

УДК 81-116.6

DOI 10.30914/2072-6783-2023-17-3-406-412

КОНЦЕПЦИЯ БАЗОВОЙ ЕДИНИЦЫ ЯЗЫКА В РАМКАХ МИНИМАЛИСТСКОЙ ПРОГРАММЫ Н. ХОМСКОГО

А. И. Габитов, Л. Х. Шаяхметова

Казанский (Приволжский) федеральный университет, г. Казань, Российская Федерация

Аннотация. Введение. Теория минималистской программы Н. Хомского создала основу для поиска и идентификации универсальной грамматики естественных языков. Данная теория в свою очередь описывает врожденные компоненты способности человека к языкам. В рамках универсальной грамматики важную роль в построении синтаксиса играют лексические единицы. В работах исследователей данной тематики встречаются выводы, указывающие на сходство работы синтаксиса естественных языков и языков программирования. **Целью данного исследования** является формирование концепции базовой единицы языка, содержащей в себе основополагающие принципы работы синтаксиса для естественных языков и языков программирования. **Материалы и методы.** Исследование проводится на основе научных работ Н. Хомского, в частности на основе минималистской программы и теории универсальной грамматики. В исследовании применялись методы лингвистического наблюдения и описания, конструктивный метод, сравнительно-сопоставительный анализ потенциальных базовых единиц языка. В качестве таких единиц нами были выбраны объекты и лексические единицы для языков программирования и естественных языков соответственно. **Результаты исследования.** Анализ принципов функционирования объектов в языках программирования показывают наличие множественных точек соответствия с принципами, заложенными в теории минималистской программы. В сравнении с естественными языками языки программирования не имеют ограничений в лингвистическом исследовании устройства грамматики языка на уровне вычислительной системы. **Заключение.** Данные, полученные при лингвистическом изучении объектов в языках программирования и их обратном проектировании, могут быть применены для объяснения принципов работы лексических единиц в грамматике естественных языков.

Ключевые слова: контрастивный анализ, лексикон, минималистская программа, теория синтаксиса, универсальная грамматика, языки программирования

Авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Габитов А. И., Шаяхметова Л. Х. Концепция базовой единицы языка в рамках минималистской программы Н. Хомского // Вестник Марийского государственного университета. 2023. Т. 17. № 3. С. 406–412. DOI: <https://doi.org/10.30914/2072-6783-2023-17-3-406-412>

THE CONCEPT OF THE BASIC UNIT OF LANGUAGE IN THE SYSTEM OF N. CHOMSKY'S MINIMALIST PROGRAM

A. I. Gabitov, L. Kh. Shayakhmetova

Kazan Federal University, Kazan, Russian Federation

Abstract. Introduction. N. Chomsky's Minimalist program theory created the basis for the search and identification of a universal grammar of natural languages, a theory describing the innate components of a person's ability to use languages. According to this theory, lexical units play an important role in the construction of syntax. In the works of researchers on this topic there are conclusions indicating the similarity of the work of the syntax of natural languages and programming languages. **Purpose.** This study aims to form the concept of the basic unit of language containing the fundamental principles of the syntax for natural languages and programming languages. **Materials and methods.** The study is based on the scientific works of N. Chomsky, in particular on the basis of the Minimalist program and the theory of universal grammar. The study used methods of linguistic observation and description, constructive method, comparative analysis of potential basic units of the language. As such units, we have chosen objects and lexical units for programming languages and natural languages, respectively. **Results, discussion.** Analysis of the principles of functioning of objects in programming languages shows the presence of multiple points of correspondence with the principles laid down in the theory of the Minimalist program. Compared to natural languages, programming languages have no limitations in the

linguistic study of the structure of language grammar at the level of a computing system. **Conclusion.** The data obtained during the linguistic study of objects in programming languages and their reverse engineering can be used to explain the principles of operation of lexical units in the grammar of natural languages.

Keywords: contrastive analysis, lexicon, Minimalist program, programming languages, syntax theory, universal grammar

The authors declare no conflict of interest.

For citation Gabitov A. I., Shayakhmetova L. Kh. The concept of the basic unit of language in the system of N. Chomsky's Minimalist program. *Vestnik of the Mari State University*, 2023, vol. 17, no. 3, pp. 406–412. (In Russ.). DOI: <https://doi.org/10.30914/2072-6783-2023-17-3-406-412>

Введение

Язык занимает значимое место в регулировании любого человеческого сообщества: в воспитании и образовании, в написании законов, в культуре и идентификации его членов¹. В свою очередь, язык регулируется собственной внутренней структурой, определяющей закономерности взаимодействия единиц языка при применении в устной и письменной речи. Как и в любой другой структуре, в основе языка должна находиться минимальная единица, образующая его. Лингвистическая наука может исследовать язык только опосредованно, через языковые факты и явления, и при помощи обратного проектирования приходиться от частных выводов к общим правилам. Определение базовой единицы языка предоставляет возможность понять то, как язык устроен на самом глубинном уровне, и переосмыслить многие процессы, в которых он задействован. В данном исследовании мы предпринимаем попытку определения базовой единицы языка для получения с ее помощью доступа к универсальной грамматике, предполагаемому набору правил или принципов, присущих каждому человеческому языку.

Цель исследования: формирование концепции базовой единицы языка, содержащей в себе основополагающие принципы работы синтаксиса для естественных языков и языков программирования.

Материалы и методы

Учеными-лингвистами было предложено много теоретических моделей синтаксиса, такие как грамматика зависимостей Л. Теньера, категориальная грамматика К. Айдукевича и статистическая

грамматика Д. Брауна. В настоящее время одной из преобладающих моделей описания синтаксиса является модель Н. Хомского. Слово «синтаксис» пришло из греческого языка и буквально означало аранжировку². В своей первоначальной модели, названной Синтаксические Структуры (Syntactic Structures) [1], Н. Хомский внес существенный вклад в развитие теории «генеративная грамматика» (generative grammar), которая рассматривает лингвистику как изучение гипотетической врожденной грамматической структуры. В 1965 году он опубликовал статью «Аспекты теории синтаксиса» (Aspects of the Theory of Syntax), ставшей основой для книги «Стандартная теория» (Standard theory) [2]. Эта научная работа отличалась тем, что в ней признавалась глубокая и поверхностная структура предложения. Постепенно пересматривая свои взгляды на теорию синтаксиса, Н. Хомский пришел к созданию минималистской программы (Minimalist program) [3]. Данная программа построена на основе модели управляющей привязки Н. Хомского и предполагает максимально возможное упрощение теории синтаксиса любого естественного языка для определения основополагающих правил. В ее рамки входят такие понятия, как «простота», «экономичность», «симметричность» и «неизбыточность». Согласно Н. Хомскому язык имеет уровни представления: фонетическая форма (phonetic form, PF) и логическая форма (logical form, LF). Он утверждает, что, поскольку каждый человеческий язык состоит из звука и значения, их необходимо интерпретировать по отдельности [2, с. 77].

В ходе работы над определением правил взаимодействия лексических единиц в английском языке Н. Хомский высказывает предположение,

¹ Brown K., Miller J. *Syntax: A Linguistic Introduction to Sentence Structure*. 2nd Ed. Routledge, 1996. P. 16.

² Жеребило Т. В. *Словарь лингвистических терминов*. Изд. 5-е, испр. и доп. Назрань : ООО «Пилигрим». 2010. 486 с.

что параметры универсальной грамматики относятся не к вычислительной системе человеческого мозга, а исключительно к словарному запасу. Потенциально это означает, что каждый параметр относится к свойствам конкретных элементов лексики или к категориям лексических единиц. Существует только один человеческий язык, наполняемый различной лексикой, овладение языком является вопросом определения правил взаимодействия лексических единиц. Свойства лексики ограничены универсальной грамматикой или другими системами человеческого разума. Если содержательные элементы, такие как части речи (глаголы, существительные и т. д.), взяты из универсального словаря, то параметризованными будут только функциональные элементы [4, с. 119].

Результаты исследования, обсуждения

Лексикон представляет собой набор лексических элементов, каждый из которых представляет собой артикулированную систему признаков. Он указывает для каждого подобного элемента фонетические, семантические и синтаксические свойства, присущие ему, но не более того. Если признаки лексической единицы относят ее к какой-либо большой категории, не все признаки соответствующей категории будут наследоваться. Для английского языка примером будет категория глаголов действия, которые могут наследовать отдельные признаки категории глаголов. Каждая лексическая единица, относящаяся к данной категории, должна содержать параметры и свойства своей категории [5, с. 48]. Лексическое значение слова, обозначающего конкретный глагол, должно указывать ровно столько его свойств, сколько необходимо, чтобы определить его звуковую, смысловую и синтаксическую роли посредством действия общих принципов, параметризованных для рассматриваемого языка. В нем не должно быть избыточной информации [4, с. 118].

Помимо этого, для минималистской программы характерен принцип *экономии представления* (economy of representation). Синтаксическая структура должна ограничиваться минимально необходимыми для функционирования элементами. Н. Хомский делает вывод, что на фоне минималистского подхода к интерпретации синтаксиса выделяются очертания некоторых еще более общих принципов, которые мы могли бы считать руководящими принципами.

Данные принципы имеют более высокий приоритет в системе синтаксиса [6; 4, с. 137].

Присвоение лексической единице синтаксической категории основано на понятии «каноническая структурная реализация» (canonical structural realization, CSR). CSR физического объекта – N, действия – V и так далее. При отсутствии иных доказательств индивид, взаимодействующий с языком, будет считать, что слово относится к определенной канонической структурной реализации. К примеру, слово, относящееся к действию, является глаголом. Существуют и исключения, в языках обычно есть существительные, относящиеся к действиям, а также глаголы, которые не относятся к действиям [7; 4, с. 28].

Выводы, сделанные Н. Хомским, граничат с областью разработки программного обеспечения. Принципы, заложенные в универсальной грамматике, имеют дублеров в принципах функционирования языков программирования. Для сравнения нам потребуется определить уровни языков и минимальную единицу, на основе которых формируются эти языки. Большинство современных языков программирования считаются объектно-ориентированными языками (ООП). В некоторые языки, которые изначально разрабатывались как языки процедурного программирования, с годами были добавлены функции ООП. В объектно-ориентированных языках объекты имеют идентичность, состояние и поведение, связанные с ними. Данные процессы регулируются набором принципов¹.

Принцип *абстрагирования* определяет скрытие кода реализации, который не требуется для использования другими объектами. Абстрагирование – это способ выделить набор наиболее важных атрибутов и методов и исключить незначимые. Соответственно, абстракция – это использование всех таких характеристик для описания объекта. Важно представить объект минимальным набором полей и методов без ущерба для решаемой задачи. Принцип *полиморфизма* позволяет объекту определяться или использоваться по-разному в разных контекстах. Принцип *наследования* делает классы объектов способными повторно использовать код других классов. Например, отношения

¹ Задорожный С. С., Фадеев Е. П. Объектно-ориентированное программирование на языке Python. Учебно-методическое пособие по дисциплине «Введение в компьютерные технологии». М. : Физический факультет МГУ имени М. В. Ломоносова, 2022. С. 4.

между объектами могут быть назначены для создания иерархии классов, чтобы создать семейства связанных объектов. Более общее понятие становится «родителем», а более специфичное и подробное – «наследником». У одного «родителя» может быть несколько дочерних структур. Например, класс *животное* будет предоставлять признаки таким объектам, как *лошадь*, *обезьяна*, *тигр*. Принцип *инкапсуляции* данных утверждает, что объекты содержат все необходимое для функционирования, включая методы объекта и любые связанные данные. Затем объект может сделать свои интерфейсы доступными для других объектов, чтобы они могли использовать этот объект. Каждый объект – независимая структура.

ООП позволяет использовать один и тот же элемент программного кода в различных ситуациях. Множество объектов создается на основе классов, у каждого объекта могут быть собственные параметры. Перечисленные принципы подтверждают присутствие определенной экономии средств, участвующих в процессе написания программного кода, это схоже с принципами экономии в системе синтаксиса естественных языков. Объект можно представить в виде целостного набора данных, для манипуляции с объектами используются специальные инструменты обработки называемыми методами. Регуляция взаимодействия объектов изначально определена в наборе данных, определяющих сам объект. Каждый объект превращается в универсальный класс объектов, и определяются еще более общие классы, чтобы объекты могли совместно использовать модели и повторно использовать определения классов в своем коде. Объект – это ключевой элемент, управляющий мышлением компьютера, именно с объектов начинается определение правил регулирования их взаимодействия в программной среде [8].

Различия между компьютерными и естественными языками обширны. Однако с развитием информационных наук стало очевидно, что между гуманитарной областью языкознания и технической областью программирования имеются точки соприкосновения [9]. Существенное отличие заключается в более высокой степени абстракции языков программирования. Они имеют дело не с объектами физического мира, а с отвлеченными понятиями. Языки программирования отличаются от естественных языков способом их интерпретации. Естественные языки, используемые для общения, в человеческом сознании оперируют субъек-

тивными категориями и образами, полученными из окружающего мира. Компьютерные языки при помощи специальных программ преобразуются в машинный код, управляющий устройствами компьютера напрямую. Аналоги грамматических правил, которым подчиняются естественные языки, для компьютера будут находиться в машинном коде. Естественные языки сложнее и их интерпретация требует не только знания компьютерных команд, но и культурного контекста, в котором находится говорящий. Естественные языки не ограничены работой исключительно с объективной информацией, в них присутствуют категории данных, которые не свойственны машинному языку, к примеру эмоции. В отличие от них языки программирования должны однозначно описывать алгоритм, состоящий из последовательности определенных действий, и они являются расширенной формой математических формул [8].

Языки программирования, как и естественные языки, представлены в значительном количестве и разделены на определенные области применения. Помимо этого, они разделяются на высокий и низкий уровни в зависимости от того, насколько не опосредовано взаимодействие кода с вычислительными возможностями машины. Любой алгоритм есть последовательность предписаний. Степень детализации предписаний определяет уровень языка программирования – чем меньше детализация, тем выше уровень языка. Языки программирования высокого уровня отличаются также и тем, что больше похожи на человеческий язык ввиду более легкого понимания человеком написанного. В дополнение к этому эти языки не участвуют в управлении памятью машины и абстракции функций [4, с. 23].

Языки программирования низкого уровня отличаются от высокоуровневых. В них нет абстракции, они предельно точны, что делает их возможными для понимания компьютерами. В языках программирования существуют элементы, понятные на базовом уровне, например операторы сложения или умножения. Однако большее количество конструкций будут уникальны для своего языка программирования и потребуют от пользователя специальных знаний, как и в случае с естественными языками. Схожими элементами будут наличие структуры и принципов, по которым они работают, семантические и синтаксические правила, словарный состав. Если провести аналогию с естественными языками, то машинные языки низкого уровня

будут являться универсальной грамматикой, описываемой Н. Хомским. Языки, находящиеся на глубинном уровне, оперируют когнитивными способностями мозга (если это человек) или вычислительными возможностями машины (если это компьютер). В свою очередь, языки программирования высокого уровня представляют собой аналоги человеческих языков.

Между лексикой естественных языков и языков программирования существует разница. Слова в естественном языке отличаются друг от друга по звучанию, прямые аналоги встречаются тем реже, чем дальше языки находятся друг от друга в языковых семьях. При этом для общения активный словарь должен включать значительное количество слов для минимально возможного построения. Языки программирования, как правило, имеют большое количество схожих ключевых слов, операторов и идентификаторов.

В то время как синтаксис отвечает за техническую корректность конструкций языка, семантика отвечает за смысловое наполнение. В человеческом языке слово может иметь много прямых, переносных и контекстуальных значений. В языках программирования смысловое значение конкретной лексической единицы не изменяется в зависимости от контекста и человеческого фактора. Однако неправильно или неточно сформулированная команда приведет к ошибке.

Если естественные языки образуются на основе лексических единиц, то для языков программирования это будут объекты. Все объекты обладают тремя основными характеристиками: идентичностью, состоянием и поведением. Понимание этих характеристик имеет значение для понимания того,

как работают объекты и объектно-ориентированная логика. Идентичность означает, что каждый объект может быть отличен от всех других объектов. Состояние относится к свойствам объекта. Значения переменных в объекте содержат данные, которые можно добавлять, изменять, удалять. Характеристика поведения определяет действия, которые может выполнять объект или быть подвергнутым воздействию другого объекта [8]. В число элементов в программировании, которые могут быть определены как объекты, входят следующие:

- 1) переменные, содержащие значения, которые можно изменить;
- 2) структуры данных для организации и обработки данных;
- 3) функции – процедуры, выполняющие определенную задачу;
- 4) методы, которые представляют собой запрограммированные процедуры, определенные как компоненты родительского класса и включенные в любой экземпляр этого класса.

В языках программирования взаимодействие и управление объектами задается самим человеком. Рассмотрим пример создания объекта на языке программирования C# (см. рис. 1).

В данном отрывке кода мы можем видеть создание объекта под названием «System». Объект принадлежит к классу TransportCard. За данным объектом закреплена внутренняя переменная *int balance*. Управление данным объектом осуществляется прикрепленным за ним методом TransportCard (int balance). Все элементы задумываются человеком и реализуются при помощи специальных команд, заложенных в языке программирования.

```
using System;
namespace HW_12
{
    internal class TransportCard
    {
        public event TransactionNotifier Notification;
        public event TransactionNotifier LowBalance;
        private int _balance;
        public TransportCard (int balance)
        {
            if (balance >= 0)
                _balance = balance;
            else
                Console.WriteLine("Баланс не может быть отрицательным");
        }
        public void Add (int a)
        {
            if (a > 0)
            {
                _balance += a;
                Notification.Invoke($"Пополнение на сумму {a}. Баланс = {_balance}");
            }
        }
        public delegate void TransactionNotifier(string message);
    }
}
```

Рис. 1. Создание объекта на языке программирования C# / Fig. 1. Creating an object in C#

Как уже было ранее упомянуто, согласно минималистской теории базовой единицей языка, содержащей грамматические правила, являются лексические единицы [4, с. 26]. Однако в отличие от объектов в языках программирования внутренние параметры слов не явны. Нам понадобится провести обратное проектирование выявленных закономерностей их взаимодействия. Интерес представляют семантический отбор и тематические свойства ведущих лексических единиц, именуемых вершинами. Они определяют структуру фразы, указывая, сколько других элементов будет в ней и какую семантическую роль получает каждый из них. Например, глагол *to give* должен быть указан в роли агента, роли темы и роли цели / получателя [10].

Одним из фундаментальных понятий, применяемых в универсальной грамматике, является «команда» (command). Каждая лексическая единица во фразе управляет или управляется другой, что схоже с логикой взаимодействия объектов в программировании. Существенный объем исследований в минималистской традиции посвящен именно тому, как фраза получает надлежащий порядок. Задача состоит в том, чтобы удалить все избыточные элементы в пользу максимально простого представления. Начальное правило составления фразы исходит из вершины, некоторые другие, скрытые правила, применяются к ним операциями универсальной грамматики. При помощи операций «слияние» (merge) и «перемещение» (move) элементы фразы достигают финальной точки определения взаимоотношений [8]. Визуально это осуществляется при помощи X-штрих-теории. У каждой фразы есть ведущий лексический элемент, он проецируется на всю фразу, взаимоисключающие или избыточные взаимоотношения исключаются при проверке. Продемонстрируем сказанное при помощи рисунка 2.

В вершине, отмеченной как VP, помещена глагольная группа. Каждый узел обязательно разбивается на две ветки, левая из которых будет являться спецификатором, узлом, который находится в сестринском отношении с вышестоящим узлом, в данном случае NP. В правую ветку VB выделяется узел-посредник, от которого в свою очередь отде-

ляются глагол, помеченный как V, и наречие, помеченное как AP. Разделение продолжается, пока каждый член предложения не будет в идеально минимизированных взаимоотношениях с другими.

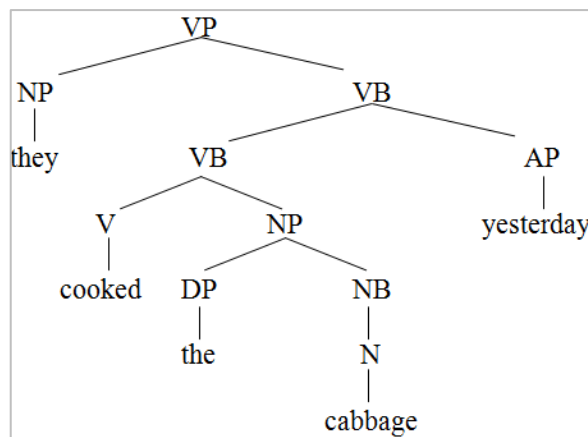


Рис. 2. X-штрих-теории представление фразы “They cooked the cabbage yesterday” / Fig. 2. X-bar theory representation of the phrase “They cooked the cabbage yesterday”

Заключение

Таким образом, несмотря на существенную разницу между языками программирования и естественными языками, через призму минималистской теории нами были выявлены следующие сходства:

- естественные языки и языки программирования взаимодействуют с более глубинным языком, располагающим основополагающими правилами грамматики;
- языки программирования и естественные языки имеют схожие принципы экономии в составлении правил грамматики;
- правила грамматики естественных языков и языков программирования содержатся в базовых единицах языка.

Невозможность выявить правила универсальной грамматики напрямую приводит к необходимости провести обратное проектирование и исследование базовых единиц естественного языка согласно минималистской программе. Используя существующие знания о методе функционирования и разработке языков программирования, мы по аналогии можем проследить путь от базовых единиц языка до глубинных правил универсальной грамматики.

1. Chomsky N. Syntactic Structures. The Hague/Paris : Mouton, 1957. 117 p.
2. Chomsky N. Aspects of the Theory of Syntax. Cambridge, Massachusetts : MIT Press, 1965. 261 p.
3. Сак А. Н., Бессонова Е. В. Сравнение синтаксического анализа предложения естественного языка // Балтийский гуманитарный журнал. 2021. Т. 10. № 1 (34). С. 373–377. DOI: <https://doi.org/10.26140/bgз3-2021-1001-0089>
4. Chomsky N. The Minimalist Program. 20th Anniversary Edition. Cambridge, Massachusetts : The MIT Press, 2015. 409 p.

5. Hornstein N. A Theory of Syntax: Basic Operations and the Minimalist. Cambridge : Cambridge University Press, 2009. 194 p.
6. Абдурахманова Р. Г. Универсальная грамматика как основа минималистской программы // Балтийский гуманитарный журнал. 2020. Т. 9, № 3 (32). С. 224–227. DOI: <https://doi.org/10.26140/bgз3-2020-0903-0055>
7. Ritonga M. The Role in SLA played by Universal Grammar // Journal of Applied Linguistics. 2022. Vol. 2. No. 1. Pp. 23–29. DOI: <https://doi.org/10.52622/joal.v2i2.76>
8. Fedorenko E., Ivanova A., Dhamala R., Bers M. U. The language of programming: a cognitive perspective // Trends in Cognitive Sciences. 2019. Vol. 23, no. 7. Pp. 525–528. DOI: <https://doi.org/10.1016/j.tics.2019.04.010>
9. Casalnuovo C., Sagae K., Devanbu P. Studying the difference between natural and programming language corpora // Empirical Software Engineering. 2019. Vol. 24. Pp. 1823–1868. DOI: <https://doi.org/10.48550/arXiv.1806.02437>
10. Umarlebbe J. H., Said S. M. B., Universal Grammar: Arguments for its Existence // Advances in Language and Literary Studies. 2021. Vol. 12. No. 31. P. 31.

Статья поступила в редакцию 16.05.2023 г.; одобрена после рецензирования 19.06.2023 г.; принята к публикации 24.07.2023 г.

Об авторах

Габитов Азат Ирекович

ассистент кафедры языковой и межкультурной коммуникации, Казанский (Приволжский) федеральный университет (420056, Российская Федерация, г. Казань, ул. Татарстан, д. 2), ORCID: <https://orcid.org/0000-0002-7995-5795>, gabit.azat@gmail.com

Шаяхметова Лейсан Хабировна

кандидат филологических наук, доцент, доцент кафедры языковой и межкультурной коммуникации, Казанский (Приволжский) федеральный университет (420056, Российская Федерация, г. Казань, ул. Татарстан, д. 2), ORCID: <https://orcid.org/0000-0003-3166-5219>, habirshah@mail.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

1. Chomsky N. Syntactic structures. The Hague/Paris, Mouton Publ., 1957, 117 p. (In Eng.).
2. Chomsky N. Aspects of the theory of syntax. Cambridge, Massachusetts, MIT Press, 1965, 261 p. (In Eng.).
3. Sak A. N., Bessonova E. V. Sravnenie sintaksicheskogo analiza predlozheniya estestvennogo yazyka [Comparison of parsing methods for the analysis of a natural language sentence]. *Baltiiskii gumanitarnyi zhurnal = Baltic Humanitarian Journal*, 2021, vol. 10, no. 1 (34), pp. 373–377. (In Russ.). DOI: <https://doi.org/10.26140/bgз3-2021-1001-0089>
4. Chomsky N. The Minimalist program. 20th Anniversary edition. Cambridge, Massachusetts, MIT Press, 2015, 409 p. (In Eng.).
5. Hornstein N. A Theory of syntax: basic operations and the minimalist. Cambridge, Cambridge University Press, 2009, 194 p. (In Eng.).
6. Abdurakhmanova R. G. Universal'naya grammatika kak osnova minimalistskoi programmy [Universal grammar as the basis of the Minimalist program]. *Baltiiskii gumanitarnyi zhurnal = Baltic Humanitarian Journal*, 2020, vol. 9, no. 3 (32), pp. 224–227. (In Russ.). DOI: <https://doi.org/10.26140/bgз3-2020-0903-0055>.
7. Ritonga M. The role in SLA played by universal grammar. *Journal of Applied Linguistics*, 2022, vol. 2, no. 1, pp. 23–29. (In Eng.). DOI: <https://doi.org/10.52622/joal.v2i2.76>
8. Fedorenko E., Ivanova A., Dhamala R., Bers M. U. The language of programming: a cognitive perspective. *Trends in Cognitive Sciences*, 2019, vol. 23, no. 7, pp. 525–528. (In Eng.). DOI: <https://doi.org/10.1016/j.tics.2019.04.010>
9. Casalnuovo C., Sagae K., Devanbu P. Studying the difference between natural and programming language corpora. *Empirical Software Engineering*, 2019, vol. 24, pp. 1823–1868. (In Eng.). DOI: <https://doi.org/10.48550/arXiv.1806.02437>
10. Umarlebbe J. H., Said M. S. B., Universal grammar: arguments for its existence. *Advances in Language and Literary Studies*, 2021, vol. 12, no. 31, p. 31. (In Eng.).

The article was submitted 16.05.2022; approved after reviewing 19.06.2023; accepted for publication 24.07.2023.

About the author

Azat I. Gabbitov

Lecturer of the Department of Linguistic and Intercultural Communication, Kazan Federal University (2 Tatarstan St., Kazan 420008, Russian Federation), ORCID: <https://orcid.org/0000-0002-7995-5795>, gabit.azat@gmail.com

Leysan Kh. Shayakhmetova

Ph. D. (Philology), Associate Professor, Associate Professor of the Department of Linguistic and Intercultural Communication, Kazan Federal University (2 Tatarstan St., Kazan 420008, Russian Federation), ORCID: <https://orcid.org/0000-0003-3166-5219>, habirshah@mail.ru

All authors have read and approved the final manuscript.