki v Runete: trendy i ozhidaniya [The market for web development in Runet: trends and expectations]. Available at: <http://www.ratingruneta.ru/research/webanalytics2011/#2.1>.
7. Protasevich S. Ob"ektno-orientirovanny PHP: spetsial'no dlya nachinayushchikh [Object-oriented PHP: especially for beginners]. Available at: http://ruseller.com/lessons.php?rub_id=37&id=1145.
8. Kuznetsov Maksim, Simdyanov Igor'. Ob"ektno-orientirovannoe programmirovaniye na PHP [Object-oriented programming in PHP]. St. Petersburg.: BKhV-Peterburg, 2007, 608 p.
9. Dinamicheskie yazyki programmirovaniya [Dynamic programming languages]. Available at: <http://bourabai.kz/alg/dynamic.htm>.
10. Koterov D.V. Dzhentl'menskiy nabor Web-razrabotchika [Gentleman's set a Web developer]. Available at: <http://www.denwer.ru/>.
11. Koterov Dmitriy, Kostarev Aleksey. PHP. V podlinnike [PHP. In the original]. St. Petersburg: BKhV-Peterburg, 2008, 1104 p.
12. Rogozov Yu.I., Sviridov A.S. Podkhod k postroeniyu informatsionnykh sistem na osnove metodologicheskogo podkhoda [The approach to the construction of information systems on the basis of the methodological approach], *Sbornik statey Mezhdunarodnoy nauchno-prakticheskoy konferentsii "Informatsionnye tekhnologii i didaktika v obuchenii"* [Collection of articles of International scientific-practical conference "Information technologies and didactics in education"]. Vol. 1. Taganrog: Izd-vo YuFU, 2014, pp. 3-8.
13. Rogozov Yu.I. Podkhod k opredeleniyu metasistemy kak sistemy [The approach to the definition of the meta-system as], *Trudy ISA RAN* [Proceedings of ISA RAS], 2013, Vol. 63, No. 4, pp. 92-110.
14. Kucherov S.A., Rogozov Yu.I., Sviridov A.S. Metod postroeniya strukturno-nezavisimyykh baz dannykh s ispol'zovaniem relyatsionnykh tekhnologiy [Method of Constructing Structure-Independent Databases with Using Relational Technologies], *Informatsionnye tekhnologii* [Information Technology], 2011, No. 2, pp. 52-59.
15. Rogozov Yu.I., Sviridov A.S., Kucherov S.A. Strukturno-nezavisimyye bazy dannykh dlya razrabotki konfiguriruemykh pol'zovatelem informatsionnykh sistem [Structurally independent database to develop user-configurable information systems], *Informatizatsiya i svyaz'* [Informatization and Communication], 2013, No. 3, pp. 20-23.

Статью рекомендовал к опубликованию д.т.н., профессор В.Н. Иванченко.

Свиридов Александр Славьевич – Южный федеральный университет; e-mail: a.sviridov@tsure.ru; 347928, г. Таганрог, ГСП 17-А, Некрасовский 44; тел.: 88634371787; кафедра системного анализа и телекоммуникаций; к.т.н.; доцент.

Лазарев Владимир Сергеевич – e-mail: PhillPro.Lazarev@yandex.ru; кафедра системного анализа и телекоммуникаций; аспирант.

Sviridov Alexander Slavevich – Southern Federal University; e-mail: a.sviridov@tsure.ru; 347928, Taganrog, GSP 17-A, Nekrasovsky, 44; phone: +78634371787; the department of system analysis and telecommunications; cand. of eng. sc.; associate professor.

Lazarev Vladimir Sergeevich – e-mail: PhillPro.Lazarev@yandex.ru; the department of systems analysis and telecommunications; postgraduate student.