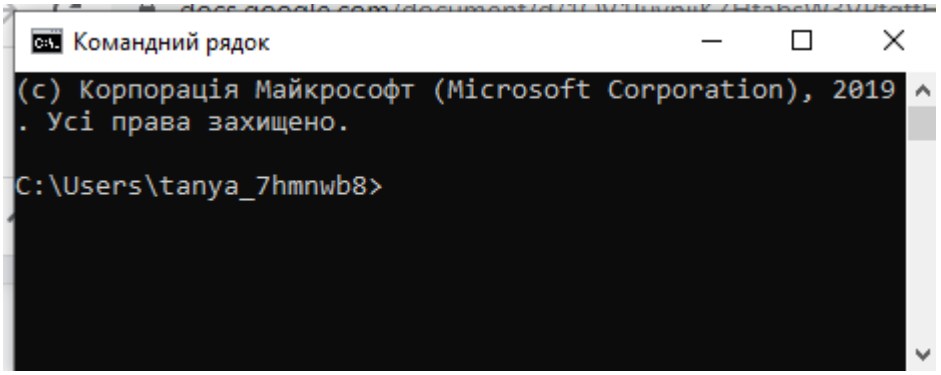
Інтерфейс користувача на мові Python. Створення вікон та налаштування їх властивостей

1. Інтерфейс користувача у мові Python

Інтерфейс *– це сукупність засобів і правил, що забезпечують взаємодію пристроїв обчислювальної системи або програм з користувачем.*

Існує 2 види інтерфейсу програм:

- **командний:** взаємодія користувача та комп'ютера відбувається за допомогою команд, які користувач вводить в командний рядок з клавіатури



- **графічний** інтерфейс є більш зручнішим для користувача, адже взаємодія відбувається за допомогою **кнопок, полів, та інших елементів управління**.

Ми навчимося створювати мовою програмування Python програми, які будуть мати свій власний графічний інтерфейс.

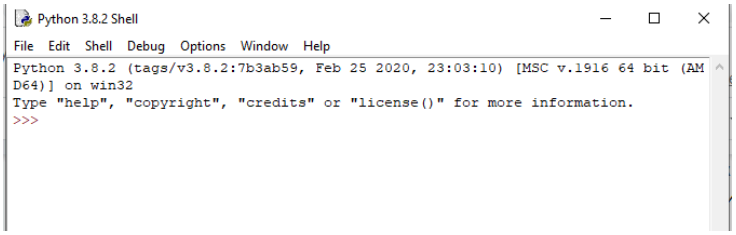
Форма *- об'єкт, в якому можна розмістити різні компоненти (елементи керування), зокрема кнопки, поля, написи, меню та інше. Зазвичай форми представляють собою звичайні програмні вікна,* в яких будуть відображатися вищезазначені елементи. Тож і ми спробуємо створити вікно програми мовою Python з різноманітними властивостями.

Мова Python більше орієнтована на командний інтерфейс, але в ній існує спеціальний модуль за допомогою якого можна створити інтерфейс користувача.

Модулем *у програмуванні називають певний пакет додаткових*

функцій для роботи над розробкою програм.

Сам IDLE не зможе створити окреме вікно, максимум, що ми можемо отримати – це рядки з текстом, безпосередньо у вікні самого IDLE.



Потрібно створити новий файл і там писати програму. Оскільки IDLE не може створювати окреме вікно програми самостійно, то для цього ми повинні підключити додатковий модуль **tkinter**.

Для **підключення будь-яких модулів** використовується конструкція:

import *назва_модуля*

Для того, **щоб повторно не вказувати модуль при використанні функцій** використовують конструкцію:

from *назва_модуля* **import ***

Знак "*****" вказує, що з даного модуля **ми імпортуємо всі функції**.

Модуль, який нам знадобиться для створення графічного інтерфейсу, програми називається **tkinter**. За допомогою цього модуля можна виконувати як окремі графічні побудови, так і створити повноцінний графічний інтерфейс користувача.

Перший рядок коду в нашому новому файлі буде завжди виглядати так:

```
from  tkinter  import *          (1)
```

---

2. Створення вікон та налаштування їх властивостей

Після підключення модулів, створимо вікно.

Для створення вікна використовується функція **Tk()**, але, так як таких вікон можна створити нескінченно багато, то кожному вікну присвоюється певне ім'я:

назва_вікна=**Tk()**

Назва вікна може складатися з великих і малих літер англійського алфавіту, цифр та знаку нижнього пробілу. Наприклад, створимо вікно *Window1*:

Window1=Tk() **(2)**

Назва вікна потрібна для того, щоб потім при написанні коду ми змогли звертатися до даного вікна, присвоювати йому певні властивості та вставляти у нього елементи управління.

Важливо!!! після створення вікна та його елементів управління потрібно вказати інтерпретатору, що ми закінчили працювати з вікном за допомогою метода ***mainloop()***. Застосовується він так:

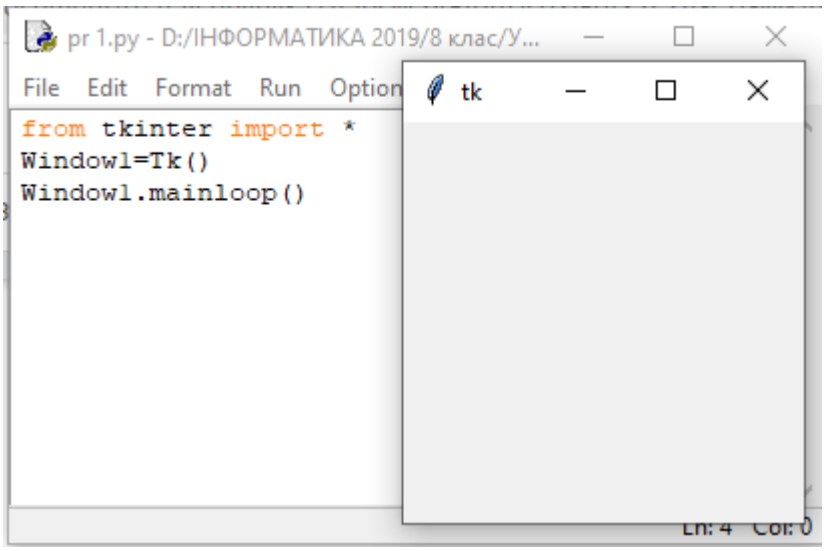
назва_вікна.**mainloop()**

У випадку із нашим вікном цей рядок виглядатиме так:

Window1.mainloop() **(3)**

Ці всі команди та рядки коду (1-3) об'єднаємо у один файл, який ми створили та подивимося на те, що ми отримали. Спробуємо запустити нашу програму. Як ми бачимо у нас утворилося пусте вікно:

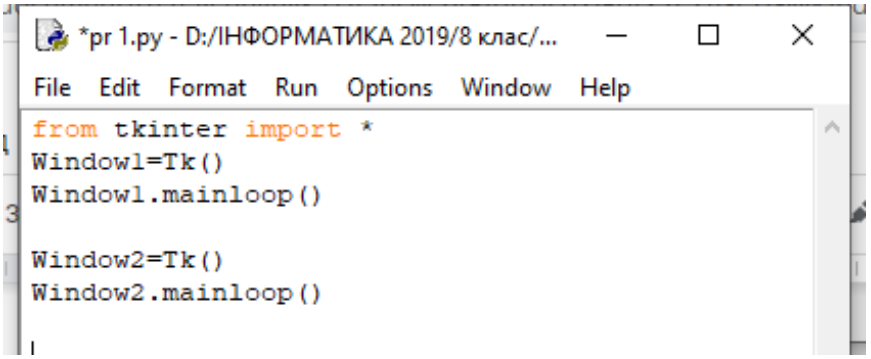
Приклад №1



Можна створювати велику кількість вікон, але в залежності від того, як ми розмістимо рядки коду, ці вікна будуть по різному відображатися.

Приклад №2

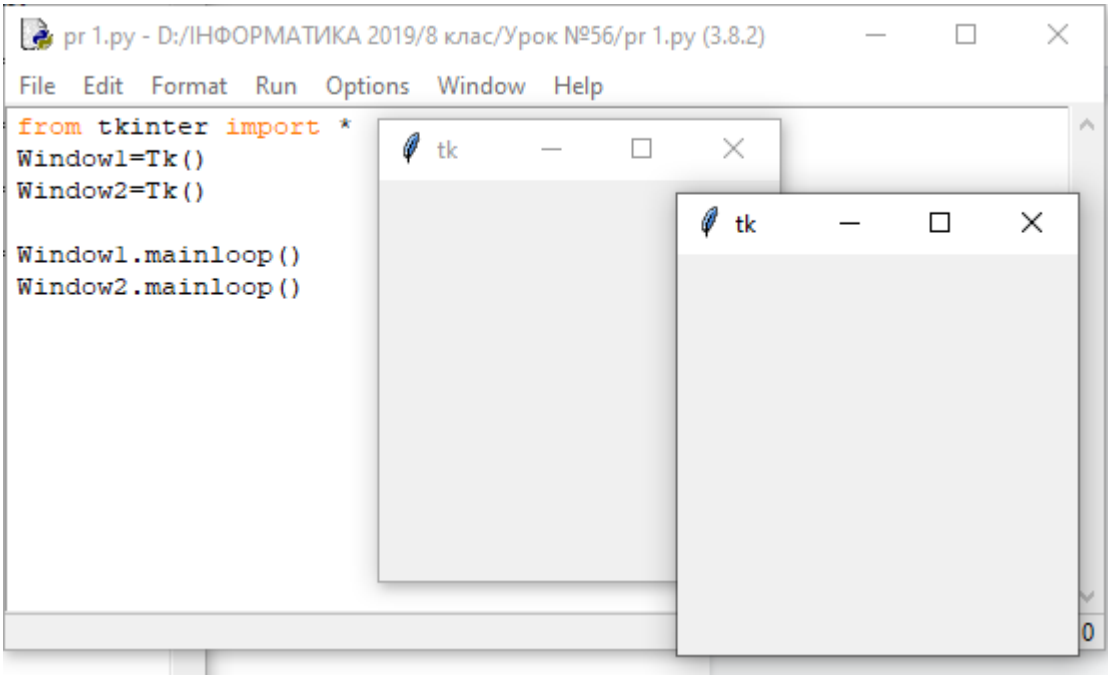
Якщо функцію створення та метод закриття одного вікна ми розмістимо послідовно з функціями та методами іншого, *то друге вікно буде створене тільки після того, як користувач закрив перше*:



Для того, щоб краще зрозуміти написане, рекомендую повторити дії та запустити команду і побачити на практиці, як це працює.

Приклад №3

А для того, щоб створити ці вікна паралельно, ми повинні послідовно розмістити відповідні команди між собою:



Властивості вікна:

• **title(“**Заголовок**”)** – заголовок вікна. За замовчуванням заголовок вікна «Tk»;

• **minsize(**x,y**)** – мінімальний розмір вікна у пікселях, x- ширина, y – висота. Якщо не встановити цю властивість, то вікно не матиме обмежень у зменшенні;

• **maxsize(**x,y**)** – максимальний розмір вікна у пікселях, , x- ширина, y – висота. Якщо не встановити цю властивість, вікно може приймати повні розміри екрану користувача;

• **geometry(“**400**x**200**+**450**+**150**”)** – розміри та положення вікна відносно розширення екрана у пікселях. 400 - ширина, 200-висота, 450- відстань від лівого краю екрану, 150 – відстань від верхнього краю екрану. Параметри відступу (відстань від лівого краю екрану та відстань від верхнього краю екрану) не є обов’язковими, вони додаються за бажанням. **Зверніть увагу!** Всі числа були введені як приклад;

• **resizable(**x,y**)** – чи може користувач змінювати розміри вікна і на скільки: x – ширина, y – висота. **Для заборони змінення розмірів вікна замість параметрів встановити нулі: resizable(**0,0**)**

До вікна властивості застосовуються таким чином:

назва_вікна.*властивість*(*параметри*)

Наприклад, **встановимо розміри нашого вікна** Window1: ширина та висота по 500 пікселів:

Window1.geometry(“500x500”) (4)

Аналогічно застосовуються й інші властивості.

Колір фону вікна

Щоб застосувати певний колір фону для вікна виконується інша конструкція:

назва_вікна[**“bg”**]=“колір”

Bg скорочено від background (фон), а параметром є назва кольору на англійській мові взята в лапки.

Наприклад, встановимо зелений колір для нашого вікна:

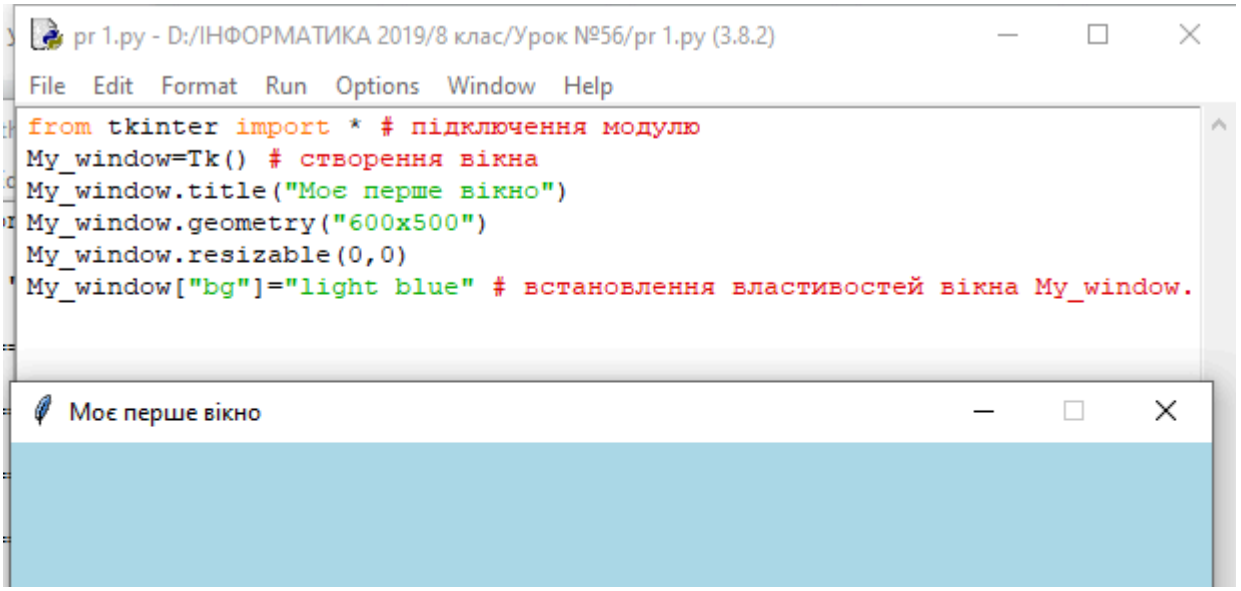
Window1[“bg”]=“green” (5)

На самостійне опрацювання! Знайти якомога більше кольорів, які можливо встановити до вікна tkinter в мові програмування Python.

Приклад №4

Створити вікно світло-блакитного кольору зі сталими розмірами: ширина 600, висота 500 та з заголовком «Моє перше вікно».

Спочатку створюємо вікно, назвемо його як завгодно, наприклад ***My_window***. Потім встановимо його розміри за допомогою властивості **geometry**, заборонимо змінювати його розміри та встановимо світло блакитний колір (**light blue**). Не забуваємо додати заголовок вікна та закриваючу команду **mainloop**. Ось що ми отримали після запуску:



Контрольні питання

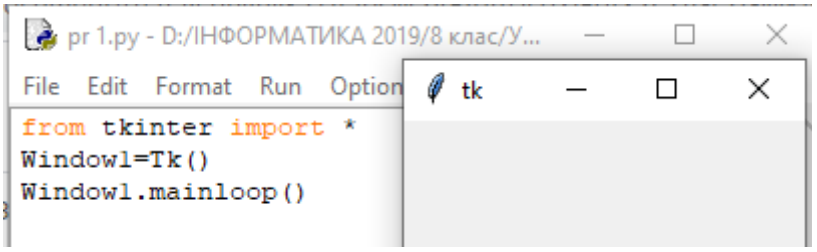
1. Які існують два види інтерфейсу? Чим вони між собою відрізняються?
2. Що таке форма?
3. Що представляють собою форми в мові програмування Python?
4. Як називається модуль для створення графічного інтерфейсу користувача на мові Python? Як його підключити?
5. Як створити вікно на мові Python з використанням модуля tkinter?
6. Які властивості можна надати вікну? Як їх застосовувати?

Практичні завдання до уроку №48

Увага! *При роботі за комп'ютером пам'ятайте про дотримання правил техніки безпеки та виконання санітарно-гігієнічних норм.*

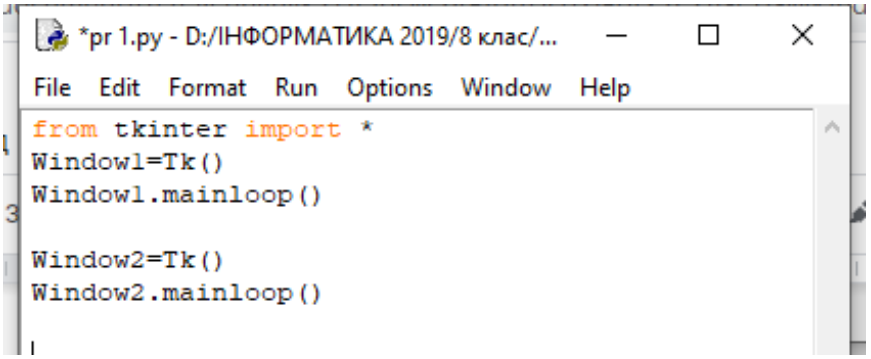
Завдання №1

Напишіть і запустіть код програми для створення вікна за допомогою модуля **tkinter**.



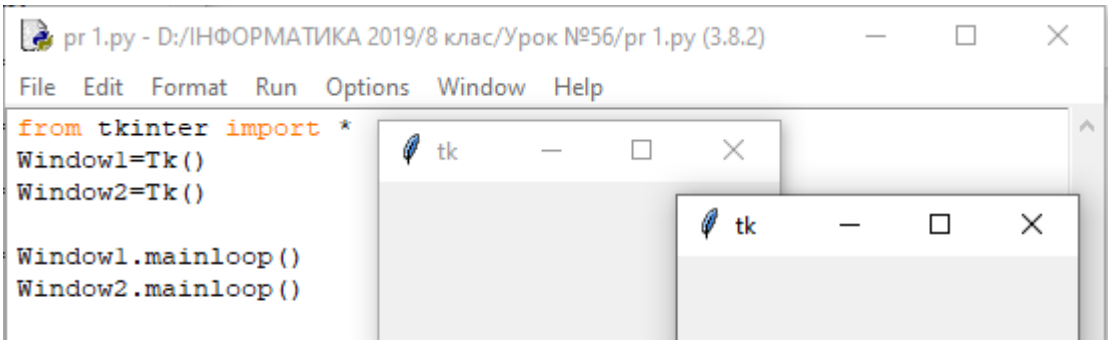
Завдання №2

Напишіть і запустіть код програми для послідовного створення двох вікон (щоб друге вікно відкривалося лише після закриття першого) за допомогою модуля **tkinter**.



Завдання №2.1 Напишіть і запустіть код програми для послідовного створення 4 вікон (щоб друге вікно відкривалося лише після закриття першого) за допомогою модуля **tkinter**.(**див. завдання 2**)

Завдання №3 Напишіть і запустіть код програми для одночасного створення двох вікон за допомогою модуля **tkinter**.

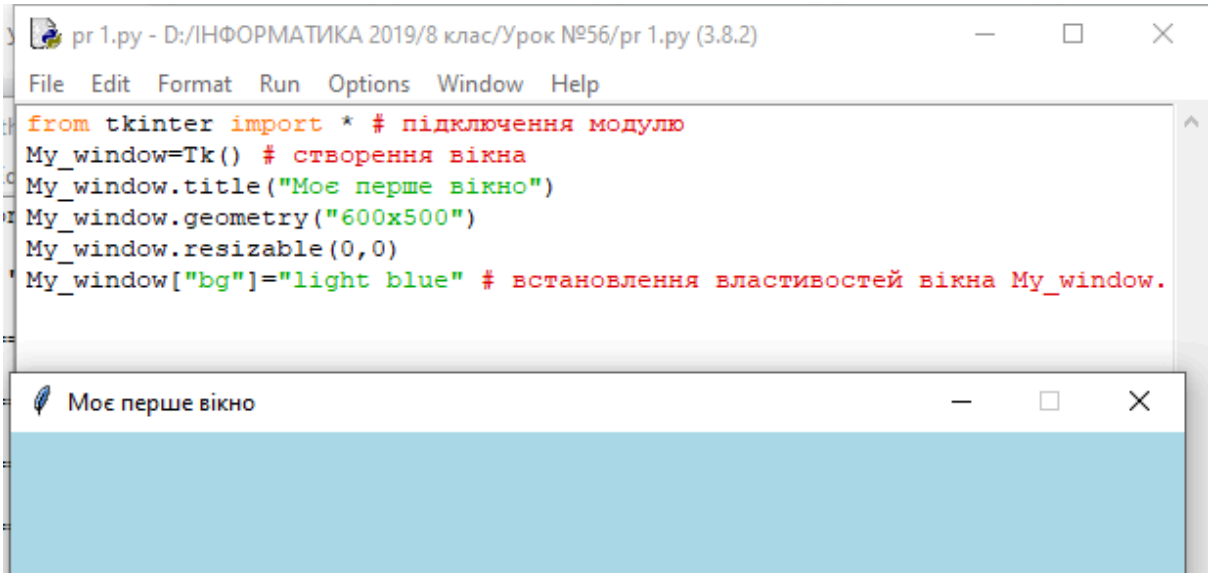


Завдання №3.1 Напишіть і запустіть код програми для одночасного створення 4 вікон за допомогою модуля **tkinter**.(**див. завдання 3**)

Завдання №4Створити вікно світло-блакитного кольору зі сталими розмірами:

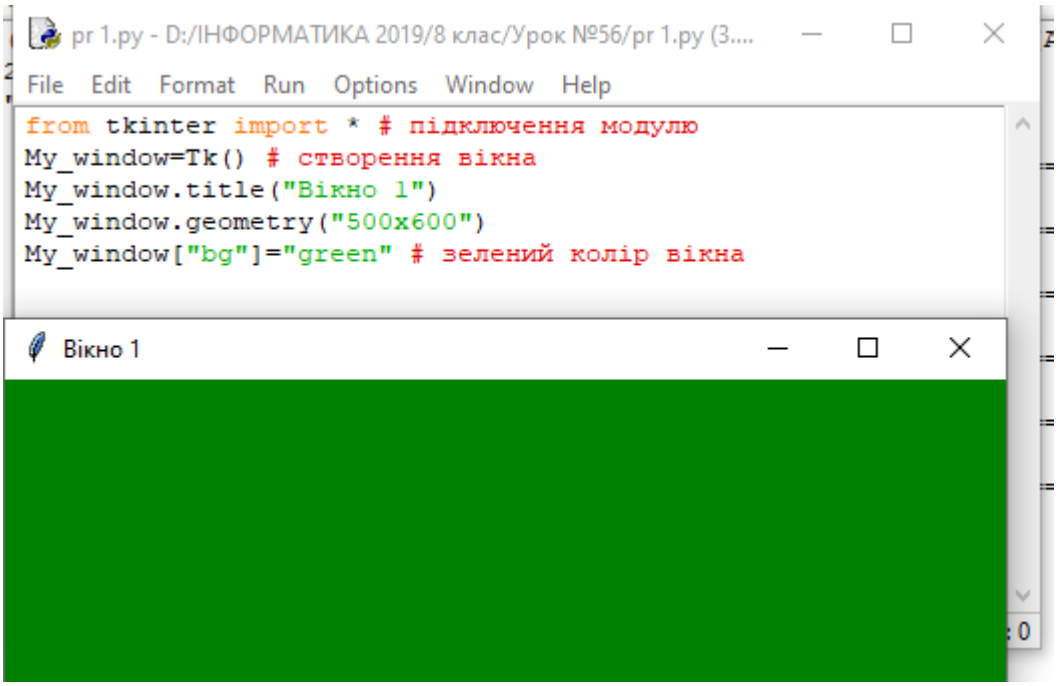
ширина 600, висота 500 та з заголовком «Моє перше вікно».

Спочатку створюємо вікно, назвемо його як завгодно, наприклад ***My_window***. Потім встановимо його розміри за допомогою властивості **geometry**, заборонимо змінювати його розміри та встановимо світло блакитний колір (**light blue**). Не забуваємо додати заголовок вікна та закриваючу команду **mainloop**. Ось що ми отримали після запуску:



Завдання №5

Створіть новий файл Python. Підключіть відповідний модуль та створіть вікно зеленого кольору, з розмірами 500х600, та заголовком “Вікно 1”.



Завдання №6

Створіть новий файл Python. Підключіть відповідний модуль та створіть вікно жовтогарячого кольору, з розмірами 400х400, з заголовком “Це вікно!”. (**див. завдання №5**)

Завдання №7

Створіть новий файл Python. Підключіть відповідний модуль та створіть вікно білого

кольору, з розмірами 1000x500, з заголовком “Заголовок вікна”, та з забороною змінювати розміри вікна. (**див. завдання №4**)

Завдання №8

Створіть новий файл Python. Підключіть відповідний модуль та створіть вікно жовтого кольору, з розмірами 654x456, відступом від лівого краю 300 та відступом від верхнього краю 400. Заголовок “Вікно8”, та заборона змінювати розміри вікна.

Завдання №9

Створіть новий файл Python. Підключіть відповідний модуль та створіть вікно блакитного кольору, з розмірами 700x300, відступом від лівого краю 100 та відступом від верхнього краю 0. Заголовок “Вікно9”, та заборона змінювати розміри вікна. Та створити вікно із заголовком “Вікно10” синього кольору, з ідентичними розмірами та забороною змінювати розмір вікна. Вікна розташувати послідовно, тобто 2-ге вікно відкриється лише після закриття першого. **(див. завдання 2 та завдання 4).**

Додаток

Кольори для Python (tkinter)