

Тема: Составление программ с использованием GUI Tkinter в IDE PyCharm Community, изучение возможностей модуля OS.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community, изучить возможности модуля OS.

Постановка задачи №1:

Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ №№ 2 – 9.

Текст программы:

```
import tkinter as tk
from PZ_9.pz_9 import init_dict, remove_element_from_dict

class DictionaryManager:
    def __init__(self, master):
        self.master = master
        master.title("Dictionary Manager")
        self.dict = {}

        self.label_size = tk.Label(master, text="Enter dictionary size:")
        self.label_size.grid(row=0, column=0)

        self.entry_size = tk.Entry(master)
        self.entry_size.grid(row=0, column=1)

        self.create_button = tk.Button(master, text="Create Dictionary",
command=self.create_dictionary)
        self.create_button.grid(row=0, column=2)

        self.label_remove = tk.Label(master, text="Enter elements to remove (space
separated):")
        self.label_remove.grid(row=1, column=0)

        self.entry_remove = tk.Entry(master)
        self.entry_remove.grid(row=1, column=1)

        self.remove_button = tk.Button(master, text="Remove Elements",
command=self.remove_elements)
        self.remove_button.grid(row=1, column=2)

        self.output_label = tk.Label(master, text="Output:")
        self.output_label.grid(row=2, column=0)

        self.output_text = tk.Text(master, width=70, height=20)
        self.output_text.grid(row=3, column=0, columnspan=3)

    def create_dictionary(self):
        size = int(self.entry_size.get())
        self.output_text.insert(tk.END, f"Creating dictionary of size {size}...\n")
```

```

self.dict = {}

init_dict(self.dict, size)

self.output_text.insert(tk.END, f"Created dictionary: {self.dict}\n")

def remove_elements(self):
    elements = list(map(int, self.entry_remove.get().split()))
    self.output_text.insert(tk.END, f"Removing elements: {elements}\n")

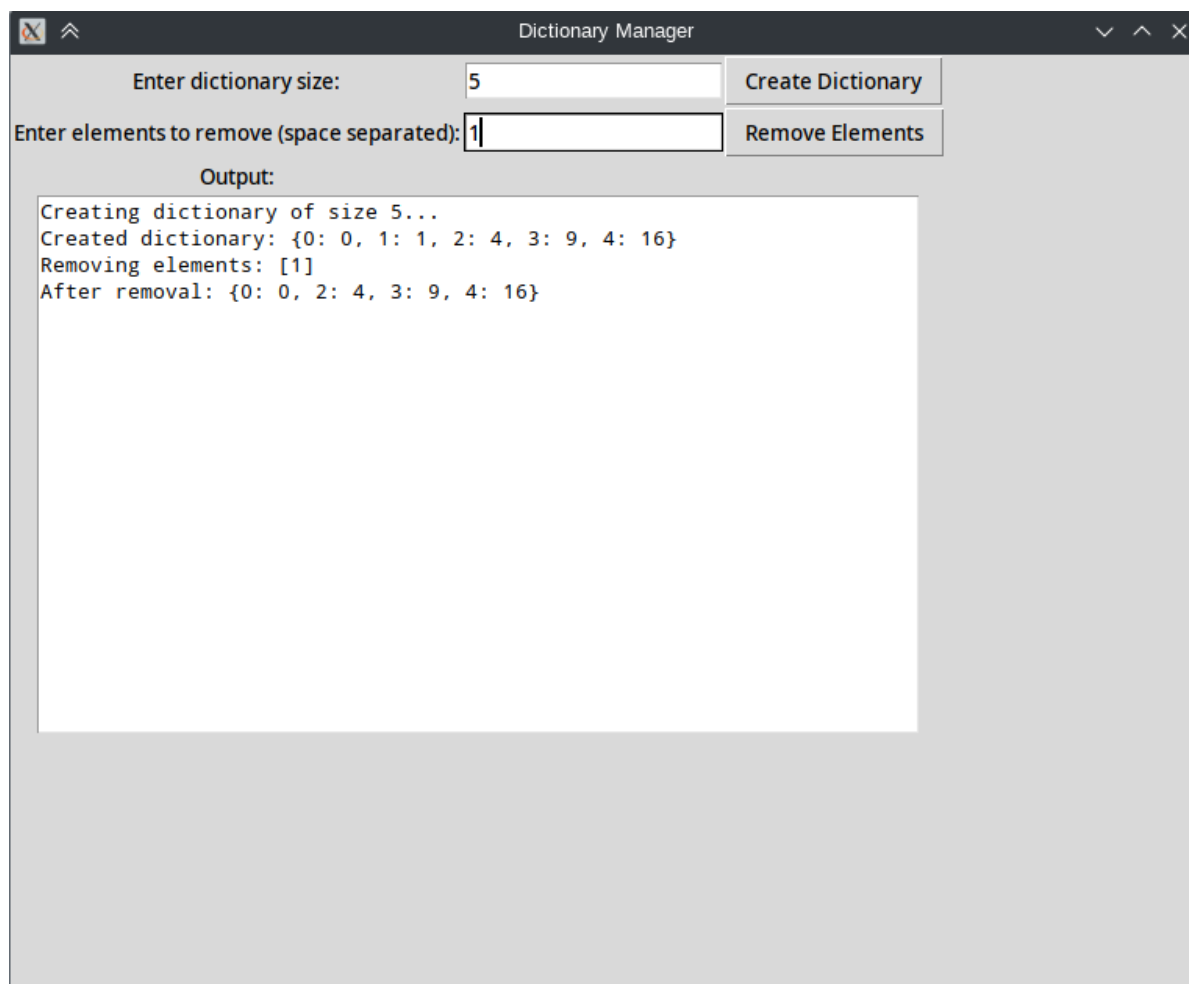
    remove_element_from_dict(self.dict, elements)

    self.output_text.insert(tk.END, f"After removal: {self.dict}\n")

root = tk.Tk()
app = DictionaryManager(root)
root.mainloop()

```

Протокол работы программы:



Постановка задачи №2:

Задание предполагает, что у студента есть проект с практическими работами (№№ 2-13), оформленный согласно требованиям. Все задания выполняются с использованием

модуля OS:

⌘ перейдите в каталог PZ11. Выведите список всех файлов в этом каталоге. Имена вложенных подкаталогов выводить не нужно.

⌘ перейти в корень проекта, создать папку с именем test. В ней создать еще одну папку test1. В папку test переместить два файла из ПЗ6, а в папку test1 - один файл из ПЗ7. Файл из ПЗ7 переименовать в test.txt. Вывести в консоль информацию о размере файлов в папке test.

⌘ перейти в папку с PZ11, найти там файл с самым коротким именем, имя вывести в консоль. Использовать функцию basename () (os.path.basename()). ⌘ перейти в любую папку где есть отчет в формате .pdf и «запустите» файл в привязанной к нему программе. Использовать функцию os.startfile().

⌘ удалить файл test.txt.

Текст программы:

```
import os

root = os.path.abspath('./')

def get_paths():
    return {
        "PZ6": os.path.join(root, 'PZ_6'),
        "PZ7": os.path.join(root, 'PZ_7', 'pz_7.py'),
        "PZ11": os.path.join(root, 'PZ_11'),
        "test": os.path.join(root, 'test'),
        "test1": os.path.join(root, 'test', 'test1'),
        "test_file": os.path.join(root, 'test', 'test1', 'test.txt'),
        "reports": os.path.join(root, 'reports'),
        "report_pdf": 'PZ_7_report.pdf'
    }

def change_directory(path):
    if not os.path.exists(path):
        print(f"Каталог {path} не найден")
        return False
    os.chdir(path)
    return True

def copy_file(source, destination):
    if not os.path.exists(source):
        print(f"Файл {source} не найден")
        return
    with open(source, 'rb') as f_src, open(destination, 'wb') as f_dst:
        f_dst.write(f_src.read())

def list_files_in_directory(path):
    if not os.path.exists(path):
        print(f"Каталог {path} не найден")
        return []
    return [f for f in os.listdir(path) if os.path.isfile(os.path.join(path, f))]

def create_directory(path):
```

```

os.makedirs(path, exist_ok=True)

def print_file_sizes(directory):
    if not os.path.exists(directory):
        print(f"Каталог {directory} не найден")
        return
    for file_s in os.listdir(directory):
        if os.path.isfile(os.path.join(directory, file_s)):
            print(f"Размер файла {file_s}: {os.path.getsize(os.path.join(directory, file_s))} байт")

os.chdir(root)

paths = get_paths()
files_in_pz11 = list_files_in_directory(paths['PZ11'])

print("Файлы в каталоге PZ_11:", files_in_pz11)

create_directory(paths['test1'])

file_to_copy = 'pz_6.py'

src = os.path.join(paths['PZ6'], file_to_copy)
dst = os.path.join(paths['test'], file_to_copy)
copy_file(src, dst)

copy_file(paths['PZ7'], paths['test_file'])

print_file_sizes(paths['test'])

if files_in_pz11:
    shortest_filename = min(files_in_pz11, key=len)
    print("Файл с самым коротким именем:", os.path.basename(shortest_filename))

if change_directory(paths['reports']) and os.path.exists(paths['report_pdf']):
    pdf_path = paths["report_pdf"]
    print(pdf_path)
    os.system(f'xdg-open "{pdf_path}"')
else:
    print(f"PDF файл {paths['report_pdf']} не найден")

if os.path.exists(paths['test_file']):
    os.remove(paths['test_file'])
    print(f"Файл test.txt успешно удален")
else:
    print(f"Файл test.txt не найден")

```

Протокол работы программы:

Файлы в каталоге PZ_11: ['pz_11(1, 2).py', '__init__.py']

Размер файла pz_6.py: 3048 байт

Файл с самым коротким именем: __init__.py

PZ_7_report.pdf

Файл test.txt успешно удален

Process finished with exit code 0

Вывод: В процессе выполнения практического занятия закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием GUI Tkinter в IDE PyCharm Community, изучить возможности модуля OS.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на **GitHub**.