

Проектная работа
Создание конвертера единиц измерения
информации средствами языка программирования
Python

Авторы:

Батуев Максим Александрович,
обучающийся 8Б класса

МБОУ СОШ № 2 г. Советский
Золотов Николай Михайлович,
обучающийся 8Б класса

МБОУ СОШ № 2 г. Советский

Руководитель:

Фридрих Татьяна Анатольевна,
учитель информатики

МБОУ СОШ № 2 г. Советский

Содержание

Введение.....	3
Глава 1. Единицы измерения информации.....	4
Глава 2. Создание графического интерфейса средствами языка программирования Python.	5
Глава 3. Разработка программы.....	10
Глава 4. Результат проекта.....	18
Список литературы.	19
Приложение 1	20

Введение

IT-отрасль — перспективная индустрия. В последние годы цифровизация коснулась всех отраслей экономики. Взять талон в больницу, купить билет на поезд, заказать справку, оформить заказ продуктов и доставку, узнать отметки и домашнее задание, все эти и многие другие услуги стали доступны в электронном виде, — поэтому в последнее время востребованность программистов очень высока. В этом году мы начали изучать программирование и посещаем курс дополнительного образования «Основы программирования на языке Python».

Python – один из самых популярных языков программирования, используемый как для разработки самостоятельных программ, так и для создания прикладных сценариев в самых разных областях применения. Это мощный, переносимый, простой в использовании и свободно распространяемый язык. Программисты, работающие в самых разных областях, считают, что ориентация Python на эффективность разработки и высокое качество программного обеспечения дает ему стратегическое преимущество, как в маленьких, так и в крупных проектах.

Разработчики программного обеспечения используют язык Python, выбрав его за гибкость, простоту использования и обеспечиваемую им высокую скорость разработки. Он позволяет создавать эффективные и надежные проекты, которые легко интегрируются с программами и инструментами, написанными на других языках. Применять язык Python можно в самых разных областях. Python используется практически каждой достаточно крупной организацией, занимающейся разработкой программного обеспечения.

В ходе изучения языка Python мы решили попробовать себя в роли разработчика программного обеспечения. Приложение, которое мы собрались создать, переводит единицы изменения информации из одной в другую. Для разработки приложения был взят материал учебника ООО «БИНОМ. Лаборатория знаний», 2019 г. Босова Л.Л. Информатика. 7класс, тема «Измерение информации». Актуальность этой темы в том, что человечество с каждым днём увеличивает объем информации многократно. Возможность контролировать информацию, измерять и сохранять - является основной задачей 21 века. Данная тема связана с вычислениями и у некоторых обучающихся вызывает затруднения, поэтому было решено создать конвертер единиц измерения информации, как вспомогательную программу для обучающихся школы.

Цель проекта: Создание конвертера единиц измерения информации средствами языка программирования Python.

Задачи проекта:

1. Изучить информационные источники о единицах измерения информации.
2. Рассмотреть возможности создания графического интерфейса средствами языка программирования Python.
3. Разработать макет окна приложения.
4. Создать программу на языке программирования Python.
5. Протестировать программу.

Глава 1. Единицы измерения информации

Информация является одним из фундаментальных понятий современной науки наряду с такими понятиями, как «вещество» и «энергия». Общее определение этому термину дать невозможно. Выделяют два подхода определения понятия «информация».

Определение 1. В содержательном подходе, информация - это снятая неопределённость. Неопределённость некоторого события - это количество возможных результатов (исходов) данного события.

Определение 2. В алфавитном подходе информация - это сообщение (последовательность символов некоторого алфавита). Причём существенными являются только размер алфавита и количество символов в сообщении. Конкретное содержание сообщения интереса не представляет. Чаще всего алфавит является двоичным (состоит из двух символов – «0» и «1»).

Мы будем рассматривать алфавитный подход к измерению информации.

В качестве основной единицы измерения информации используется бит.

При алфавитном подходе один бит - это количество информации, которое можно передать в сообщении, состоящем из одного двоичного знака («0»или «1»). Единица измерения информации бит (bit) - сокращение от английских слов binary digit, что означает двоичная цифра.

В компьютерной технике бит соответствует физическому состоянию носителя информации: намагничено - не намагничено, есть отверстие - нет отверстия. При этом одно состояние принято обозначать цифрой 0, а другое - цифрой 1. Выбор одного из двух возможных вариантов позволяет также различать логические истину и ложь. Последовательностью битов можно закодировать текст, изображение, звук или какую-либо другую информацию. Такой метод представления информации называется двоичным кодированием (binary encoding).

Глава 2. Создание графического интерфейса средствами языка программирования Python.

В информатике часто используется величина, называемая байтом (byte) и равная 8 битам. Наряду с байтами для измерения количества информации используются более крупные единицы:

1 Кбайт (килобайт) = 2^{10} байт = 1024 байта;

1 Мбайт (мегабайт) = 2^{10} Кбайт = 1024 Кбайта;

1 Гбайт (гигабайт) = 2^{10} Мбайт = 1024 Мбайта.

В последнее время в связи с увеличением объемов обрабатываемой информации входят в употребление такие производные единицы, как:

1 Терабайт (Тб) = 10^{24} Гбайта = 2^{40} байта,

1 Петабайт (Пб) = 10^{24} Тбайта = 2^{50} байта.

Глава 2. Создание графического интерфейса средствами языка программирования Python.

Для создания программы мы познакомились с особенностями создания графического интерфейса средствами пакета Tkinter языка программирования Python. Tkinter – это пакет для Python, предназначенный для работы с библиотекой Tk. Библиотека Tk содержит компоненты графического интерфейса пользователя (graphical user interface – GUI).

Под графическим интерфейсом пользователя (GUI) подразумеваются окна, кнопки, текстовые поля для ввода, скроллеры, списки, радиокнопки, флажки и другие элементы которые мы видим на экране, открывая то или иное приложение. Через них происходит взаимодействие с программой и управление ею. Все эти элементы интерфейса будем называть виджетами (widgets). В настоящее время почти все приложения, которые создаются для пользователя, имеют графический интерфейс, который более интуитивен и удобен для пользователя, чем консоль GUI. До этого на уроках информатики мы писали только консольные программы.

Tkinter импортируется стандартно для модуля Python:

```
from tkinter import *
```

Чтобы написать GUI-программу, надо выполнить приблизительно следующее:

1. Создать главное окно
2. Сформировать модель окна и разместить в нем виджеты.
3. Определить события, то есть то, на что будет реагировать программа.
4. Описать обработчики событий.
5. Протестировать работу программы.

Глава 2. Создание графического интерфейса средствами языка программирования Python.

Базовым моментом в построении графических программ является создание окна. В программе с графическим интерфейсом может быть несколько окон, которые обычно называют формами. Основные методы работы с окном:

<code>root = Tk()</code>	Для создания графического окна применяется конструктор <code>Tk()</code> . Переменную, связываемую с объектом, часто называют <code>root</code> (корень), но можно этой переменной дать другое название.
<code>root.title('Пример 1')</code>	Устанавливается заголовок окна.
<code>root.geometry('600x400')</code>	Устанавливается размер окна (передается строка в формате "Ширина x Высота").
<code>root.geometry('400x300+100+100')</code>	Устанавливается размер окна и его положение относительно левого верхнего угла экрана (передается строка в формате "Ширина x Высота + координатаX + координатаY").
<code>root.mainloop()</code>	Запускает цикл обработки событий окна для взаимодействия с пользователем. Данная строчка кода должна быть всегда в конце скрипта!

На форме размещаются элементы графического интерфейса (виджеты): поля ввода, надписи, кнопки, флажки, переключатели, выпадающие списки и др.

Виджеты – это базовые блоки для создания графического интерфейса программы. За годы развития программирования некоторые из виджетов стали стандартными во всех языках и на всех платформах. Каждый виджет – это объект определённого класса, у которого есть свойства и методы.

В библиотеке Tkinter каждый виджет имеет определенные свойства, значения которых можно задавать при их создании, а также программировать их изменение при действии пользователя и в результате выполнения программы. Рассмотрим некоторые виджеты и их свойства.

Кнопка (Button)

Для создания кнопки используется класс `Button()`.

`Name = Button (window, options)` – объявление кнопки в программе, где

Name	имя виджета (идентификатор устанавливает пользователь). Под этим именем данный объект в дальнейшем используется в программе.
window	ссылка на родительское окно (контейнер).
options	представляет собой набор параметров для данного класса

Глава 2. Создание графического интерфейса средствами языка программирования Python.

Параметры класса **Button()**:

text	текст на кнопке, можно сделать многострочный текст (\n)
width	ширина кнопки (единицами измерения являются знакоместа)
height	высота кнопки (единицами измерения являются знакоместа)
font	вид шрифта на кнопке, свойство должно иметь вид: "Имя_шрифта размер".
bg	цвет фона кнопки
fg	цвет текста
relief	рельеф кнопки (flat, groove, ridge, sunken, raised)
bd	ширина границы кнопки
image	имя картинки, отображаемой на кнопке
padx	отступ от границ кнопки до её текста справа и слева;
pady	отступ от границ кнопки до её текста сверху и снизу
activebackground	цвет фона (когда кнопка нажата)
activeforeground	цвет текста (когда кнопка нажата)
command	установка действия, то есть того, что будет происходить при нажатии на кнопку
state	normal – обычное состояние кнопки, при котором она может нажиматься и взаимодействовать с пользователем. disabled – такое состояние кнопки, при котором она не может взаимодействовать с пользователем.
compound	Расположение картинки на кнопке (center, bottom, left, right, top). Изначально картинка на кнопке будет отображаться вместо текста, но это можно изменить, изменяя значения свойства compound: bottom – картинка будет отображаться под текстом; left – картинка будет отображаться слева от текста; right – картинка будет отображаться справа от текста; top – картинка будет отображаться над текстом.

Поле ввода (Entry)

Элемент Entry представляет поле для ввода и редактирования однострочного текста.

Параметры текстового поля:

width	ширина элемента
bg	фоновый цвет
bd	толщина границы
fg	цвет текста

Глава 2. Создание графического интерфейса средствами языка программирования Python.

font	шрифт текста
justify	устанавливает выравнивание текста (left, center ,right)
selectbackground	фоновый цвет выделенного куска текста
selectforeground	цвет выделенного текста
state	состояние элемента, может принимать значения normal (по умолчанию) и disabled
relief	тип границы (flat, groove, ridge, sunken, raised)

Методы текстового поля

Из текстового поля можно взять текст. За это действие отвечает метод **get**. В текстовое поле можно вставить текст методом **insert**. Также можно удалить текст методом **delete**.

Текстовая метка (Label)

Текстовые метки в Python представлены элементом **Label**. Этот элемент позволяет выводить статический текст без возможности редактирования.

Параметры текстовой метки:

height	высота элемента
width	ширина элемента
bg	фоновый цвет
fg	цвет текста
font	шрифт текста
bd	толщина границы метки
anchor	устанавливает позиционирование текста (N, NE, E, SE, S, SW, W, NW)
bitmap	ссылка на изображение, которое отображается на метке
wrlength	при положительном значении строки текста будут переноситься для вмещения в пространство элемента

Радиокнопка (Radiobutton)

В Tkinter от класса Radiobutton создаются радиокнопки. Радиокнопки не создают по одной, а делают связанную группу, работающую по принципу переключателей. Когда включена одна, другие выключены.

Связь между переключателями устанавливается через общую переменную, разные значения которой соответствуют включению разных радиокнопок группы. У всех кнопок одной группы свойство `variable` устанавливается в одно и то же значение – связанную с группой переменную. А свойству `value` присваиваются разные значения этой переменной.

Глава 2. Создание графического интерфейса средствами языка программирования Python.

В Tkinter нельзя использовать любую переменную для хранения состояний виджетов. Для этих целей предусмотрены специальные классы-переменные пакета tkinter.

- BooleanVar - позволяет принимать своим экземплярам только булевы значения (0 или 1 и True или False)
- IntVar - позволяет принимать своим экземплярам только целые значения
- DoubleVar - позволяет принимать своим экземплярам только дробные значения
- StringVar - позволяет принимать своим экземплярам только строковые значения.

Мы в программе будем использовать класс IntVar.

Управление цветом

Цвет в программе используется в формате RGB. Для определения цвета применялся онлайн калькулятор подбора цвета с помощью палитры цветовой модели RGB. (<https://findhow.org/5060-konverter-tsvetov.html>)

Размещение виджетов в окне

В Tkinter существует три менеджера геометрии: упаковщик (pack), сетка (grid) и размещение по координатам (place).

Если к элементу интерфейса не применить какой-либо из менеджеров геометрии, то он не отобразится в окне. При этом в одном окне нельзя комбинировать разные менеджеры.

Табличный способ размещения (сетка) предпочтителен из-за его гибкости и удобства, когда дело доходит до разработки относительно сложных интерфейсов. При размещении виджетов методом grid окно программы условно разделяется на ячейки подобно таблице. Адрес каждой ячейки состоит из номера строки и номера столбца. Нумерация начинается с нуля. Ячейки можно объединять как по вертикали, так и по горизонтали.

Размещение виджета в той или иной ячейке задается через аргументы row и column, которым присваиваются соответственно номера строки и столбца. Чтобы объединить ячейки по горизонтали, используется атрибут colspan, которому присваивается количество объединяемых ячеек. Опция rowspan объединяет ячейки по вертикали.

Атрибуты padx, pady, ipadx, ipady используются для задания внешних и внутренних отступов. Кроме этого есть атрибут sticky (липкий), который принимает значения направлений сторон света (N, S, W, E, NW, NE, SW, SE). Если, например, указать NW, то виджет прибудет к верхнему левому углу ячейки. Виджеты можно растягивать на весь объем ячейки (sticky=N+S+W+E) или только по одной из осей (N+S или W+E). Эффект от "липучки" заметен, только если виджет меньше ячейки.

Глава 3. Разработка программы

1. Создаем главное окно программы.

Для создания графического окна применяется конструктор Tk(). Назовем окно переменной win. Укажем в строке заголовка окна «Конвертер единиц измерения информации». Размер окна 440 на 280.

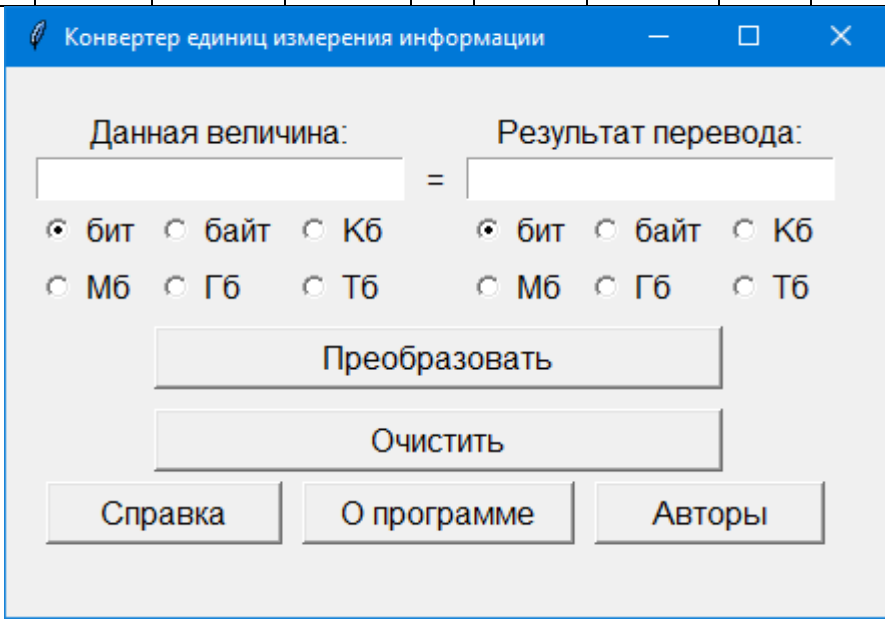
```
win = Tk()
win.title('Конвертер единиц измерения информации')
win.geometry('440x280')
```

2. Формируем модель окна и размещаем в нем виджеты.

В окне программы разместим: 3 текстовых метки, 2 поля ввода и 5 кнопок.

Для размещения виджетов используем метод grid(). Представим все виджеты в виде таблицы. Будем размещать виджеты, согласно их расположения в таблице (строка и столбец).

Таблица 1.

	0	1	2	3	4	5	6	7	8
Заголовок									
0									
1									
2									
3									
4									
5									
6									
7									

Код размещения виджетов в окне win:

```
Label(win, text='Данная величина:', font=('Arial', 12)).grid(
    row=1, column=1, columnspan=3, sticky=W + E)
Label(win, text='Результат перевода:', font=('Arial', 12)).grid(
    row=1, column=5, columnspan=3, sticky=W + E)
Label(win, text=' = ', font=('Arial', 12)).grid( row=2, column=4, sticky=W + E)
Label(win, text=' ').grid(row=0, column=0, sticky=W + E)
```

```
Label(win, text=' ').grid(row=0, column=8, sticky=W + E)
```

```
EntryA = Entry(win, font=('Arial', 12))
```

```
EntryR = Entry(win, font=('Arial', 12))
```

```
EntryA.grid(row=2, column=1, columnspan=3, sticky=W + E)
```

```
EntryR.grid(row=2, column=5, columnspan=3, sticky=W + E)
```

```
EntryA.focus_set()
```

```
var1 = IntVar()
```

```
var1.set(0)
```

```
var2 = IntVar()
```

```
var2.set(0)
```

```
bit1 = Radiobutton(text="бит", font=('Arial', 12), variable=var1, value=0).grid(  
    row=3, column=1, sticky=W)
```

```
b1 = Radiobutton(text="байт", font=('Arial', 12), variable=var1, value=1).grid(  
    row=3, column=2, sticky=W)
```

```
kb1 = Radiobutton(text="Кб", font=('Arial', 12), variable=var1, value=2).grid(  
    row=3, column=3, sticky=W)
```

```
mb1 = Radiobutton(text="Мб", font=('Arial', 12), variable=var1, value=3).grid(  
    row=4, column=1, sticky=W)
```

```
gb1 = Radiobutton(text="Гб", font=('Arial', 12), variable=var1, value=4).grid(  
    row=4, column=2, sticky=W)
```

```
tb1 = Radiobutton(text="Тб", font=('Arial', 12), variable=var1, value=5).grid(  
    row=4, column=3, sticky=W)
```

```
bit2 = Radiobutton(text="бит", font=('Arial', 12), variable=var2, value=0).grid(  
    row=3, column=5, sticky=W)
```

```
b2 = Radiobutton(text="байт", font=('Arial', 12), variable=var2, value=1).grid(  
    row=3, column=6, sticky=W)
```

```
kb2 = Radiobutton(text="Кб", font=('Arial', 12), variable=var2, value=2).grid(  
    row=3, column=7, sticky=W)
```

```
mb2 = Radiobutton(text="Мб", font=('Arial', 12), variable=var2, value=3).grid(  
    row=4, column=5, sticky=W)
```

```
gb2 = Radiobutton(text="Гб", font=('Arial', 12), variable=var2, value=4).grid(  
    row=4, column=6, sticky=W)
```

```
tb2 = Radiobutton(text="Тб", font=('Arial', 12), variable=var2, value=5).grid(
    row=4, column=7, sticky=W)

# размещаем кнопку Преобразовать
but_p = Button(win, text='Преобразовать', font=('Arial', 12), command=converting).grid(
    row=5, column=2, columnspan=5, pady=5, sticky=W + E)

# размещаем кнопку Очистить
but_del = Button(win, text='Очистить', font=('Arial', 12), command=clear).grid(
    row=6, column=2, columnspan=5, pady=5, sticky=W + E)

# размещаем кнопку Справка, открывающую окно Единицы измерения информации
but_del = Button(win, text='Справка', font=('Arial', 12), command=help).grid(
    row=7, column=1, columnspan=2, padx=5, sticky=W + E)

# размещаем кнопку О программе
but_del = Button(win, text='О программе', font=('Arial', 12), command=instr).grid(
    row=7, column=3, columnspan=3, padx=5, sticky=W + E)

# размещаем кнопку Авторы
but_del = Button(win, text='Авторы', font=('Arial', 12), command=avtor).grid(
    row=7, column=6, columnspan=2, padx=5, sticky=W + E)
```

3. Определяем события.

В программе имеется пять кнопок. При нажатии на эти кнопки происходят определенные события.

Кнопка «Преобразовать». При корректной работе программы в текстовое поле «Данная величина» должно быть внесено числовое значение (целое или действительное число), выбрана единица изменения данного числа и единица измерения результата. При нажатии на кнопку «Преобразовать» (Таблица 1.) в текстовом поле «Результат перевода» отображается значение в выбранной для результата единице измерения.

Кнопка «Очистить». При нажатии на кнопку «Очистить» (Таблица 1.) в текстовых полях «Данная величина» и «Результат перевода» все удаляется.

Кнопка «Справка» При нажатии на кнопку «Справка» открывается окно «Единицы измерения информации» (Рис.1)

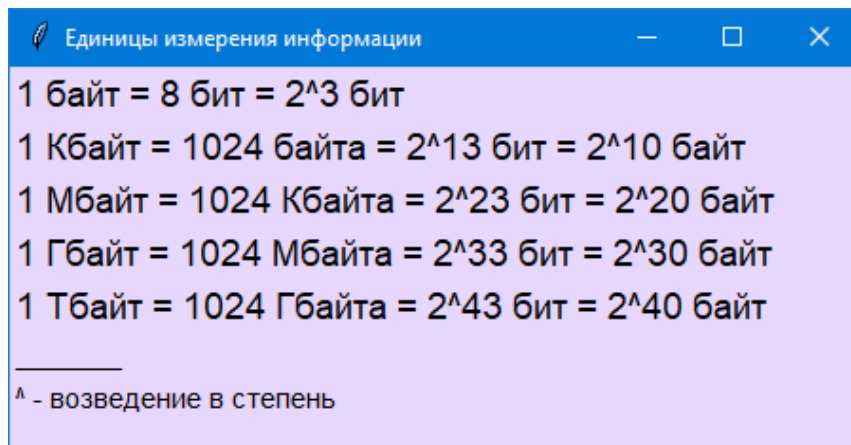


Рис.1

Кнопка «О программе» При нажатии на кнопку «О программе» открывается окно «О программе» (Рис.2) с краткой инструкцией по использованию программы.

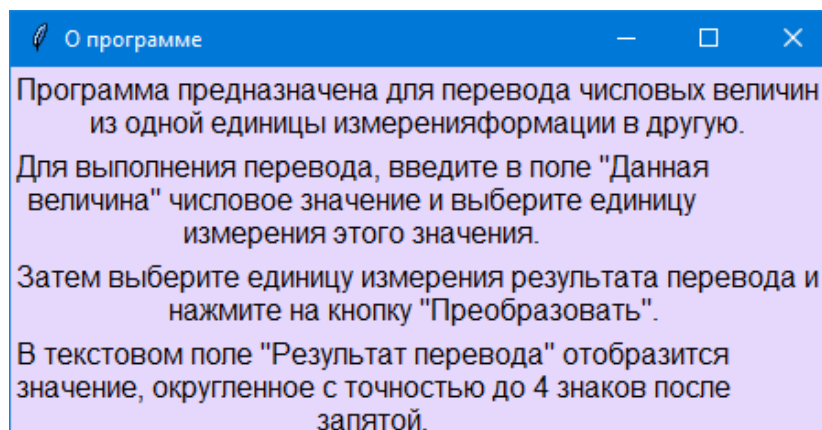


Рис.2

Кнопка «Авторы» При нажатии на кнопку «Авторы» открывается окно «Авторы» (Рис.3) с информацией о разработчиках программы.

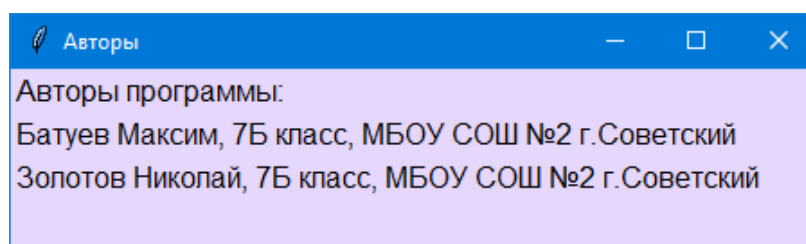


Рис.3

4. Обработчики событий. Для каждой кнопки пишем функцию.

При нажатии на кнопку «Преобразовать» выполняется функция `converting`. В функции из текстового поля «Данная величина» `EntryA` считывается текст с помощью метода `get()` и преобразуется в число. Далее считываются с помощью метода `get()` активные положения переключателей групп `var1` и `var2`. С помощью условного оператора проверяется условие, соответствующее активному положению переключателей и вычисляется результат. Результат с помощью функции `round` округляется до 4 знаков после запятой и преобразуется

в строку. Поле результат EntryR очищается командой EntryR.delete(0, END) и в него помещается вычисленное значение с помощью команды EntryR.insert(0, result)

Полный код функции:

```
def converting():
    # берем текст из поля A и преобразуем его в число
    a = float(EntryA.get())
    # если выполняется условие, вычисляем результат
    if var1.get() == var2.get():
        result = str(a)
    elif var1.get() == 0 and var2.get() == 1: # из бит в байты
        result = str(round(a / 8,4))
    elif var1.get() == 0 and var2.get() == 2: # из бит в Кбайты
        result = str(round(a / 2**13, 4))
    elif var1.get() == 0 and var2.get() == 3: # из бит в Мбайты
        result = str(round(a / 2**23,4))
    elif var1.get() == 0 and var2.get() == 4: # из бит в Гбайты
        result = str(round(a / 2**33, 4))
    elif var1.get() == 0 and var2.get() == 5: # из бит в Тбайты
        result = str(round(a / 2**43,4))

    elif var1.get() == 1 and var2.get() == 0: # из байт в биты
        result = str(a * 8)
    elif var1.get() == 1 and var2.get() == 2: # из байт в Кбайты
        result = str(round(a / 2**10, 4))
    elif var1.get() == 1 and var2.get() == 3: # из байт в Мбайты
        result = str(round(a / 2**20, 4))
    elif var1.get() == 1 and var2.get() == 4: # из байт в Гбайты
        result = str(round(a / 2**30, 4))
    elif var1.get() == 1 and var2.get() == 5: # из байт в Тбайты
        result = str(round(a / 2**40, 4))

    elif var1.get() == 2 and var2.get() == 0: # из Кбайт в биты
        result = str(a * 2**13)
    elif var1.get() == 2 and var2.get() == 1: # из Кбайт в байты
        result = str(a * 2**10)
```

```
elif var1.get() == 2 and var2.get() == 3: # из Кбайт в Мбайты
    result = str(round(a / 2**10, 4))
elif var1.get() == 2 and var2.get() == 4: # из Кбайт в Гбайты
    result = str(round(a / 2**20, 4))
elif var1.get() == 2 and var2.get() == 5: # из Кбайт в Тбайты
    result = str(round(a / 2**30, 4))

elif var1.get() == 3 and var2.get() == 0: # из Мбайт в биты
    result = str(a * 2**23)
elif var1.get() == 3 and var2.get() == 1: # из Мбайт в байты
    result = str(a * 2**20)
elif var1.get() == 3 and var2.get() == 2: # из Мбайт в Кбайты
    result = str(a * 2**10)
elif var1.get() == 3 and var2.get() == 4: # из Мбайт в Гбайты
    result = str(round(a / 2**10, 4))
elif var1.get() == 3 and var2.get() == 5: # из Мбайт в Тбайты
    result = str(round(a / 2**20, 4))

elif var1.get() == 4 and var2.get() == 0: # из Гбайт в биты
    result = str(a * 2**33)
elif var1.get() == 4 and var2.get() == 1: # из Гбайт в байты
    result = str(a * 2**30)
elif var1.get() == 4 and var2.get() == 2: # из Гбайт в Кбайты
    result = str(a * 2**20)
elif var1.get() == 4 and var2.get() == 3: # из Гбайт в Мбайты
    result = str(a * 2**10)
elif var1.get() == 4 and var2.get() == 5: # из Гбайт в Тбайты
    result = str(round(a / 2**10, 4))

elif var1.get() == 5 and var2.get() == 0: # из Тбайт в биты
    result = str(a * 2**43)
elif var1.get() == 5 and var2.get() == 1: # из Тбайт в байты
    result = str(a * 2**40)
elif var1.get() == 5 and var2.get() == 2: # из Тбайт в Кбайты
    result = str(a * 2**30)
```

```
elif var1.get() == 5 and var2.get() == 3: # из Тбайт в Мбайты
    result = str(a * 2**20)
elif var1.get() == 5 and var2.get() == 4: # из Тбайт в Гбайты
    result = str(a * 2**10)

EntryR.delete(0, END) # очищаем текстовое поле полностью
EntryR.insert(0, result) # вставляем результат в начало
```

При нажатии на кнопку «Очистить» выполняется функция `clear`. Для очистки текстовых полей используем метод `delete`. Для полной очистки поля нужно первым аргументом указать `0`, вторым – `END`.

Полный код функции:

```
def clear():
    EntryA.delete(0, END)
    EntryR.delete(0, END)
```

При нажатии на кнопку «Справка» выполняется функция `help`. Открывается новое окно с заголовком «Единицы измерения информации», размера `450x200`. С помощью метода `grid()` в окне размещаются текстовые метки о единицах измерения информации.

Полный код функции:

```
def help():
    win1 = Tk()
    win1.title('Единицы измерения информации')
    win1.geometry('450x200')
    win1.config(bg='#e6d8fd')
    Label(win1, text='1 байт = 8 бит = 2^3 бит', bg='#e6d8fd',
          font=('Arial', 14)).grid(row=0, column=0, sticky=W)
    Label(win1, text='1 Кбайт = 1024 байта = 2^13 бит = 2^10 байт', bg='#e6d8fd',
          font=('Arial', 14)).grid(
        row=1, column=0, sticky=W)
    Label(win1, text='1 Мбайт = 1024 Кбайта = 2^23 бит = 2^20 байт', bg='#e6d8fd',
          font=('Arial', 14)).grid(
        row=2, column=0, sticky=W)
    Label(win1, text='1 Гбайт = 1024 Мбайта = 2^33 бит = 2^30 байт', bg='#e6d8fd',
          font=('Arial', 14)).grid(row=3, column=0, sticky=W)
```



```
Label(win1, text='1 Тбайт = 1024 Гбайта = 2^43 бит = 2^40 байт', bg='#e6d8fd',
      font=('Arial', 14)).grid(row=4, column=0, sticky=W)
Label(win1, text='_____', bg='#e6d8fd',
      font=('Arial', 11)).grid(row=5, column=0, sticky=W)
Label(win1, text='^ - возведение в степень', bg='#e6d8fd',
      font=('Arial', 11)).grid(row=6, column=0, sticky=W)
win1.mainloop()
```

При нажатии на кнопку «О программе» выполняется функция `instr`. Открывается новое окно с заголовком «О программе», размера 450x200. С помощью метода `grid()` в окне размещаются текстовые метки с инструкцией об использовании программы.

Полный код функции:

```
def instr():
    win2 = Tk()
    win2.title('О программе')
    win2.geometry('450x200')
    win2.config(bg='#e6d8fd')
    Label(win2, text='Программа предназначена для перевода числовых величин из
одной единицы измерения информации в другую.', bg='#e6d8fd', font=('Arial', 12),
wraplength=450).grid(row=0, column=0, sticky=W)
    Label(win2, text='Для выполнения перевода, введите в поле "Данная величина"
числовое значение и выберите единицу измерения этого значения.', bg='#e6d8fd', font=('Arial',
12), wraplength=450).grid(row=1, column=0, sticky=W)
    Label(win2, text='Затем выберите единицу измерения результата перевода и
нажмите на кнопку "Преобразовать".', bg='#e6d8fd', font=('Arial', 12),
wraplength=450).grid(row=2, column=0, sticky=W)
    Label(win2, text='В текстовом поле "Результат перевода" отобразится значение,
округленное с точностью до 4 знаков после запятой.', bg='#e6d8fd', font=('Arial', 12),
wraplength=450).grid(row=3, column=0, sticky=W)
    win2.mainloop()
```

При нажатии на кнопку «Авторы» выполняется функция `avtor`. Открывается новое окно с заголовком «Авторы», размера 450x100. С помощью метода `grid()` в окне размещаются текстовые метки с информацией об авторах программы.

Полный код функции:

```
def avtor():
    win3 = Tk()
```

```
win3.title('Авторы')
win3.geometry('450x100')
win3.config(bg='#e6d8fd')
Label(win3, text='Авторы программы:', bg='#e6d8fd',
       font=('Arial', 12)).grid(row=0, column=0, sticky=W)
Label(win3, text='Батуев Максим, 7Б класс, МБОУ СОШ №2 г.Советский',
       bg='#e6d8fd', font=('Arial', 12)).grid(row=1, column=0, sticky=W)
Label(win3, text='Золотов Николай, 7Б класс, МБОУ СОШ №2 г.Советский',
       bg='#e6d8fd', font=('Arial', 12)).grid(row=2, column=0, sticky=W)
win3.mainloop()
```

5. Тестирование программы.

Созданная программа была протестирована нами и обучающимися 7 класса. Программа использовалась на уроках информатики, при решении задач на измерение количества информации. С помощью программы можно легко переводить данные из одной единицы измерения информации в другую или осуществлять самоконтроль при выполнении такого перевода. В ходе тестирования было замечено, что при запуске программы хорошо было бы установить курсор в текстовое поле «Данная величина» по умолчанию. Для этого в программу была добавлена команда `EntryA.focus_set()`, устанавливающая фокус на текстовое поле.

Глава 4. Результат проекта

В ходе выполнения проекта мы познакомились с технологией создания программы с графическим интерфейсом пользователя на языке Python. Мы использовали основные виджеты окна и, опираясь на уже известные нам алгоритмические структуры (условный оператор), создали программу с графическим интерфейсом.

Таким образом, цель работы «Создание конвертера единиц измерения информации средствами языка программирования Python», мы считаем достигнутой. Программу можно использовать на компьютерах с минимальными техническими требованиями. Сконвертированная в исполняемый файл программа работает на 64 битной системе с установленным языком программирования Python.

Перевод единиц измерения информации требует от учащихся 7 классов хороших математических знаний и навыков решения задач. Программа позволит отработать навык перевода одной единицы измерения в другую. Также программу можно использовать в качестве примера демонстрации возможностей создания графического интерфейса пользователя средствами пакета Tkinter языка программирования Python.

Список литературы.

1. Доусон М. Програмируем на Python. – СПб.: Питер, 2014. – 416 с.
2. МакГрат, М. Программирование на Python для начинающих / М. МакГрат. - М.: Эксмо, 2015. - 192 с.
3. С. Шапошникова. Основы программирования на Python. Учебник. Вводный курс. – версия 2. – 2011. – 44 с.
4. Сузи Р. А. Язык программирования Python: Учебное пособие. – М.: ИНТУИТ, БИНОМ. Лаборатория знаний, 2006. – 328 с.
5. Босова Л.Л. Информатика. 7класс: учебник. – ООО «БИНОМ. Лаборатория знаний», 2019 г. – 240 с.
6. Tkinter. Программирование GUI на Python. [Электронный ресурс] - URL: <https://younglinux.info/tkinter/course> (Дата обращения: 15.02.2023)
7. Самоучитель Python. [Электронный ресурс] - URL: <https://pythonworld.ru/samouchitel-python> (Дата обращения: 15.02.2023)