

ПРОЕКТИРОВАНИЕ ПРОГРАММ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ ДЛЯ РАБОТЫ С КОМПЬЮТЕРНЫМИ СИСТЕМАМИ СЧИСЛЕНИЯ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Информатика

Калёнов А.Д.

*г. Тверь, Академическая гимназия Тверского государственного университета,
11 класс*

*Научный руководитель: Наумова А.И., г. Тверь, преподаватель информатики
высшей категории, Академическая гимназия Тверского государственного
университета*

Введение

Данная статья является реферативным изложением основной работы. Полный текст научной работы, иллюстрации и иные дополнительные материалы доступны на сайте XXIV Международного конкурса научно-исследовательских и творческих работ учащихся “Старт в науке” по ссылке: <https://school-science.ru/24/4/61027>.

В данной работе представлен материал по разработке проекта “Проектирование программ с графическим интерфейсом для работы с компьютерными системами счисления на языке программирования Python”. Тема достаточно *актуальная* и представляет *повышенный интерес для учащихся профильных классов*.

Цель данной работы заключается в том, чтобы получить *дополнительные знания* по работе с компьютерными системами счисления и провести анализ полученных результатов.

Задача состоит в том, чтобы подобрать *соответствующий материал* с последующей систематизацией, обобщением и иллюстрацией текста.

Работа состоит из двух частей: *описательной* (даны описание компьютерных систем счисления, преобразование компьютерных чисел) и *практической* (приведён пример разработки калькулятора с двумя целыми

положительными числами и рассмотрены *варианты перевода исходных данных и полученного результата в компьютерные системы счисления с использованием соответствующих функций*).

Описание компьютерной системы счисления

Определение систем счисления

“Всё есть число”, - говорили пифагорейцы, подчёркивая необычайно *важную роль чисел* в практической деятельности. Известно множество способов представления чисел. В любом случае число изображается символом или группой символов (словом) некоторого алфавита. Будем называть такие символы *цифрами*. Для представления чисел используются непозиционные и позиционные системы счисления [7].

Система счисления – это правила записи чисел с помощью специальных знаков – цифр, а также соответствующие правила выполнения операций с этими числами.

Позиционная система счисления – это такая система, в которой значение цифры (её вес) полностью определяется её местом (позицией) в записи числа.

Алфавит системы счисления – это используемый в ней набор цифр.

Разряд – это позиция цифры в записи числа. Разряды в записи целых чисел нумеруются с *нуля справа налево* [3].

В данной работе представлен материал по работе с *компьютерными системами счисления*.

Десятичная система счисления

Десятичная система счисления имеет основание 10, поскольку в ней используются 10 цифр от 0 до 9. В десятичной системе счисления последовательные позиции слева от десятичной точки представляют единицы, десятки, сотни, тысячи и так далее. Каждая позиция представляет определённую степень основания (10). Например, десятичное число 1234 состоит из цифры 4 в позиции единиц, 3 в позиции десятков, 2 в позиции сотен и 1 в позиции тысяч. Его значение можно записать как:

$$(1 \times 1000) + (2 \times 100) + (3 \times 10) + (4 \times 1)$$

$$(1 \times 10^3) + (2 \times 10^2) + (3 \times 10^1) + (4 \times 10^0)$$

$$1000 + 200 + 30 + 4 = 1234.$$

Двоичная система счисления

Все данные в компьютерных устройствах хранятся и обрабатываются как числа, представленные в *двоичной системе счисления*.

Характеристика двоичной системы счисления:

- Использует две цифры: 0 и 1;
- Называется системой счисления с основанием 2;
- Каждая позиция в двоичном числе представляет собой определённую степень основания (2). Пример 2^0 ;
- Последняя позиция в двоичном числе представляет собой x степень основания (2). Пример 2^x , где x представляет последнюю позицию – (количество цифр в числе минус 1);

Пример

Двоичное число: 10101_2

Вычисление десятичного эквивалента:

Шаг	Двоичное число	Десятичное число
1	10101_2	$((1 \cdot 2^4) + (0 \cdot 2^3) + (1 \cdot 2^2) + (0 \cdot 2^1) + (1 \cdot 2^0))_{10}$
2	10101_2	$(16 + 0 + 4 + 0 + 1)_{10}$
3	10101_2	21_{10}

Примечание: 10101_2 обычно записывается как 10101 .

Восьмеричная система счисления

Восьмеричная система счисления применяется для *краткой записи двоичных кодов*.

Характеристика восьмеричной системы счисления:

- Использует восемь цифр: 0, 1, 2, 3, 4, 5, 6, 7;
- Также называется системой счисления с базой 8;
- Каждая позиция в восьмеричном числе представляет собой определённую степень основания (8). Пример 8^0 ;

- Последняя позиция в восьмеричном числе представляет собой x степень основания (8). Пример 8^x , где x представляет последнюю позицию – (количество цифр в числе минус 1);

Пример

Восьмеричное число: 12570_8

Вычисление десятичного эквивалента:

Шаг	Восьмеричное число	Десятичное число
1	12570_8	$((1 \cdot 8^4) + (2 \cdot 8^3) + (5 \cdot 8^2) + (7 \cdot 8^1) + (0 \cdot 8^0))_{10}$
2	12570_8	$(4096 + 1024 + 320 + 56 + 0)_{10}$
3	12570_8	5496_{10}

Примечание: 12570_8 обычно записывается как 12570.

Шестнадцатеричная система счисления

Шестнадцатеричная система счисления используется для записи адресов и содержимого ячеек памяти компьютера. Её алфавит содержит 16 цифр, 10 арабских цифр (0..9) и первые шесть буквы латинского алфавита (A=10, B=11, C=12, D=13, E=14, F=15).

Характеристика шестнадцатеричной системы счисления:

- Использует 10 цифр и 6 букв: 0, 1, 2, 3, 4, 5, 6, 7, A, B, C, D, E, F;
- Буквы обозначают числа, начинающиеся с 10. A=10, B=11, C=12, D=13, E=14, F=15;
- Также называется системой счисления с базой 16;
- Каждая позиция в шестнадцатеричном числе представляет собой определённую степень основания (16). Пример: 16^0 ;
- Последняя позиция в шестнадцатеричном числе представляет собой x степень основания (16). Пример 16^x , где x представляет последнюю позицию – (количество цифр в числе минус 1);

Пример

Шестнадцатеричное число: $19FDE_{16}$

Вычисление десятичного эквивалента:

Шаг	Шестнадцатеричное число	Десятичное число
1	19FDE ₁₆	$((1 \cdot 16^4) + (9 \cdot 16^3) + (F \cdot 16^2) + (D \cdot 16^1) + (E \cdot 16^0))_{10}$
2	19FDE ₁₆	$((1 \cdot 16^4) + (9 \cdot 16^3) + (15 \cdot 16^2) + (13 \cdot 16^1) + (14 \cdot 16^0))_{10}$
3	19FDE ₁₆	$(65536 + 36864 + 3840 + 208 + 14)_{10}$
4	19FDE ₁₆	106462 ₁₀

Примечание: 19FDE₁₆ обычно записывается как 19FDE [6].

Преобразование компьютерных чисел

Десятичная система в другую базовую систему

Шаг 1 – Разделите десятичное число, подлежащее преобразованию, на значение нового основания.

Шаг 2 – Получите остаток от шага 1 как крайнюю правую цифру (наименьшую значащую цифру) нового базового числа.

Шаг 3 – Разделите частное от предыдущего деления на новое основание.

Шаг 4 – Запишите остаток от шага 3 в качестве следующей цифры (слева) нового базового числа.

Повторяйте Шаги 3 и 4, получая остатки справа налево, пока частное не станет равным нулю на Шаге 3.

Последний остаток, полученный таким образом, будет самой значащей цифрой (MSD) нового базового числа.

Пример

Десятичное число: 29₁₀

Вычисление двоичного эквивалента:

Шаг	Операция	Результат	Остаток
Шаг 1	29 / 2	14	1
Шаг 2	14 / 2	7	0
Шаг 3	7 / 2	3	1
Шаг 4	3 / 2	1	1
Шаг 5	1 / 2	0	1

Как упоминалось на этапах 2 и 4, остатки должны быть расположены в *обратном порядке* так, чтобы первый остаток стал *наименее значащей* цифрой (LSD), а последний остаток стал *наиболее значащей* цифрой (MSD).

Десятичное число: 29_{10} = Двоичное число: 11101_2 .

Другая базовая система в десятичную

Шаг 1 – Определите столбчатое (позиционное) значение каждой цифры (это зависит от положения цифры и основания системы счисления).

Шаг 2 – Умножьте полученные значения столбцов (на шаге 1) на цифры в соответствующих столбцах.

Шаг 3 – Суммируйте произведения, рассчитанные на шаге 2. Итогом является эквивалентное значение в десятичной системе счисления.

Пример

Двоичное число: 11101_2

Вычисление десятичного эквивалента:

Шаг	Двоичное число	Десятичное число
Шаг 1	11101_2	$((1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0))_{10}$
Шаг 2	11101_2	$(16 + 8 + 4 + 0 + 1)_{10}$
Шаг 3	11101_2	29_{10}

Двоичное число: 11101_2 = Десятичное число: 29_{10} .

Быстрый метод – преобразование двоичной системы счисления в восьмеричную

Шаг 1 – Разделите двоичные цифры на группы по три (начиная справа).

Шаг 2 – Преобразуйте каждую группу из трёх двоичных цифр в одну восьмеричную.

Пример

Двоичное число: 10101_2

Вычисление восьмеричного эквивалента:

Шаг	Двоичное число	Восьмеричное число
1	10101 ₂	010 101
2	10101 ₂	2 ₈ 5 ₈
3	10101 ₂	25 ₈

Двоичное число: 10101₂ = Восьмеричное число: 25₈.

Быстрый метод – преобразование восьмеричной системы счисления в двоичную

Шаг 1 – Преобразуйте каждую восьмеричную цифру в *трехзначное двоичное* число (восьмеричные цифры для этого преобразования могут рассматриваться как десятичные).

Шаг 2 – Объедините все полученные двоичные группы (по 3 цифры в каждой) в одно двоичное число.

Пример

Восьмеричное число: 25₈

Вычисление двоичного эквивалента:

Шаг	Восьмеричное число	Двоичное число
1	25 ₈	2 ₁₀ 5 ₁₀
2	25 ₈	010 ₂ 101 ₂
3	25 ₈	010101 ₂

Восьмеричное число: 25₈ = Двоичное число: 10101₂.

Быстрый метод – преобразование двоичной системы счисления в шестнадцатеричную

Шаг 1 – Разделите двоичные цифры на группы по четыре (начиная справа).

Шаг 2 – Преобразуйте каждую группу из четырех двоичных цифр в один шестнадцатеричный символ.

Пример

Двоичное число: 10101₂

Вычисление шестнадцатеричного эквивалента:

Шаг	Двоичное число	Шестнадцатеричное число
1	10101 ₂	0001 0101
2	10101 ₂	1 ₁₆ 5 ₁₆
3	10101 ₂	15 ₁₆

Двоичное число: 10101₂ = Шестнадцатеричное число: 15₁₆.

Быстрый метод – преобразование шестнадцатеричной системы счисления в двоичную

Шаг 1 – Преобразуйте каждую шестнадцатеричную цифру в 4-значное двоичное число (при этом шестнадцатеричные цифры могут рассматриваться как десятичные).

Шаг 2 – Объедините все полученные двоичные группы (по 4 цифры в каждой) в одно двоичное число.

Пример

Шестнадцатеричное число: 15₁₆

Вычисление двоичного эквивалента:

Шаг	Шестнадцатеричное число	Двоичное число
1	15 ₁₆	1 ₁₀ 5 ₁₀
2	15 ₁₆	0001 ₂ 0101 ₂
3	15 ₁₆	00010101 ₂

Шестнадцатеричное число 15₁₆ = Двоичное число 10101₂ [6].

Проектирование программы с графическим интерфейсом

Содержательная постановка задачи

Большинство современных программ, предназначенных для пользователя, управляются с помощью *графического интерфейса*. Рассмотрим основные принципы построения таких программ на языке Python.

Рассмотренные преобразования компьютерных чисел можно выполнять в *компьютерном варианте*, выполнив разработку проекта “Калькулятор для работы с компьютерными системами счисления”.

Формальная модель

В проекте первоначально предусмотрена разработка *обычного* калькулятора: выполнение арифметических операций с *двумя целыми положительными числами*: сложение (+), вычитание (-), умножение (*) и деление (/) [5]; далее рассмотрен вариант *перевода* исходных данных и полученного результата в *двоичную, восьмеричную и шестнадцатеричную* системы счисления с использованием соответствующих функций: *bin()*, *oct()*, *hex()*.

Компьютерная модель

Разработка графического интерфейса проекта

В программе может быть несколько окон, которые называются *формами*. На форме размещаются *элементы* графического интерфейса: поля ввода, кнопки и др., которые называются *виджетами* (англ. widget – элемент) или (как в большинстве аналогичных сред) *компонентами*. Каждый компонент – это объект определённого класса, у которого есть *свойства и методы* [4].

В отличие от других объектно-ориентированных языков программирования создание формы (окна) с компонентами и их свойствами *прописываются непосредственно в программе* с использованием графической библиотеки *Tkinter*.

Программа (*скрипт*) с компонентами (*виджетами*) на языке Python

Для ввода и вывода информации создать компоненты *Entry* (текстовые окна) - 13, для добавления текста – компоненты *Label* (метки) - 17, для выполнения программы – командную кнопку *Button* [1, 2].

```
#Проектирование программ с графическим интерфейсом
```

```
#Graphical user interface - GUI
```

```
#Создать Приложение-Калькулятор с арифметическими операциями
```

```
#для работы с компьютерными системами счисления
```

```
#Создать Форму с компонентами (виджетами)
```

```
#Подключить графическую библиотеку Tkinter
```

```
from tkinter import *
```

```

import tkinter
top=tkinter.Tk()
#Функция пользователя для командной кнопки (скрипт - программный код)
def process():
    L2=Entry.get(E1);L3=Entry.get(E2);L4=Entry.get(E3);L7=Entry.get(E5)
    L8=Entry.get(E6);L9=Entry.get(E7);L11=Entry.get(E8);L12=Entry.get(E9)
    L13=Entry.get(E10);L14=Entry.get(E11); L15=Entry.get(E12);L16=Entry.get(E13)
    #Преобразовать тип данных (строковую переменную в числовую)
    L2=int(L2); L3=int(L3)
    #Арифметические операции
    if L4 == "+":L5=L2+L3
    if L4 == "-":L5=L2-L3
    if L4 == "*":L5=L2*L3
    if L4 == "/":L5=int(L2/L3)
    #Перевести в 2-ную систему счисления
    L7=bin(L2); L8=bin(L3);L9=bin(L5)
    #Перевести в 8-ную систему счисления
    L11=oct(L2);L12=oct(L3);L13=oct(L5)
    #Перевести в 16-ную систему счисления
    L15=hex(L2); L16=hex(L3); L17=hex(L5)
    #Занести полученный результат (10 с/с) в текстовое окно E4
    Entry.insert(E4,0,L5)
    #Занести полученные результаты (2 с/с) в текстовые окна E5,E6,E7
    Entry.insert(E5,0,L7);Entry.insert(E6,0,L8);Entry.insert(E7,0,L9)
    #Занести полученные результаты (8 с/с) в текстовые окна E8,E9,E10
    Entry.insert(E8,0,L11);Entry.insert(E9,0,L12);Entry.insert(E10,0,L13)
    #Занести полученные результаты (16 с/с) в текстовые окна E11,E12,E13
    Entry.insert(E11,0,L15);Entry.insert(E12,0,L16); Entry.insert(E13,0,L17)
    #Полученный результат выдать в окне выполнения программы
    print("Исходные данные и результат в десятичной системе: ")
    print(L2,L3,L5)

```

```

print("Исходные данные и результат в двоичной системе:")
print(L7,L8,L9)
print("Исходные данные и результат в восьмеричной системе:")
print(L11,L12,L13)
print("Исходные данные и результат в шестнадцатеричной системе:")
print(L15,L16,L17);print()
#Для выполнения следующей арифметической операции в текстовые окна E4, E5
#E6, E7, E8, E9, E10, E11, E12, E13 занести пробелы между числами
Entry.insert(E4,0," ")
Entry.insert(E5,0," ");Entry.insert(E6,0," ");Entry.insert(E7,0," ")
Entry.insert(E8,0," ");Entry.insert(E9,0," ");Entry.insert(E10,0," ")
Entry.insert(E11,0," ");Entry.insert(E12,0," "); Entry.insert(E13,0," ")
#Создать интерфейс для десятичной системы счисления
#Создать 6 компонентов Label (метка): название, № позиций (строка, столбец)
#Позиции устанавливаются с помощью функции grid()
L1=Label(top,text="Перевод чисел",).grid(row=0,column=1)
L1=Label(top,text="Десятичная система счисления",).grid(row=1,column=1)
L2=Label(top,text="Первое число",).grid(row=2,column=1)
L3=Label(top,text="Второе число",).grid(row=2,column=2)
L4=Label(top,text="Операция",).grid(row=2,column=3)
L4=Label(top,text="Результат",).grid(row=2,column=4)
#Для ввода информации создать 4 компонента Entry (текстовое окно)
E1=Entry(top,bd =5);E1.grid(row=3,column=1)
E2=Entry(top,bd =5);E2.grid(row=3,column=2)
E3=Entry(top,bd =5);E3.grid(row=3,column=3)
E4=Entry(top,bd =5);E4.grid(row=3,column=4)
#Создать интерфейс для двоичной системы счисления
#Создать 4 компонента Label (метка): название, № позиций (строка, столбец)
#Позиции устанавливаются с помощью функции grid
L6=Label(top,text="Двоичная система счисления",).grid(row=5,column=1)
L7=Label(top,text="Первое число",).grid(row=6,column=1)

```

```

L8=Label(top,text="Второе число",).grid(row=6,column=2)
L9=Label(top,text="Результат",).grid(row=6,column=4)
#Для ввода информации создать 3 компонента Entry (текстовое окно)
E5=Entry(top,bd =5);E5.grid(row=7,column=1)
E6=Entry(top,bd =5);E6.grid(row=7,column=2)
E7=Entry(top,bd =5);E7.grid(row=7,column=4)
#Создать интерфейс для восьмеричной системы счисления
#Создать 4 компонента Label (метка): название, № позиций (строка, столбец)
#Позиции устанавливаются с помощью функции grid
L11=Label(top,text="Восьмеричная система счисления",).grid(row=11,column=1)
L11=Label(top,text="Первое число",).grid(row=12,column=1)
L12=Label(top,text="Второе число",).grid(row=12,column=2)
L13=Label(top,text="Результат",).grid(row=12,column=4)
#Для ввода информации создать 3 компонента Entry (текстовое окно)
E8=Entry(top,bd =5);E8.grid(row=13,column=1)
E9=Entry(top,bd =5);E9.grid(row=13,column=2)
E10=Entry(top,bd =5);E10.grid(row=13,column=4)
#Создать интерфейс для шестнадцатеричной системы счисления
#Создать 4 компонента Label (метка): название, № позиций (строка, столбец)
#Позиции устанавливаются с помощью функции grid
L15=Label(top,text="Шестнадцатеричная система
счисления",).grid(row=14,column=1)
L15=Label(top,text="Первое число",).grid(row=15,column=1)
L16=Label(top,text="Второе число",).grid(row=15,column=2)
L17=Label(top,text="Результат",).grid(row=15,column=4)
#Для ввода информации создать 3 компонента Entry (текстовое окно)
E11=Entry(top,bd =5);E11.grid(row=16,column=1)
E12=Entry(top,bd =5);E12.grid(row=16,column=2)
E13=Entry(top,bd =5);E13.grid(row=16,column=4)
#Создать командную кнопку Button, ввести её название - Выполнить
B=Button(top,text="Выполнить",command = process).grid(row=22,column=3)

```

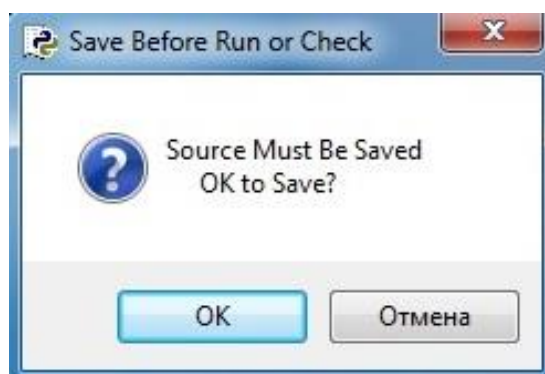
#Для работы с окнами (без функции `mainloop()`) на экране проект не отобразится)
`top.mainloop()`

Тестовые примеры (таблица входных и выходных данных)

Первое число	Второе число	Операция	Результат
Тестовый пример № 1			
12	3	+	15
1100	11		1111
14	3		17
c	3		f
Тестовый пример № 2			
12	3	*	36
1100	11		100100
14	3		44
c	3		24

Компьютерный эксперимент

1. Войти в программную среду Пайтон, выполнив команды *File – New File*.
2. В текстовом редакторе набрать исходный модуль программы (скрипт).
3. Запустить программу на выполнение: *Run – Run Module (F5)*.
4. В появившемся окне сохранить модуль, указав имя программы и место сохранения:



5. Ввести значения первого и второго числа, арифметическую операцию и нажать на командную кнопку *Выполнить*. Результат будет получен как на форме (рис. 1), так и в окне программной среды (рис. 2).

tk

Перевод чисел

Десятичная система счисления

Первое число: 12 Второе число: 3 Операция: * Результат: 36 15

Двоичная система счисления

Первое число: 0b1100 0b1100 Второе число: 0b11 0b11 Результат: 0b100100 0b1111

Восьмеричная система счисления

Первое число: 0o14 0o14 Второе число: 0o3 0o3 Результат: 0o44 0o17

Шестнадцатеричная система счисления

Первое число: 0xc 0xc Второе число: 0x3 0x3 Результат: 0x24 0xf

Выполнить

Рис. 1. Операция сложения, умножения и перевод чисел из десятичного эквивалента в другие системы счисления на форме

```

Исходные данные и результат в десятичной системе:
12 3 15
Исходные данные и результат в двоичной системе:
0b1100 0b11 0b1111
Исходные данные и результат в восьмеричной системе:
0o14 0o3 0o17
Исходные данные и результат в шестнадцатеричной системе:
0xc 0x3 0xf

Исходные данные и результат в десятичной системе:
12 3 36
Исходные данные и результат в двоичной системе:
0b1100 0b11 0b100100
Исходные данные и результат в восьмеричной системе:
0o14 0o3 0o44
Исходные данные и результат в шестнадцатеричной системе:
0xc 0x3 0x24

Исходные данные и результат в десятичной системе:
12 3 4
Исходные данные и результат в двоичной системе:
0b1100 0b11 0b100
Исходные данные и результат в восьмеричной системе:
0o14 0o3 0o4
Исходные данные и результат в шестнадцатеричной системе:
0xc 0x3 0x4

Исходные данные и результат в десятичной системе:
12 3 9
Исходные данные и результат в двоичной системе:
0b1100 0b11 0b1001
Исходные данные и результат в восьмеричной системе:
0o14 0o3 0o11
Исходные данные и результат в шестнадцатеричной системе:
0xc 0x3 0x9

```

Рис. 2. Окно выполнения программы с арифметическими операциями (сложение, умножение, деление и вычитание)

Анализ полученных результатов

Во всех системах счисления (кроме десятичной) в полученных результатах числа обозначены *префиксами (начало)*, которые указывают на *использованную* систему счисления:

- 0b → соответствует *двоичной* системе счисления (b → binary);
- 0o → соответствует *восьмеричной* системе счисления (o → octal);
- 0x → соответствует *шестнадцатеричной* системе счисления (x → hexadecimal).

После ввода двух целых положительных чисел, арифметической операции и нажатия на кнопку *Выполнить* в окне выполнения на форме в соответствующих текстовых окнах появятся *полученные результаты*. Удалив информацию (исходные данные, знак операции) в первой строке (десятичная система счисления), можно *заново ввести два числа* (или оставить предыдущие), ввести *знак другой операции* и вновь нажать на кнопку *Выполнить*. В этом случае появятся *новые результаты* выполнения программы. На форме в окнах для систем счисления (рис. 1) *новые результаты* появятся с *левой стороны*; предыдущий результат *сохраняется* и сдвигается через пробел вправо. В окне выполнения программы в среде программирования исходные данные и результат выданы *отдельными блоками* (рис. 2).

Таким образом, рассмотренный вариант работы на языке программирования Python для проектирования программ с *графическим интерфейсом* можно применять на уроках информатики в старших классах инженерного, информационно-технологического, физико-математического и естественно-научного профилей.

Заключение

В данной работе показано проектирование программы с *графическим интерфейсом*. Приведены примеры арифметических операций с *целыми положительными числами* и последующим переводом десятичных чисел в эквиваленты компьютерных систем счисления.

В ходе разработки проекта получены дополнительные знания по теме “Компьютерные системы счисления”. Проведённый компьютерный эксперимент наглядно показывает *практическую значимость* проведённых исследований.

Python – современный высокоуровневый профессиональный язык программирования, изучение которого начинается в *профильных классах* общеобразовательных учреждений (элективные занятия по курсу: “Программирование”) на примере использования не только его основных конструкций, но и *его основных объектов и их свойств*, что достаточно наглядно показано в данной работе.

Программное обеспечение

1. Операционная система Windows 10
2. Среда программирования Python – 3.8.0
3. Приложение Microsoft Office Word 2016

Список использованных источников и литературы

1. Tkinter — создание графического интерфейса в Python [Электронный ресурс]. – Режим доступа: <https://python-scripts.com/tkinter>.
2. Делаем первое программное обеспечение на Python [Электронный ресурс]. – Режим доступа: <https://arduinoplus.ru/delaem-po-na-python/>.
3. Информатика (базовый и углублённый уровни) (в 2 частях). 10 класс. Ч. 1 : учебник / К.Ю. Поляков, Е.А. Еремин. – М. : БИНОМ. Лаборатория знаний, 2019. – 352 с.: ил.
4. Информатика. 11 класс (базовый и углублённый уровни) (в 2 частях): учебник. Ч. 2 / К.Ю. Поляков, Е.А. Еремин. – М. : БИНОМ. Лаборатория знаний, 2019. – 304 с.: ил.
5. Исследование информационных моделей. Элективный курс: Учебное пособие / Н.Д. Угринович – М.: БИНОМ. Лаборатория знаний, 2004. – 183 с.
6. Компьютерные системы счисления [Электронный ресурс]. – Режим доступа: https://translated.turbopages.org/proxy_u/en-ru.ru.f7b511a8-678b7043-eaee4213-74722d776562/https/www.tutorialspoint.com/computer_fundamentals/computer_number_system.htm
7. Практикум по информатике и информационным технологиям: Учебное пособие для общеобразовательных учреждений / Н. Д. Угринович, Л. Л. Босова, Н. И. Михайлова. – М.: БИНОМ. Лаборатория знаний, 2005. – 394 с.: ил.