

Тема: Наименование практического занятия: 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

Задание 1

Постановка задачи: В соответствии с номером варианта перейти по ссылке на прототип. Реализовать его в IDE PyCharm Community с применением пакета tk. Получить интерфейс максимально приближенный к оригиналу

Код программы:

```
import tkinter as tk
from tkinter import ttk

def create_order_form():
    # Create main window
    root = tk.Tk()
    root.title("Создайте заказ")

    # Create main frame
    main_frame = ttk.Frame(root, padding="10 10 10 10")
    main_frame.grid(row=0, column=0, sticky=(tk.W, tk.E, tk.N,
tk.S))

    # Order information section
    order_info_frame = ttk.LabelFrame(main_frame, text="Информация
о заказе", padding="10 10 10 10")
    order_info_frame.grid(row=0, column=0, sticky=(tk.W, tk.E,
tk.N, tk.S))

    ttk.Label(order_info_frame, text="Номер заказа *").grid(row=0,
column=0, sticky=tk.W)
    order_number_entry = ttk.Entry(order_info_frame)
    order_number_entry.grid(row=0, column=1, sticky=(tk.W, tk.E))

    ttk.Label(order_info_frame, text="Название товара").grid(row=1,
column=0, sticky=tk.W)
    product_name_entry = ttk.Entry(order_info_frame)
    product_name_entry.grid(row=1, column=1, sticky=(tk.W, tk.E))

    ttk.Label(order_info_frame, text="Количество *").grid(row=2,
column=0, sticky=tk.W)
```

```

quantity_entry = ttk.Entry(order_info_frame)
quantity_entry.grid(row=2, column=1, sticky=(tk.W, tk.E))

# Contact information section
contact_info_frame = ttk.LabelFrame(main_frame,
text="Контактная информация", padding="10 10 10 10")
contact_info_frame.grid(row=1, column=0, sticky=(tk.W, tk.E,
tk.N, tk.S))

    ttk.Label(contact_info_frame, text="Ваше имя").grid(row=0,
column=0, sticky=tk.W)
    name_entry = ttk.Entry(contact_info_frame)
    name_entry.grid(row=0, column=1, sticky=(tk.W, tk.E))

    ttk.Label(contact_info_frame, text="Ваш email *").grid(row=1,
column=0, sticky=tk.W)
    email_entry = ttk.Entry(contact_info_frame)
    email_entry.grid(row=1, column=1, sticky=(tk.W, tk.E))

    ttk.Label(contact_info_frame, text="Ваш телефон *").grid(row=2,
column=0, sticky=tk.W)
    phone_entry = ttk.Entry(contact_info_frame)
    phone_entry.grid(row=2, column=1, sticky=(tk.W, tk.E))

# Delivery information section
delivery_info_frame = ttk.LabelFrame(main_frame,
text="Информация о доставке", padding="10 10 10 10")
delivery_info_frame.grid(row=2, column=0, sticky=(tk.W, tk.E,
tk.N, tk.S))

    ttk.Label(delivery_info_frame, text="Адрес *").grid(row=0,
column=0, sticky=tk.W)
    address_entry = ttk.Entry(delivery_info_frame)
    address_entry.grid(row=0, column=1, sticky=(tk.W, tk.E))

    ttk.Label(delivery_info_frame, text="Время
доставки").grid(row=1, column=0, sticky=tk.W)

    time_frame = ttk.Frame(delivery_info_frame)
    time_frame.grid(row=1, column=1, sticky=(tk.W, tk.E))

    hours_var = tk.StringVar()
    minutes_var = tk.StringVar()

    hours_spinbox = ttk.Spinbox(time_frame, from_=0, to=23,
textvariable=hours_var, width=2, format="%02.0f")
    hours_spinbox.grid(row=0, column=0)
    ttk.Label(time_frame, text=":").grid(row=0, column=1)

```

```

minutes_spinbox = ttk.Spinbox(time_frame, from_=0, to=59,
textvariable=minutes_var, width=2, format="%02.0f")
minutes_spinbox.grid(row=0, column=2)

# Run the application
root.mainloop()

create_order_form()

```

Протокол работы:

Process finished with exit code -1

Задание 2:

Постановка задачи: Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ №№ 2 – 9(дано трёхзначное число. Вывести сначала его последнюю цифру (единицы) а затем среднюю(десятки))

Код программы:

```

import tkinter as tk

def check_number():
    try:
        number = int(entry.get())
        if number > 99 and number < 1000:
            result1_label.config(text=f"первая цифра: {number // 100}")
            result2_label.config(text=f"средняя цифра: {number % 100 // 10}")
        else:
            result1_label.config(text="Введите ТРЁХЗНАЧНОЕ целое число")
            result2_label.config(text="")
    except ValueError:
        result1_label.config(text="Введите целое число")

# Создаем окно
root = tk.Tk()
root.title("Проверка числа")

# Создаем виджеты
entry_label = tk.Label(root, text="Введите целое число:")
entry_label.pack()
entry = tk.Entry(root)
entry.pack()

```

```

check_button = tk.Button(root, text="Проверить",
command=check_number)
check_button.pack()

result1_label = tk.Label(root, text="")
result1_label.pack()

result2_label = tk.Label(root, text="")
result2_label.pack()

# Запускаем цикл обработки событий
root.mainloop()

```

Протокол работы: Process finished with exit code 0

Задание 3

Постановка задачи: 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

Код программы:

```

import os
import shutil

os.chdir("../PZ-11")
files_in_pz11 = [f for f in os.listdir() if os.path.isfile(f)]
print("Список файлов в каталоге PZ_11:", files_in_pz11)

# 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.

os.chdir('../')
os.makedirs('test/test1', exist_ok=True)

file_from_pz6_1 = 'PZ-6/task_1/6.1.py'
file_from_pz6_2 = 'PZ-6/task_2/6.2.py'

```

```

file_from_pz7 = 'PZ-7/task_1/7.1.py'

shutil.move(file_from_pz6_1, 'test/')
shutil.move(file_from_pz6_2, 'test/')
shutil.move(file_from_pz7, 'test/test1/test.txt')


files_in_test = [f for f in os.listdir('test') if
os.path.isfile(os.path.join('test', f))]
for file in files_in_test:
    file_size = os.path.getsize(os.path.join('test', file))
    print(f"Размер файла {file} в папке test: {file_size} байт")


# Найти файл с самым коротким именем в папке PZ_11
os.chdir('./PZ-11')
shortest_name_file = min((f for f in os.listdir() if
os.path.isfile(f)), key=len)
print("Файл с самым коротким именем:",
os.path.basename(shortest_name_file))


# перейти в любую папку где есть отчет в формате .pdf и
«запустите» файл в
# привязанной к нему программе. Использовать функцию
os.startfile().
pdf_report_path = '../reports\Отчёт PZ-5.pdf'
os.startfile(pdf_report_path)


# удалить файл test.txt.
test_file_path = '../test/test1/test.txt'

os.remove(test_file_path)
print("Файл test.txt удален")

```

Протокол работы:

Список файлов в каталоге PZ_11: ['11.1.py', '11.2.py', 'data.txt', 'data_1.txt', 'text18_15(result).txt', 'text18_15.txt']

Размер файла 6.1.py в папке test: 0 байт

Размер файла 6.2.py в папке test: 0 байт

Файл с самым коротким именем: 11.1.py

Файл test.txt удален

Process finished with exit code 0

Вывод: Выполняя практическую работу, я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрёл навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community, изучил возможности модуля OS