

Студенческий научный электронный журнал

# StudArctic Forum

## 2022. Т. 7, № 1

**Главный редактор**

И. М. Суворова

**Заместитель главного редактора**

А. А. Малышко

**Ответственный секретарь**

П. С. Воронина

**Редакционный совет**

С. В. Волкова  
М. И. Зайцева  
Г. Н. Колесников  
В. С. Сютёв  
В. А. Шлямин

**Редакционная коллегия**

А. Ю. Борисов  
Р. В. Воронов  
Т. А. Гаврилов  
Е. О. Графова  
Л. А. Девятникова  
А. А. Ившин  
А. А. Кузьменков  
Е. Н. Лузгина  
Ю. В. Никонова  
М. И. Раковская  
А. А. Скоропадская  
Е. И. Соколова  
И. М. Соломещ  
А. А. Шлямина

**Службы поддержки**

Е. В. Голубев  
А. А. Малышко

**Издатель**

ФГБОУ «Петрозаводский государственный университет»  
Российская Федерация, г. Петрозаводск, пр. Ленина, 33

**Адрес редакции**

185910, Республика Карелия, г. Петрозаводск, ул. Ленина, 33.  
E-mail: [saf@petrsu.ru](mailto:saf@petrsu.ru)  
<http://saf.petrso.ru>

Scientific journal  
**StudArctic Forum**

**2022. Vol. 7, No. 1**

**Editor-in-Chief**

Irina M. Suvorova

**Deputy Editor-in-Chief**

Anton A. Malyshko

**Editorial secretary**

Polina S. Voronina

**Editorial Council**

Svetlana V. Volkov  
Maria I. Zaitseva  
Gennadiy N. Kolesnikov  
Vladimir S. Syunev  
Valery A. Shlyamin

**Editorial Team**

Alexey Yu. Borisov  
Roman V. Voronov  
Timmo A. Gavrilov  
Elena O. Grafova  
Lyudmila A. Devyatnikova  
Alexander A. Ivshin  
Alexander Kuzmenkov  
Elena N. Luzgina  
Yulia V. Nikonova  
Marina I. Rakovskaya  
Anna Skoropadskaya  
Evgeniya Sokolova  
Ilya M. Solomeshch  
Anastasia A. Shlyamina

**Support Services**

Evgeniy V. Golubev  
Anton A. Malyshko

**Publisher**

© Petrozavodsk State University, 2012—2022

**Address**

33, Lenin av., 185910 Petrozavodsk, Republic of Karelia, Russia  
E-mail: [saf@petsu.ru](mailto:saf@petsu.ru)  
<http://saf.petsu.ru>

## Информатика и вычислительная техника

**ПАВЛОВ**  
**Максим Павлович**

бакалавриат, Петрозаводский государственный университет  
(Петрозаводск, Российская Федерация)  
*maksim\_pavlov\_2003@list.ru*

### Разработка компьютерного приложения для вычислений в системах счисления

**Научный руководитель:**

Крышень Михаил Александрович  
Статья поступила: 29.03.2022;  
Принята к публикации: 31.03.2022;

**Аннотация.** В статье рассматривается процесс создания компьютерного приложения для работы в системах счисления. Приоритетное внимание уделяется большим и трудоёмким вычислениям. Автор предлагает конкретную реализацию алгоритма перевода и обработки чисел. Проводится их тестирование на точность, скорость, объём максимальных входных данных. В результате исследований создаётся компьютерное приложение с графическим интерфейсом, которое может производить вычислительные операции с числами, значительно превышающими машинные типы данных. Оно может быть использовано в дальнейшем для оптимизации хранения и передачи числовых последовательностей. Рассматриваются будущие сферы применения систем счисления на практике.

**Ключевые слова:** приложение, системы счисления, представление чисел, применение систем счисления, цепной код, числовая последовательность, хранение и обработка данных

**Для цитирования:** Павлов М. П. Разработка компьютерного приложения для вычислений в системах счисления // StudArctic Forum. 2022. Т. 7, № 1. С. 3—7.

Тема «Системы счисления» входит в школьную программу по информатике и начинает изучаться уже с 8 класса. Она включает в себя трудоёмкие вычислительные операции, на проверку которых вручную уходит много времени. Но до сих пор не рассматривалась тема длинной арифметики и перевода большой числовой последовательности в системах счисления. Эти идеи активно используются в криптографии, математике [Будкина], финансах, вычислении общеизвестных трансцендентных чисел с высокой точностью, высококачественном изображении фракталов. Программа, позволяющая производить вычисления и переводы над длинными числовыми последовательностями в системах счисления, может быть полезна для особого круга прикладных задач, когда обычные калькуляторы уже не смогут справиться с объёмом входных данных.

На данный момент существует несколько решений. Первое – длинная арифметика языков программирования Python, Java и других. Они позволяют быстро и эффективно вести подсчёты больших числовых последовательностей, но не поддерживают переводы в системы счисления, превышающие 36. Ещё одним решением является калькулятор командной строки GNU bc. Он позволяет использовать длинную арифметику с возможностью поразрядного вывода в произвольные системы счисления. Однако полученный в системе счисления большей 36 результат повторно использовать не получится: максимальная система счисления входных данных (ibase) не может превышать 36.

Основная цель данной статьи – создать решение, стремящееся совместить быстроту и эффективность длинной арифметики Python с работой в системах счисления больше тридцатишестиричной калькулятора GNU bc.

Главные задачи исследования: на примере данного приложения изучить возможности языка программирования Python; выявить практические сферы применения систем счисления; рассмотреть возможные трудности и нюансы создания приложения.

Для разработки был выбран высокоуровневый язык программирования Python, который позволит: быстро и качественно написать, объединить алгоритмы и элементы кода; создать графический интерфейс программы (в проекте использовалась python библиотека tkinter); собрать проект в исполняемый (exe) файл, готовый к отправке пользователям (при сборке использовалась программа PyInstaller); сохранится возможность добавления новых функций или замены существующих на аналоги, написанные на низкоуровневом языке (например, C++).

Основной экран приложения (Рисунок 1) должен содержать: поля ввода первого и второго числа; списки систем счисления первого, второго числа и результата; кнопки математических операций с введёнными числами (сложение, вычитание, умножение, деление), перевода первого числа, экран вывода результатов. Поля ввода/вывода могут считать последовательность любой длины. Также поддерживается перемещение вправо/влево по введённым данным.

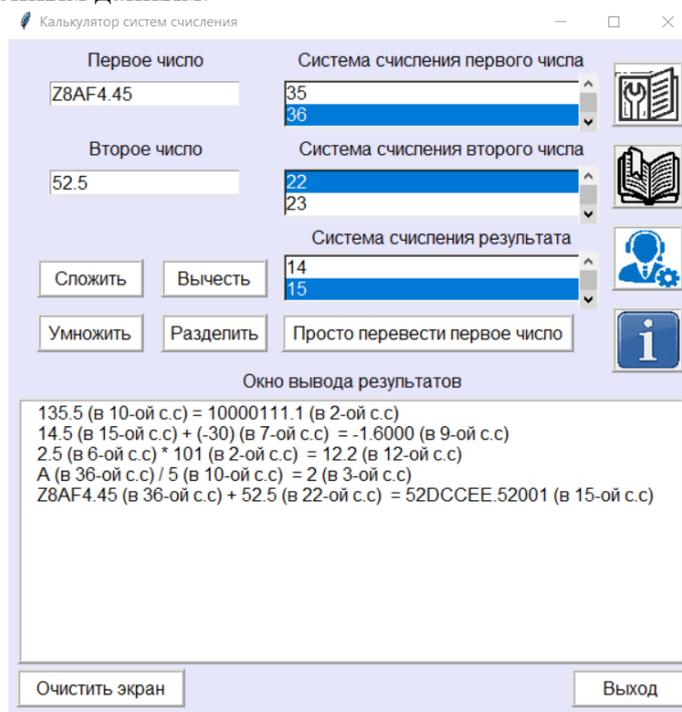


Рис. 1. Экран приложения

Для организации работы приложения, упрощения дальнейших улучшений и поддержания программного кода была создана модульная архитектура ПО (Рисунок 2). Она включает 5 основных частей, реализующих математические операции (сложение, умножение, деление, вычитание и перевод первого числа). Каждый модуль базируется на трёх основных алгоритмах: перевод целой части числа, дробной части в десятичную систему счисления и обратно.

Математические операции осуществляются по одному типовому алгоритму. Введённое число преобразовывается в строковую переменную. По наличию точки определяется наличие дробной части. Строка разбивается на две части: целую и дробную (при наличии). Обе части переводятся в десятичную систему счисления. Определение представления текстовых символов (отличных от цифр) осуществляется с помощью ASCII (разность номера текущего символа в таблице и символа «0») для более приятного восприятия символьного результата. Это позволяет расширить системы счисления дальше тридцатисемиричной (в данной реализации предельная система счисления 93). После символов английского алфавита будут использованы соответствующие символы ASCII (используются лишь однобайтовые символы основной латиницы, начиная с символа «0» и заканчивая «~»). Первый диапазон символов – цифры ASCII (номера 48-57), второй – буквы латинского алфавита в верхнем регистре (65-90), третий – символы и знаки препинания (91-

96), четвертый – буквы латиницы в нижнем регистре (97-122), пятый – оставшиеся однобайтовые символы (32-47, 58-64, 123-126). Разбиение ASCII на такие диапазоны и связь их порядковых номеров с основанием системы счисления обусловлена уже устоявшимися обозначениями. Аналогичные операции производятся и со вторым числом. Затем над целым и дробными частями осуществляются математические операции, обе части результата переводятся в нужную систему счисления и выводятся пользователю. Такой способ реализации основан на общепринятых правилах перевода в системах счисления (совпадает с «ручным» переводом), что позволяет проконтролировать основные этапы вычисления.

В дополнение основным реализованы два модуля, которые раскрывают историю систем счисления, правила математических операций (Информация) и знакомят пользователя с программой (Инструкция). Итоговая программа состоит из 12 модулей, 1016 строк программного кода, из которых 37 пустых и 30 - комментарии.

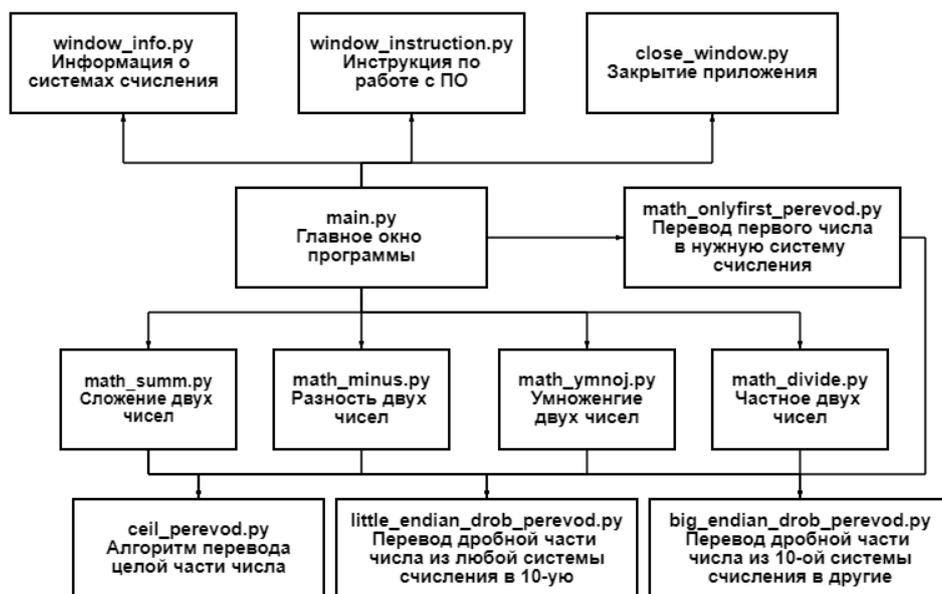


Рис. 2. Архитектура работы приложения

Рассмотрим предельные данные, с которыми может работать алгоритм. Так как ввод организован через текстовое поле, то программа может считать данные достаточно большого объёма. Единственным ограничением будет объём оперативной и постоянной памяти на компьютере. Алгоритм перевода работает лишь со строковыми представлениями, что позволяет обрабатывать данные без потерь точности (целочисленные типы используются лишь для работы с конкретными элементами строк, хранения их длинны). А для записи чисел в разрядах используются символы таблицы ASCII, чтобы рассматривать большие 36 системы счисления. Сложность вычислений возрастает с каждым новым символом, так как результат перевода равен  $\sum_{i=0}^j (a * b^i)$ , где  $j$  – количество чисел в исходном,  $i$  – разряд,  $a$  – цифра в  $i$ -ом разряде,  $b$  – система счисления результата.

При тестировании алгоритма перевода был написан код, который вводил данные в программу из файла input.txt и записывал результаты в файл output.txt, также фиксируя время работы программы. Для конкретности и определённости сравнения полученных результатов с эталоном все символы файла input.txt состоят из символов «Z» (соответствует числу 35) и переводятся из тридцатишестиричной системы счисления в десятичную. Результаты работы алгоритма стоит считать верными. Объём символов вывода входит в правильный диапазон, определённый количеством чисел в большей степени числа ( $N = [a * \lg(b)] + 1$ , где  $N$  – количество цифр,  $a$  – степень,  $b$  – возводимое в степень число). Проанализировав результат, можно сделать вывод - последние цифры верны. Время работы приемлемое для такого количества вычислительных операций.

Таблица 1. Результаты тестирования алгоритма

Число символов «Z» в input.txt	Время работы алгоритма с выводом результата, сек.	Кол-во символов в файле output.txt	Результат аналитической и ручной проверки.
1	0,109 – 0,114	2	Совпадает с результатом, полученным вручну.
10	0,108 – 0,113	16	
100	0,113 – 0,119	156	
1000	0,116 – 0,121	15557	Кол-во символов результата входит в верный диапазон, последние цифры совпадают с полученными аналитически.
10000	1,1 – 1,2	15564	
100000	356	155631	
200000	2061,9	311261	
300000	5819,1	466891	

Данная программа может использоваться как для проверки небольших вычислений, сделанных вручну, так и для длинной арифметики с высокой точностью в различных системах счисления (включая привычную десятичную). Возможно дальнейшее добавление дополнительных функций, реализующих другие математические операции (возведение в степень, вычисление факториала числа и другие), поддержание работы приложения, его оптимизация (например, перевод алгоритма работы на низкоуровневый язык) и увеличение максимальной поддерживаемой системы счисления путём использования диапазонов символов Unicode.

Системы счисления активно использовались на этапах зарождения ЭВМ, телеграфии (кодирование букв и чисел последовательностью нулей и единиц), создания штрих-кодов [Гашков]. Но в наши дни одна из главных сфер применения данного приложения – кодирование (сжатие) числовой последовательности (например, цепного кода). «...Любую упорядоченную последовательность можно представить в виде ряда упорядоченных множеств и для любого упорядоченного множества можно создать свою позиционную систему счисления...» [Лебедева : 21] Рационально использовать данный алгоритм и для отправки данных, так как при передаче данных используются другие системы счисления (например, кодирование base64), когда есть дополнительные ограничения на то, какие значения могут иметь передаваемые байты (например, это должны быть только печатные символы ASCII). Предположим, есть последовательность из 15565 десятичных чисел в текстовом формате (файл output.txt в результате перевода 10000 «Z»). Результат перевода данной последовательности из десятичной в восьмидесятеричную систему счисления составил 8179 символов, а с помощью base64 - 20755 символов. Гораздо больше исходного файла, так как три восьмерки битов сводятся к четырем шестеркам в виде символов ASCII. Но данный подход всегда менее эффективен, чем использование двоичного формата.

### СПИСОК ЛИТЕРАТУРЫ

Будкина И. М. Научному прогрессу - творчество молодых: материалы XV международной молодежной научной конференции по естественнонаучным и техническим дисциплинам (Йошкар-Ола, 17-18 апреля 2020 года): в 2 ч. / редкол.: Д. В. Иванов [и др // Системы счисления. Йошкар-Ола: Поволжский государственный технологический университет, 2020. Ч. 1. С. 14—16.

Гашков С. Б. Системы счисления и их применение. Москва : МЦНМО, 2004. 52 с.

Лебедева А. В. Система счисления рядов упорядоченных множеств // Математика и математическое моделирование: сборник материалов XII всероссийской молодежной научно-инновационной школы. Саратов: Интерконтакт, 2018. С. 21—22.

Maksim P. PAVLOV

bachelor, Petrozavodsk State University  
(Petrozavodsk, Russian Federation)  
*maksim\_pavlov\_2003@list.ru*

## Digital transformation of the bidding procedure

**Scientific adviser:**

Michael A. Kryshen

Paper submitted on: 03/29/2022;

Accepted on: 03/31/2022;

**Abstract.** The article analyzes the process of creating software for working in number systems. Large and time-consuming calculations is in the priority . The author considers a particular implementation of the algorithm for translating and processing numbers. It is tested for accuracy, speed working and the maximum volume of input data. The result of the research is computer software with a graphical interface which can perform computational operations on numbers far in excess of machine data types. In the future it can be used to optimize holding and processing of numeric sequences. This is vital and very important for posterity. The memory on devices is not unlimited and requires a lot of natural resources for its production.

**Keywords:** software, number systems, representation of numbers, application of number systems, chain code, numerical sequence, data holding and processing

**For citation:** Pavlov M. P. Development of a computer application for computing in number systems. *StudArctic Forum*. 2022; 7(1): 3—7.

### REFERENCES

Budkina I. M. Sistemy schisleniia [Number systems]. Nauchnomu progressu - tvorchestvo molodykh: materialy XV mezhdunarodnoi molodezhnoi nauchnoi konferentsii po estestvennonauchnym i tekhnicheskim distsiplinam (Yoshkar-Ola, 17-18 apreliia 2020 goda). Yoshkar-Ola, Volga State University of Technology Publ., 2020. Part. 1. P. 14–16. (in Russian)

Gashkov S. B. Sistemy schisleniia i ikh primenenie [Number systems and their application]. Moscow, Moscow Center for Continuing Mathematical Education Publ., 2004. 52 p. (in Russian)

Lebedeva A. V., Spliukhin D. V., Martynova I. A. Sistema schisleniia riadov uporiadochennykh mnozhestv [The number system of series of ordered sets]. Matematika i matematicheskoe modelirovanie: sbornik materialov XII vserossiiskoi molodezhnoi nauchno-innovatsionnoi shkoly. Saratov, Interkontakt Publ., 2018. P. 21—22. (in Russian).