

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по домашнему заданию на тему
“Разработка продвинутого калькулятора”

Выполнил:

Студент группы ИУ5-35Б

Евдокимов М. С.

Преподаватель:

Нардид А. Н.

Москва 2025

Полученное изначально задание

1. Выберите язык программирования (который Вы ранее не изучали) и (1) напишите по нему реферат с примерами кода или (2) реализуйте на нем небольшой проект (с детальным текстовым описанием).
2. Реферат (проект) может быть посвященциальному аспекту (аспектам) языка или содержать решение какой-либо задачи на этом языке.
3. Необходимо установить на свой компьютер компилятор (интерпретатор, транспилятор) этого языка и произвольную среду разработки.
4. В случае написания реферата необходимо разработать и откомпилировать примеры кода (или модифицировать стандартные примеры).
5. В случае создания проекта необходимо детально комментировать код.
6. При написании реферата (создании проекта) необходимо изучить и корректно использовать особенности парадигмы языка и основных конструкций данного языка.
7. Приветствуется написание черновика статьи по результатам выполнения ДЗ. Черновик статьи может быть подготовлен группой студентов, которые исследовали один и тот же аспект в нескольких языках или решили одинаковую задачу на нескольких языках.

Листинг программы

```
import tkinter as tk
from tkinter import messagebox

def add_digit(digit):
    value = calc.get()
    if value[0] == '0' and len(value) == 1:
        value = value[:-1]
    calc['state'] = tk.NORMAL
    calc.delete(0, tk.END)
    calc.insert(0, value + digit)
    calc['state'] = tk.DISABLED

def add_operation(operation):
    value = calc.get()

    if value == '0' and operation == '-':
        calc['state'] = tk.NORMAL
        calc.delete(0, tk.END)
        calc.insert(0, '-')
        calc['state'] = tk.DISABLED
        return

    if value[-1] in '+-*':
        value = value[:-1]

    calc['state'] = tk.NORMAL
    calc.delete(0, tk.END)
    calc.insert(0, value + operation)
    calc['state'] = tk.DISABLED

def calculate():
    calc['state'] = tk.NORMAL
    value = calc.get().strip()

    if value and value[-1] in '+-*':
        value = value[:-1]

    try:
        result = eval(value)

        if isinstance(result, float) and result.is_integer():
            result = int(result)

        calc.delete(0, tk.END)
        calc.insert(0, str(result))

    except ZeroDivisionError:
        messagebox.showinfo('Внимание', 'На ноль делить нельзя!!! пора бы это уже выучить...')

        calc.delete(0, tk.END)
        calc.insert(0, 0)

    except (NameError, SyntaxError):
        messagebox.showinfo('Внимание', 'нужно только цифры!!! Вы ввели другие символы!!!')

        calc.delete(0, tk.END)
        calc.insert(0, 0)
```

```
except Exception as e:
    messagebox.showinfo('Ошибка', f'Что-то пошло не так: {e}')
    calc.delete(0, tk.END)
    calc.insert(0, 0)

calc['state'] = tk.DISABLED

def clear():
    calc['state'] = tk.NORMAL
    calc.delete(0, tk.END)
    calc.insert(0, 0)
    calc['state'] = tk.DISABLED

def make_digit_button(digit):
    return tk.Button(text=digit, bd=5, font=('Ariel', 15), command=lambda: add_digit(digit))

def make_operation_button(operation):
    return tk.Button(text=operation, bd=5, font=('Ariel', 15), fg='red', command=lambda: add_operation(operation))

def make_calc_button(operation):
    return tk.Button(text=operation, bd=5, font=('Ariel', 15), fg='red', command=calculate)

def make_clear_button(operation):
    return tk.Button(text=operation, bd=5, font=('Ariel', 15), fg='red', command=clear)

def press_key(event):
    ch = event.char or ""
    ks = event.keysym or ""

    if ks in ('Return', 'KP_Enter'):
        calculate()
        return "break"

    if ks == 'Escape':
        clear()
        return "break"

    if ks in ('Delete', 'BackSpace'):
        value = calc.get()
        calc['state'] = tk.NORMAL
        if len(value) <= 1:
            calc.delete(0, tk.END)
            calc.insert(0, '0')
        else:
            calc.delete(0, tk.END)
            calc.insert(0, value[-1])
        calc['state'] = tk.DISABLED
        return "break"

    if ch.isdigit():
        add_digit(ch)
        return "break"

    if ch in '+-*/'：
        add_operation(ch)
        return "break"
```

```

if ks in ('KP_Add', 'KP_Subtract', 'KP_Multiply', 'KP_Divide'):
    mapping = {
        'KP_Add': '+',
        'KP_Subtract': '-',
        'KP_Multiply': '*',
        'KP_Divide': '/',
    }
    add_operation(mapping[ks])
    return "break"

if ch == '=':
    calculate()
    return "break"

return

win = tk.Tk()
win.geometry = (f'240*270+100+200')
win['bg'] = '#33ffe1'
win.title('Калькулятор')

win.bind_all('<Key>', press_key)

calc = tk.Entry(win, justify=tk.RIGHT, font=('Ariel', 15))
calc.insert(0, '0')
calc['state'] = tk.DISABLED
calc.grid(row=0, column=0, columnspan=4, stick='we', padx=5)

make_digit_button('1').grid(row=1, column=0, stick='wens', padx=5, pady=5)
make_digit_button('2').grid(row=1, column=1, stick='wens', padx=5, pady=5)
make_digit_button('3').grid(row=1, column=2, stick='wens', padx=5, pady=5)
make_digit_button('4').grid(row=2, column=0, stick='wens', padx=5, pady=5)
make_digit_button('5').grid(row=2, column=1, stick='wens', padx=5, pady=5)
make_digit_button('6').grid(row=2, column=2, stick='wens', padx=5, pady=5)
make_digit_button('7').grid(row=3, column=0, stick='wens', padx=5, pady=5)
make_digit_button('8').grid(row=3, column=1, stick='wens', padx=5, pady=5)
make_digit_button('9').grid(row=3, column=2, stick='wens', padx=5, pady=5)
make_digit_button('0').grid(row=4, column=0, stick='wens', padx=5, pady=5)

make_operation_button('+').grid(row=1, column=3, stick='wens', padx=5, pady=5)
make_operation_button('-').grid(row=2, column=3, stick='wens', padx=5, pady=5)
make_operation_button('/').grid(row=3, column=3, stick='wens', padx=5, pady=5)
make_operation_button('*').grid(row=4, column=3, stick='wens', padx=5, pady=5)

make_calc_button('=').grid(row=4, column=2, stick='wens', padx=5, pady=5)

make_clear_button('C').grid(row=4, column=1, stick='wens', padx=5, pady=5)

win.grid_columnconfigure(0, minsize=60)
win.grid_columnconfigure(1, minsize=60)
win.grid_columnconfigure(2, minsize=60)
win.grid_columnconfigure(3, minsize=60)

win.grid_rowconfigure(1, minsize=60)
win.grid_rowconfigure(2, minsize=60)
win.grid_rowconfigure(3, minsize=60)
win.grid_rowconfigure(4, minsize=60)

win.mainloop()

```

Скриншоты работы программы

