

УДК 004.02

А.Н. Беликов, С.А. Белоусова, Е.А. Борисова

**РЕАЛИЗАЦИЯ МЕТОДА ПОСТРОЕНИЯ АБСТРАКЦИИ
МАТЕМАТИЧЕСКИХ ОПЕРАЦИЙ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ
ВЫСОКОГО УРОВНЯ***

Целью статьи является анализ реализации математических операций на языках программирования высокого уровня. Для этого были созданы программы на языках Python и Java, в основу которых положена созданная базовая абстракция выполнения математических операций. Базовая абстракция типов математических операций является системой действия, которая состоит из типов действий, которые необходимо выполнить для получения цели. Предполагается, что любой пользователь, заполнив ее своим собственным содержанием, может выполнять различные по сложности арифметические операции. Для создания более сложных программ, информационных систем необходим набор таких базовых абстракций, каждая из которых будет предназначена для выполнения различных типов действий. В свою очередь, единичное действие состоит из механизма действия: элементов, функций, инструментов. В статье базовая абстракция была заполнена конкретным содержанием – действиями, которые выполняются для создания программы выполнения математических операций. Основной задачей статьи является определение адекватности языка программируемому объекту – абстракции. Для выявления адекватности используемых языков для реализации базовой абстракции, она была запрограммирована на двух языках. Анализ эффективности языков проводился по следующим критериям: динамическая типизация, сложность реализации программы в веб, и вместе с тем, сложность веб-фреймворка, объектно-ориентированные возможности. По каждому из критериев описываются достоинства и недостатки каждого языка. Можно сделать вывод, что для решения описанной задачи целесообразнее использовать Python, т.к. он удовлетворяет требованиям, и по перечисленному набору критериев он превосходит язык Java.

Python; java; структура действия; базовая абстракция; веб-программирование.

A.N.Belikov, S.A.Belousova, E.A.Borisova

**IMPLEMENTATION OF THE METHOD OF MATHEMATICAL OPERATIONS
ABSTRACTION BUILDING IN HIGH LEVEL PROGRAMMING LANGUAGES**

The aim of the article is to analyze the implementation of mathematical operations on the high-level languages. For this, there have been established programs in Python and Java, which are based on basic abstraction of performing the mathematical operations. Basic abstraction of mathematical operations types consists of actions that must be performed for the purpose obtaining. It is assumed that any user can perform various arithmetic operations of any complexity by filling it with their own content. To create more complex programs and information systems there is needed a set of basic abstractions, each of which is designed to perform various types of actions. In turn, a single action consists of elements, functions, tools. In the article the basic abstraction was filled with concrete content – single actions that are performed to create a program to perform mathematical operations. The main objective of the article is to determine the adequacy of language to programmable object – abstraction. To identify the adequacy of the language for the implementation of the basic abstraction, it has been programmed in both languages. Analysis of the effectiveness of languages was held on the following criteria: dynamic typing, the complexity of the program on the web, and at the same time, the complexity of the web-framework, object-oriented features. For each of the criteria the advantages and disadvantages of each language are described. It can be concluded that it is advisable to use Python to solve the described problem, because it satisfies the requirements, and according to the listed set of criteria it transcends Java language.

Python; java; action structure; basic abstraction; web-programming.

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-07-04250.

Введение. В настоящее время при разработке информационных систем используются различные подходы [11]. Однако, ни один из существующих подходов не позволяет описать процесс разработки конфигурируемой ИС.

Конфигурируемые системы предназначены для самостоятельного создания конечным пользователем (экспертом предметной области) прикладной ИС для собственных нужд. В процессе такого создания пользователь выполняет совокупность действий. Хотя действия различны по содержанию, но имеют схожую структуру. Предполагается, что любой пользователь, заполнив структуру действия своим собственным содержанием, может выполнять различные по сложности действия. Для создания более сложных программ, информационных систем необходим набор таких базовых абстракций, каждая из которых будет предназначена для выполнения различных типов действий.

Целью статьи является рассмотрение структуры действия на примере выполнения математических операций и анализ реализации математических действий на языках программирования высокого уровня.

Постановка задачи. Необходимо получить структура действия по выполнению определенного типа математических операций, которую необходимо отразить на языке программирования высокого уровня – реализовать ее.

Для решения поставленной задачи необходимо представить структуру системы действия и механизма действия, которые требуется в дальнейшем представить в виде абстракций языка программирования. Необходимо показать степень адекватности абстракции языка программирования и реализуемых с его помощью структурных элементов программы, реализующей выполнение математических операций.

Для этого необходимо: 1) создать структуру математической абстракции; 2) выделить из нее составляющие компоненты; 3) определить типы связей между ними. Со стороны языка программирования необходимо показать, каким образом и насколько сложно можно описать компоненты математической абстракции с помощью двух используемых языков – Python и Java. Необходимо определить такой язык, с помощью которого можно описать абстракцию выполнения математических операций, используя минимальное количество операторов и структур абстракции языка программирования.

Для этого необходимо переложить структуру действия (структуру математической абстракции) в термины языка программирования. Тогда элементами действия будут являться переменные и константы. Функциями будут являться операции, в том числе математические. Каждый структурный элемент действия будет представлен на языке программирования с помощью класса. Для создания программы, реализующей математическую операцию определенного типа на языке программирования, необходимо программно реализовать выполнение действия, состоящего из переменных, являющихся элементами данного действия, математической функции, являющейся отношением между элементами данного действия, и конструктора класса действия, являющегося инструментом для данного действия.

Описание структуры базовой абстракции. Базовая абстракция представлена на рис. 1. Анализ и синтез являются известными методами системного подхода. Основой любого действия является цель, которая анализируется и разбивается на элементы, функции и инструменты действия [1, 2]. Разбивая цель на такие компоненты, мы получаем действие по получению данной цели. Аналогично анализируются элементы, функции и инструменты, при этом мы имеем действия по их получению, и т.д. Полученные в ходе анализа элементы, функции, инструменты действия более низкого уровня синтезируются, чтобы снова получить цель, как показано на рис. 1. Если цель не получена, то необходимо проверить содержание компонентов действия и изменить их.

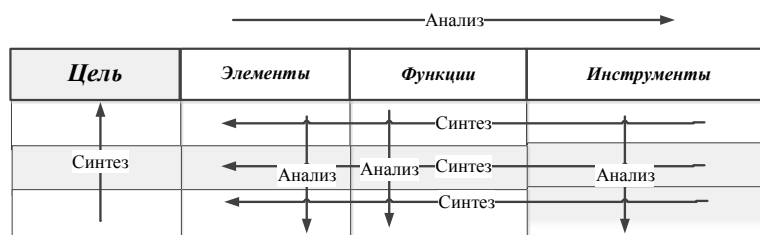


Рис. 1. Структура системы действия

Реализация базовой абстракции математической операции. Первым шагом будет представление элементов, функций и инструментов действия по получению математической операции на языке программирования высокого уровня в виде действий. [3–5].

Таким образом, представление элемента «Переменная» (Э1 и Э2) в виде действия можно описать следующим образом: элементом является значение переменной, функциями для данного действия являются Ф1 – задать значение и Ф2 – получить значение, а инструментом И1.1 – класс языка программирования.

Представим в таком же виде функцию Ф1 – математическая операция, получим следующее: элементом является Эф – «Параметр», функцией для действия по созданию математической операции является Фф – «Выполнить», а инструментом Иф – «Структура функции языка программирования».

Теперь составим действие для получения инструмента «Конструктор класса действия». Данное действие состоит из Эи1 – «Правила алгебры» состоящее из:

1. Выделение переменных;
2. Выделение констант;
3. Выделение алгебраических операций;
4. Получение алгебраического результата.

Вторым элементом действия по получению инструмента является Эи2 – «Готовые запрограммированные классы». Функцией для данного действия является Фи – «Сопоставление алгебры классовой структуре», а инструментом Ии – «Синтаксис языка программирования». Структура действия представлена на рис. 2.



Рис. 2. Система действия «Запрограммировать математическую операцию»

Из представленного шаблона некоторые элементы, функции и инструменты являются конечными и не имеет смысла расписывать действия по их созданию, а некоторые имеет.

Инструмент – это вид отношений между элементом и функцией (это правила по которым устанавливаются отношения).

М. Метод создания программы на языках программирования высокого уровня. Состоит из следующих шагов:

- 1) Выделение переменных
- 2) Выделение математической функции
- 3) Использование конструктора класса действия

Распишем функцию «Сопоставления алгебры классовой структуре» входящую в действие по созданию инструмента «Конструктор класса действия».

Итак, данное действие состоит из элементов Эф1.1 – «Тип элемента алгебраической формулы», который состоит из четырех подэлементов:

- 1) Эфи* 1. – число;
- 2) Эфи* 2. – латинская буква;
- 3) Эфи* 3. – Алгебраическая операция.

Второй элемент Эф1.2 «Тип класса», который состоит из трех подэлементов:

- 1) Эфи** 1. – Класс константы;
- 2) Эфи** 2. – Класс переменной;
- 3) Эфи** 3. – Класс операции.

Функцией данного действия является Фф - «Выбор типа элемента». Инструмент представляет собой набор правил по которым функция выполняется над элементами. Иф1. Правила сопоставления:

- 1) Если часть формулы – число, то выбрать класс-константу.
- 2) Если часть формулы – латинская буква, то класс – переменная.
- 3) Если часть формулы – алгебраическая операция, то класс – операция.

Описанные действия показаны на рисунке 3.

Для элемента Эф1.1.2 «Тип класса» распишем действие.

Элементами действия по получению подэлемента Эфи** 2. – Класс переменной, являются Э1. Функция получения значения переменной и Э2. Функция задания значения переменной. Функции данного действия Ф1. «Задать функцию получить значение» и Ф2. «Задать функцию задать значение», а инструментом является структура класса. Описанные действия показаны на рис. 4.

Элементами действия по получению подэлемента Эфи** 3. – Класс операции, являются Э1. Имя класса, Э2. Метод выполнения операции и Э3. Атрибуты метода выполнения операции. Функции данного действия Ф1. Объявить класс, Ф2. Задать Метод выполнения операции, Ф3. Передать Атрибуты метода выполнения операции. Инструментом также как и в предыдущих двух действиях является И**1 – «Структура класса».

Описанные действия показаны на рис. 5.

Цель	Элемент	Функция	Инструмент
Сопоставления алгебры классовой структуре	Эфи1.1.1 Часть формулы Эфи1.1.2 Тип элемента формулы (Эфи* 1. - Алгебраическая константа Эфи* 2. - Алгебраическая переменная Эфи* 3. - Алгебраическая операция) Эфи1.1.3 Типы классов (Эфи** 1. - Класс константы Эфи** 2. - Класс переменной Эфи** 3. - Класс операции)	Ф1. Выбор типа элемента	И1. Правила сопоставления: 1) Если часть формулы – число, то выбрать класс-константу 2) Если часть формулы – латинская буква, то класс – переменная. 3) Если часть формулы – математическая операция, то класс – операция.

Рис. 3. Механизм действия

Цель	Элемент	Функция	Инструмент
Определение Типа класса	(Для Эфи**2. - Класс переменной) Э1. Функция получения значения Э1. Функция задания значения (Для Эфи**3. - Класс операции) Э1. Имя класса Э2. Метод выполнения операции Э3. Атрибуты метода выполнения операции	(Для Эфи**2. - Класс переменной) Ф1. Задать функцию получить значение Ф2. Задать функцию задать значение (Для Эфи**3. - Класс операции) Ф1. Объявить класс Ф2. Задать Метод выполнения операции Ф3. Передать Атрибуты метода выполнения операции	И**1. Структура класса

Рис. 4. Механизм действия получения элемента Типы классов

Цель	Элемент	Функция	Инструмент
Получение структуры класса	Э1. Объект класса. Э2. Экземпляр класса	Ф1. Методы класса	И1. Синтаксис языка программирования

Рис. 5. Механизм действия получения инструмента Структура класса

На рис. 2–5 представлены действия программиста, которые ему необходимо совершить для того, чтобы реализовать определенный тип математических операции на языке программирования. Согласно описанной структуре были созданы программы, реализующие выполнение математических операций на языках Python и Java [6–8].

Анализ реализации математических операций на языках высокого уровня. Была произведена попытка реализовать описанную ранее систему действия, при этом структурными элементами программы также являются механизмы действия, поэтому необходимо определить, какие объекты языка программирования позволяют нам реализовывать эти действия. По модели действия (рис. 1, 2, 3) проанализируем объекты абстракции рассматриваемых языков, которые необходимы для выполнения того или иного типа действия. Для этого построим табл. 1, в которой показаны различия между рассматриваемыми языками путем перечисления объектов их языковых абстракций на примере выполнения типа действия «Программирование операции умножения».

Так как на двух языках была реализована одна и та же программа, она предъявляет одинаковый набор требований ко всем языкам. Такой прием сужает область сравнения, но делает ее однородной. Кроме того, для каждого языка было проанализировано несколько отдельных реализаций программы, подготовленных различными программистами. Групповой подход имеет два преимущества. Во-первых, сглаживаются различия между отдельными программистами, которые могут лишить достоверности любые сравнения, основанные на единственном «образце» для каждого языка. Во-вторых, появляется возможность сравнить изменчивость характеристик программ, составленных на разных языках.

В ходе сравнительного исследования сопоставлялись различные аспекты каждого языка. Основными критериями анализа языков программирования применительно к решаемой задаче являются следующие: динамическая типизация, сложность реализации программы в веб, и вместе с тем, сложность веб-фреймворка, объектно-ориентированные возможности.

При объявлении какой-либо переменной мы используем конструкции языка программирования – это есть не что иное, как абстракции языка программирования. Для рассматриваемых языков эти абстракции разные.

Языковые абстракции объявления переменной, константы, операции в Java подразумевают, что для каждого типа данных (вещественный, целый и т.д.) необходимо использовать определенный тип абстракции языка программирования. Эта ситуация продемонстрирована на рис. 4 и в табл. 1.

Например, для действия составления классовой структуры программы одним из элементов является класс переменной, который в Java необходимо использовать несколько раз для описания переменных разного типа (вещественный, целый и т.д.), а в Python необходимо описать этот класс только единственной языковой абстракцией, и в ходе выполнения программа сама определяет какой тип данных используется, что заложено в правилах языковой абстракции. Т.е. в языке Python, например, для объявления переменной любого типа используется один класс, а в языке Java для каждого типа переменных необходимо было создавать отдельный класс, которых на данный момент имеется три.

Таблица 1

Сравнение языковых абстракций языков программирования на примере нескольких типов действий

Тип действия	Python		Java		
	Компоненты действия	Объекты абстракции языка программирования	Компоненты действия	Подтип действия	Объекты абстракции языка программирования
Программирующие операции умножения	Элементы	Переменные	Элементы	Умножение целых чисел	Переменная int
				Умножение вещественных чисел	Переменная float
	Функции	Класс языка	Функции	Умножение целых чисел	Класс сложения целых чисел
				Умножение вещественных чисел	Класс сложения вещественных чисел

В правилах использования языковых абстракций каждого из рассматриваемых языков наблюдается свойство динамической типизации и от этих правил зависит длина кода создаваемой программы.

Заключение. Получена структура выполнения арифметической операции. Была создана базовая абстракция для выполнения арифметических операций на языке программирования высокого уровня. Предполагается, что любой пользователь, заполнив ее своим собственным содержанием, может выполнять различные по сложности арифметические операции.

Для создания более сложных программ, информационных систем необходим набор таких базовых абстракций, каждая из которых будет предназначен для выполнения различных действий. Но все эти базовые абстракции будут связаны между собой, будут взаимопроникать [9, 10, 11].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Рогозов Ю.И.* Методология создания субъектно-ориентированных систем // Информатизация и связь. – 2014. – № 2. – С. 6-10.
2. *Rogozov Yu.I., Sviridov A.S.* The Concept of Methodological Information Systems Development // Proceedings of IEEE 8th International Conference on Application of Information and Communication Technologies – AICT2014. IEEE Catalog Number CFP1456H-PRT. ISBN: 978-1-4799-4120-92. – P. 120-123.

3. *Sergey Kucherov, Alexander Sviridov, S.A. Belousova.* The formal model of structure-independent databases // Proceedings of 3rd International Conference on Data Management Technologies and Applications, Vienna, Austria, Scitepress - Science and Technology Publications. ISBN: 978-989-758-035-2. – P. 146-152.
4. *Alexander Sviridov, Andrey Grishchenko, Svetlana Belousova.* Performance estimation of selecting and inserting procedures in the structure-independent database // Proceedings of IEEE 8th International Conference on Application of Information and Communication Technologies – AICT2014. IEEE Catalog Number CFP1456H-PRT. ISBN: 978-1-4799-4120-92. – P. 202-207.
5. *Yu.I. Rogozov, A.S. Sviridov and A.N. Belikov.* Method of a Structure-Independent Databases Design in Configurable Information Systems // Proceedings of 3rd International Conference on Data Management Technologies and Applications, Vienna, Austria, Scitepress – Science and Technology Publications. ISBN: 978-989-758-035-2. – P. 139-145.
6. *Django vs Flask vs Pyramid: Choosing a Python Web Framework.* [Электронный ресурс] – Режим доступа: <https://www.airpair.com/python/posts/django-flask-pyramid>.
7. *Languages Versus D.* [Электронный ресурс] – Режим доступа: <http://www.prowiki.org/wiki4d/wiki.cgi?LanguagesVersusD>.
8. *Васильев А.Н.* Java: объектно-ориентированное программирование: Учебное пособие. – СПб.: Питер, 2011. – С. 19.
9. *Sviridov A., Lipko Y., Belikov A.* Approach to automatized formation of workplaces structure in configurable information systems // Proceedings of «8th International Conference on Application of Information and Communication Technologies» (AICT-2014), Astana, Kazakhstan, ISBN: 978-1-4799-4120-92. 2014. – P. 383-386.
10. *Белуцова С.А., Розозов Ю.И.* Анализ подходов к созданию пользовательского интерфейса // Известия ЮФУ. Технические науки. – 2014. – № 6 (155). – С. 142-148.
11. *Беликов А.Н.* Обзор и анализ методов и средств проектирования конфигурируемых информационных систем // Известия ЮФУ. Технические науки. – 2014. – № 6 (155). – С. 137-141.

REFERENCES

1. *Rogozov Yu.I.* Metodologiya sozdaniya sub"ektno-orientirovannykh sistem [The methodology for the creation of subject-oriented systems], *Informatizatsiya i svyaz'* [Informatization and communication], 2014, No. 2, pp. 6-10.
2. *Rogozov Yu.I., Sviridov A.S.* The Concept of Methodological Information Systems Development, *Proceedings of IEEE 8th International Conference on Application of Information and Communication Technologies – AICT2014.* IEEE Catalog Number CFP1456H-PRT. ISBN: 978-1-4799-4120-92, pp. 120-123.
3. *Kucherov S., Sviridov A., Belousova S.A.* The formal model of structure-independent databases, *Proceedings of 3rd International Conference on Data Management Technologies and Applications, Vienna, Austria, Scitepress – Science and Technology Publications.* ISBN: 978-989-758-035-2, pp. 146-152.
4. *Sviridov A., Grishchenko A., Belousova S.* Performance estimation of selecting and inserting procedures in the structure-independent database, *Proceedings of IEEE 8th International Conference on Application of Information and Communication Technologies – AICT2014.* IEEE Catalog Number CFP1456H-PRT. ISBN: 978-1-4799-4120-92, pp. 202-207.
5. *Rogozov Yu.I., Sviridov A.S. and Belikov A.N.* Method of a Structure-Independent Databases Design in Configurable Information Systems, *Proceedings of 3rd International Conference on Data Management Technologies and Applications, Vienna, Austria, Scitepress – Science and Technology Publications.* ISBN: 978-989-758-035-2, pp. 139-145.
6. *Django vs Flask vs Pyramid: Choosing a Python Web Framework.* Available at: <https://www.airpair.com/python/posts/django-flask-pyramid>.
7. *Languages Versus D.* Available at: <http://www.prowiki.org/wiki4d/wiki.cgi?LanguagesVersusD>.
8. *Vasil'ev A.N.* Java: ob"ektno-orientirovannoe programmirovaniye: Uchebnoye posobie [Java: object-oriented programming: a tutorial]. St. Petersburg: Piter, 2011, pp. 19.

9. *Sviridov A., Lipko Y., Belikov A.* Approach to automatized formation of workplaces structure in configurable information systems, *Proceedings of «8th International Conference on Application of Information and Communication Technologies» (AICT-2014)*, Astana, Kazakhstan, ISBN: 978-1-4799-4120-92. 2014, pp. 383-386.
10. *Belousova S.A., Rogozov Yu.I.* Analiz podkhodov k sozdaniyu pol'zovatel'skogo interfeysa [Analysis of approaches to user interface building], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 6 (155), pp. 142-148.
11. *Belikov A.N.* Obzor i analiz metodov i sredstv proektirovaniya konfiguriruemyykh informatsionnykh system [Review and analysis of methods and tools of configurable information systems design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 6 (155), pp. 137-141.

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

Беликов Александр Николаевич – Южный федеральный университет; e-mail: belikovmv@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; кафедра САиТ; аспирант; ассистент.

Белюсова Светлана Алексеевна – e-mail: s.belousova2011@gmail.com; кафедра САиТ; аспирант; ассистент.

Борисова Елена Александровна – e-mail: lb20062006@yandex.ru; кафедра САиТ; к.т.н.

Belikov Alexandr Nikolaevich – Southern Federal University; e-mail: belikovmv@gmail.com; 44, Nekrasovskiy, Taganrog, 347928, Russia; the department of system analysis and telecommunications; postgraduate student; assistant.

Belousova Svetlana Alekseevna – e-mail: s.belousova2011@gmail.com; the department of system analysis and telecommunications; postgraduate student; assistant.

Borisova Elena Aleksandrovna – e-mail: lb20062006@yandex.ru; the department of system analysis and telecommunications; cand. of eng. sc.; associate professor.