

PYTHON

5 клас

Любінецька Алла Павлівна

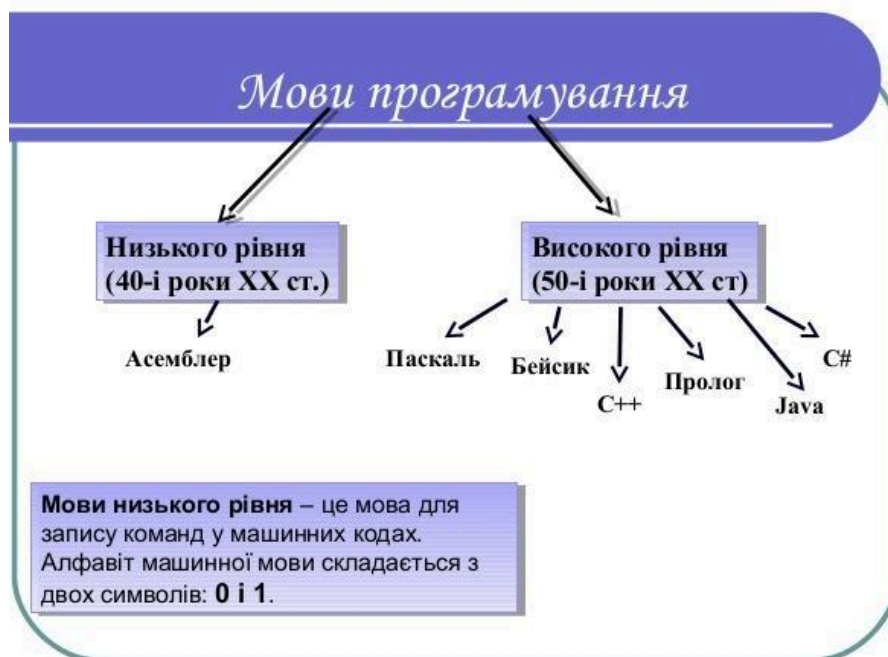
§ 1. Мови програмування, що це?



Мови програмування — це формальні комп'ютерні мови. Звичайно, в реальному житті на них ніхто не розмовляє. Вони потрібні для "переговорів" з машинами. На різних мовах програмування пишуться різні програми, які висловлюють певні алгоритми або контролюють поведінку комп'ютера. Від природних мов вони різко відрізняються. І, щоб почати вчити якусь мову, спочатку треба зрозуміти принцип його роботи.

Матір'ю програмування вважають Аду Лавлейс, що створила в 1842 році найпершу значиму програму до аналітичної механічної машини Чарльза Беббіджа. Але, звичайно, перші ключові парадигми, що діють для програмування і сьогодні, були створені в 60-70-х роках минулого століття. До теперішнього часу в школах викладаються основи програмування на уроках інформатики, а професії програміста і веб-розробника вважаються найбільш затребуваними.

Сьогодні неможливо працювати в сфері розробки і знати тільки одну мову програмування. І судити про те, який краще, а який гірше — абсолютно безглуздо, тому що всі вони створені для досягнення максимальної ефективності певних задач. Щороку з'являються нові мови. А основні затребувані розширюються і прогресують.



§ 2. Програмування та його види

Програма визначається як “алгоритм, тобто послідовність інструкцій, який записано мовою, що зрозуміла комп’ютеріві. Програма складається з підсистем, які певним чином взаємодіють між собою. Програма являє собою певну модель, яка описує з деяким наближенням закономірності реального світу. Програма є засобом для вирішення конкретних практичних задач.

Відповідно до цього можна підходити до програмування як до процесу конструювання програм з різних точок зору.

Виділяють чотири основні типи програмування:

1. Процедурне програмування.

Концепція процедурного програмування є історично першою та найбільш близькою до класичного визначення програми.

В основі програми, побудованої за процедурними принципами, лежить послідовна зміна вхідних даних, поки не буде отриманий результат, причому кожна операція розписується в явному вигляді. Дані, з якими оперує програма, зберігаються в іменованих ділянках оперативної пам’яті, які називаються змінними (Паскаль і Сі).

2. Функціональне програмування.

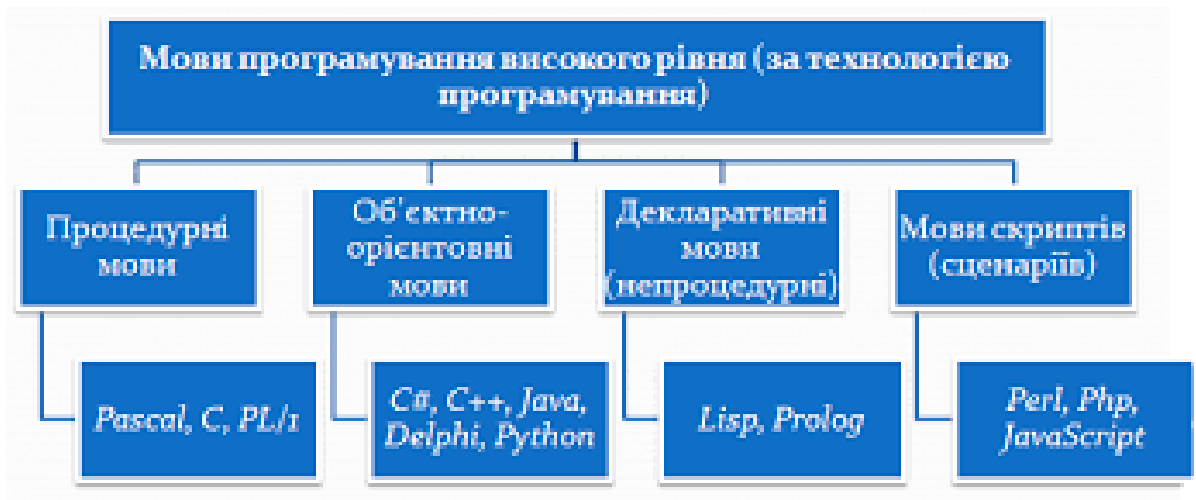
Виконання програми розглядається як виклик деякої функції, яка, в свою чергу, може викликати інші функції.

3. Логічне програмування.

В основі виконання програми лежить механізм виведення нових фактів з даних фактів згідно із заданими логічними правилами. Найбільш відомий представник – Пролог.

4. **Об’єктно-орієнтоване програмування** – найбільш стрімко розвивається в сучасний час.

Об’єктна програма розглядається як сукупність паралельно існуючих об’єктів, які взаємодіють між собою. Кожний об’єкт вміє виконувати певні операції та характеризується певною поведінкою



Принципи об'єктно-орієнтованого програмування:

1. Все є об'єктами.
2. Всі дії та розрахунки виконуються шляхом взаємодії між об'єктами, при якій один об'єкт потребує, щоб інший об'єкт виконав деяку дію.
3. Об'єкти взаємодіють, надсилаючи і отримуючи повідомлення – запити на виконання дії.
4. Кожен об'єкт має незалежну пам'ять, яка складається з інших об'єктів.
5. Кожен об'єкт є представником (екземпляром, примірником) класу, який виражає загальні властивості об'єктів.



Завдання для самостійного виконання

Заповнити таблицю

№	Назва мови програмування	Автор	Дата створення

§ 3. Що таке Python?



Python – популярна у всьому світі багатоцільова та універсальна мова програмування. Це є ідеальний спосіб переходу від візуальних мов, адже її чіткість та простота у використанні стає у нагоді при створенні величезного обсягу продуктів: від роботи над вебплатформами до розробки програмного забезпечення.

Своє існування Python почала ще у 1991 році, а завдяки постійним вдосконаленням армія прихильників лише зростає та налічує зараз сотні тисяч людей. Величезна кількість компаній використовують Python, наприклад: Google, Facebook, Dropbox, Mozilla, NASA, Toyota та інші.

Python проста у використанні, та водночас повноцінна мова програмування, що має багато засобів для структурування і підтримки великих програм. Python має велику бібліотеку стандартних модулів, які можна використовувати як основу для нових програм або як приклади при вивченні мови. Стандартні модулі надають засоби для роботи з файлами, системними викликами, мережними з'єднаннями і навіть інтерфейсами до різних графічних бібліотек.

Python - інтерпретована мова, що дозволяє заощадити значну кількість часу, що зазвичай витрачається на компіляцію. Python дозволяє писати дуже компактні й зручні для читання програми.

Python використовується в даний час десятками тисяч програмістів в усьому світі, і число людей, що використовують його, швидко зростає, подвоюється і потроюється щороку.

Python широко використовується в різних середовищах. Будучи високо адаптованою мовою програмування, Python дозволяє легко розробляти та підтримувати проекти різного рівня складності. Найбільші переваги Python – це гнучкість, швидкий розвиток, масштабованість і відмінна продуктивність.

NASA використовує Python в проектах, що стосується математичних розрахунків параметрів польоту.

Python є важливою частиною Google. Python визнана офіційною мовою в Google, працює на багатьох внутрішніх системах Google і відображається у багатьох API Google. Python ідеально підходить для інженерного процесу в Google.

YouTube використовує Python для різних цілей: перегляд відео, керування шаблонами для веб-сайту, адміністрування відео, доступ до канонічних даних і багато іншого.

Мова програмування Python названа не на честь пітонів.

Хоча іконки файлів з написаними на ньому скриптами або логотипи середовищ розробки часто являють собою зміїну голову.

Ім'я мови автор Гвідо ван Россум вибрав через те, що був шанувальником Британського комедійного скетч-серіалу «Літаючий цирк Монті Пайтона». Ось в честь «Монті Пайтона (Monty Python)» мова названа Python.



Python займає п'яте місце в світі за популярністю!

Мова програмування Python – досить проста у вивченні.

Мову програмування Python вважають молодого. Насправді, розробка Python розпочалася в кінці 1980-х років, а перша повноцінна версія з'явилася у 1991 році.

Так що своє повноліття мова Python вже відзначила.

§ 4. Встановлення Python

Встановлення PYTHON в ОС Windows є простим. Досить мати інсталяцію. Завантажити можна з офіційного сайту python.org. Запустіть її на своїй машині; вкажіть папку, в яку інтерпретатор має бути встановлений.

Увага! Обирайте версію з врахуванням того, яку розрядність (32 чи 64 біта) має ваша версія Windows!

Наступний крок - вибір встановлюваних компонентів. Краще включити встановлення всього, що пропонується (зокрема, документацію). Повний комплект займає на жорсткому диску приблизно 20 мегабайт. Нарешті, потрібно задати назву групи для головного меню. Можна просто натиснути "Next". Програма покаже зведену інформацію про те, що і куди буде встановлено. Ще раз натискайте "Next" і чекайте закінчення. Натискайте "Finish" для виходу з програми встановлення.

Якщо у вас Windows 8 чи 10, то на робочому столі іконка може не з'явитися. Встановіть її, скопіювавши з меню "ПУСК".

Для роботи нам знадобиться додатково середовище **Wing IDE 101**, яке також потрібно встановити.

Завдання для самостійного виконання

1. Встановити Python на свій комп'ютер (або інший гаджет).
2. Записати 5 цікавих фактів про мову Python.

§ 5. Працюємо в інтерактивному режимі

Існує два способи використання інтерпретатора: командний режим і режим виконання програм з файлів. Якщо в командному рядку інтерпретатора Python ви наберете команду, то інтерпретатор одразу виведе результат її виконання. Такий режим називається **інтерактивним**.

Інтерактивність (від англ. *interaction*) - обмін діями, взаємодія.

У широкому розумінні для об'єктів інтерактивність - це стан постійно перебувати у взаємозв'язку з іншими об'єктами, це залежність від різних станів інших об'єктів.

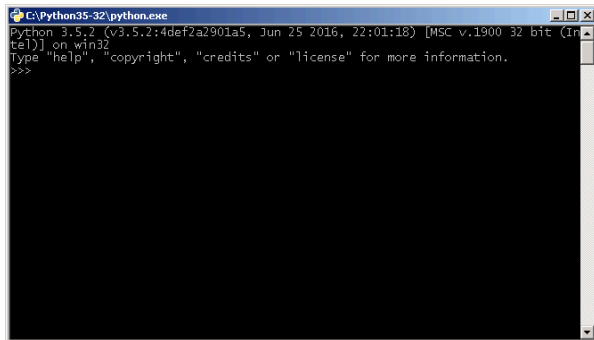
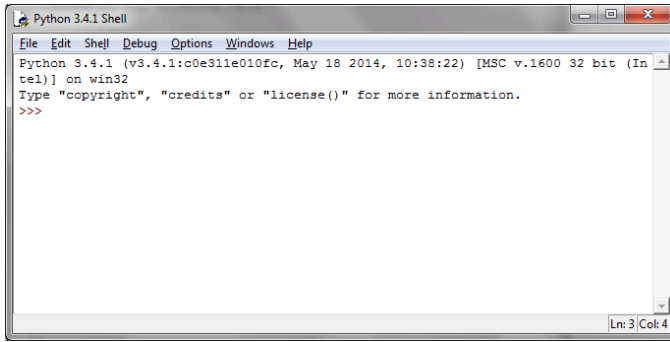
Ми будемо розглядати подібну взаємодію, але між користувачем (програмістом) та інтерпретатором мови Python:

- користувач пише команду, передає її на виконання;
- інтерпретатор намагається перекласти введене, якщо спроба вдала, команда виконується і на екрані ми бачимо результат виконання, якщо спроба виявилась невдалою, на екрані відображається повідомлення про помилку;
- користувач бачить повідомлення на екрані, реагує на нього, змінюючи команду чи дописуючи нову;
- інтерпретатор знову починає роботу;
- далі реакція користувача, потім інтерпретатора ...

Отже, під **інтерактивністю** ми будемо розуміти здатність системи, без участі людини, активно і різноманітно реагувати на дії користувача.

У інтерактивному режимі запрошення введення наступної команди виводиться у вигляді первинного запрошення, зазвичай, три знаки "більше" (`>>>`); для продовження введення рядків видається вторинне запрошення (...).

У стандартний комплект постачання Python входить **інтегроване середовище розробки IDLE**, в якому редагувати програми буде набагато зручніше, ніж в простому текстовому редакторі. IDLE написано на Python. IDLE так само має вбудовану систему відлагодження (процес пошуку помилок).



У операційній системі Windows для виклику інтерпретатора досить в меню **"Пуск > Програми > Python 2.7.15 > IDLE"** викликати середовище розробки IDLE. Інтерпретатор друкує вітаюче повідомлення, констатуючи його номер версії і зауваження про авторське право, перед видачею першого запрошення:

```
>>> print "Hello world!"
```

Hello world!

Запрошення інтерпретатора `>>>` означає, що він готовий виконувати команди. Ми набрали команду **print "Hello world!"**, тобто дали вказівку вивести на екран рядок "Hello world!", і в наступному рядку інтерпретатор вивів те, що ми просили.

Є загальне правило - спочатку треба навчитися тому, як виходити з якоїсь ситуації, системи, програми, перед тим, як робити ще щось. Щоб завершити роботу з інтерпретатором Python, у Windows можна скористатися комбінацією клавіш **Ctrl+Z**.

§ 6 . Python як калькулятор. Операції та оператори

В попередньому параграфі ми працювали з **Python** в інтерактивному режимі. Ми ввели команду (*директиву*), а інтерпретатор **Python** її тут же виконав.

Виконати команду, директиву або програму означає те ж саме, що і «запустити» її.

Обчислення в Python

Давайте спробуємо виконати щось ще в режимі взаємодії.

```
>>> 2+3
```

```
5
```

Ми змогли змусити **Python** виконати приклад, якій *самі вміємо виконувати* ще з першого класу. А якщо потрібно зробити дещо складніше? Давайте спробуємо:

```
>>> 123654789+987456321
```

```
1111111110
```

```
>>> 5236785412-1452369
```

```
5235333043
```

Якщо потрібно перемножити два числа, наприклад дізнатися, скільки буде $15 \times 3,14$, ми зазвичай користуємося калькулятором або беремо ручку і множимо в стовпчик на аркуші паперу. А що якщо використовувати для підрахунків оболонку **Python**?

```
>>> 15*3.14
```

```
47.1
```

Зверніть увагу, що під час запису числа $3,14$ використовується не кома, а крапка. Крім того, в **Python** числа перемножуються за допомогою зірочки (*), а не знаку множення ($\times$).

Для прикладу, спробуємо обчислити *скільки секунд міститься в одному місяці*. Для цього нам необхідно знайти кількість секунд в 1 годині, потім – в 1 добі, а лише потім – в 1 місяці.

```
>>> 60*60
```

```
3600
```

```
>>> 3600*24
86400
>>> 86400*31
2678400
```

Або простіше:

```
>>> 60*60*24*31
2678400
```

Отже, **Python** вміє виконувати різні математичні *операції*. Причому, числа над якими виконується операція називаються *операндами*, а знаки, якими позначається дана операція – *операторами*.

Оператори в Python

В оболонці **Python** можна множити, додавати, віднімати і ділити числа, а також здійснювати деякі інші операції. Символи, за допомогою яких виконуються математичні дії в мові **Python**, називаються *операторами*.

Основні математичні оператори

Назва операції	Оператор	Приклад застосування
Додавання	+	45+65
Віднімання	-	12-23
Множення	*	5*8
Ділення	/	14/6
Піднесення до ступеня	**	2**6
Ділення націло (неповна частка)	//	14//3
Остача від ділення	%	14%3

Порядок виконання операцій

Операції - це будь-які дії, які виконуються за допомогою операторів. Математичні операції виконуються по черзі в залежності від їх пріоритету (якщо не поставити іншу черговість за допомогою дужок). Множення і ділення мають більш високий пріоритет, ніж додавання і віднімання, і це значить, що вони будуть виконуватися першими. Інакше кажучи, при обчисленні математичного виразу **Python** спочатку перемножить і розділить числа, а потім перейде до додавання і віднімання. Наприклад, в цьому виразі спершу будуть перемножені числа 30 і 20, а потім до їх добутку буде додано число 5.

```
>>> 5+30*20  
605
```

По суті цей вираз означає *«помножити 30 на 20 і додати до результату 5»*. Виходить 605. Однак ми можемо змінити порядок операцій, помістивши перші два числа в дужки.

```
>>> (5+30)*20  
700
```

В результаті вийшло 700, а не 605, оскільки **Python** виконує операції в дужках перш, ніж операції поза дужками. Іншими словами, цей вислів означає *«додати 5 до 30 і помножити результат на 20»*. Дужки можуть бути вкладеними, тобто всередині дужок можуть стояти ще одні дужки:

```
>>> ((5+30)*20)/10  
70.0
```

У цьому прикладі **Python** спершу обчислить вираз у внутрішніх дужках, потім у зовнішніх і в самому кінці виконає операцію ділення, яка знаходиться за дужками. Інакше кажучи, цей вислів означає *«додати 5 до 30, потім помножити результат на 20, потім розділити результат на 10»*. Ось що при цьому відбувається:

- ☐ додавання 5 і 30 дає 35;
- ☐ множення 35 на 20 дає 700;
- ☐ ділення 700 на 10 дає відповідь 70.

Якби ми не використовували дужки, результат вийшов би іншим.

§ 7. Операції з текстом

Крім арифметичних, **Python** непогано справляється і з текстовими операціями. Зокрема він уміє гарно «додавати» та «множити» текстові фрагменти, вилучати з тексту символи або групу символів.

1. **Конкатенація** – додавання (об'єднання) рядочків. Виконується ця операція за допомогою знаку «+». Але потрібно пам'ятати – текст обов'язково повинен бути поміщеним у лапки (одинарні ' ' або подвійні " ").

```
>>> "Мені "+"подобається "+"програмувати!"  
'Мені подобається програмувати!'
```

2. **Дублювання** – множення рядочків. Виконується за допомогою знака '*'. Якщо рядок помножити на деяке число, то даний рядок продублюється таку кількість разів.

```
>>> 'Python ' * 20  
' Python Python Python Python Python Python Python Python Python Pyth  
on PythonPython Python Python Python Python Python Python Python Pyt  
hon Python Python '
```

3. **Вилучення** – витягування з рядочка символу. Кожен символ (літера, цифра, знак, пробіл) має свій порядковий номер, причому нумерація починається з 0, а не з 1.

```
'Python '  
0 1 2 3 4 5 6
```

Щоб вилучити символ, потрібно після рядка вказати номер цього символу в квадратних дужках.

```
>>> 'Python '[2]  
't'  
  
>>> 'Python '[5]  
'n'  
  
>>> 'Python '[6]  
' '
```

Шостим символом є пробіл, тому програма виведе на екран пробіл в лапках.

4. **Визначення довжини рядка** – це визначення кількості символів у даному рядку. Для цього використовується команда **len**.

```
>>> len('Python')
```

```
6
```

```
>>> len('Python ')
```

```
7
```

5. **Вилучення зрізу символів** – це витягування з рядочка групи символів. В квадратних дужках потрібно вказати номер першого та останнього символів, які хочемо вилучити.

```
>>> 'Python' [2:4]
```

```
'tho'
```

```
>>> 'Python ' [1:3]
```

```
'yth'
```

6. Якщо потрібно вилучити символи, що розміщені не поруч, то програма буде мати такий вигляд:

```
>>> 'Python' [0:5:2]
```

```
'Pto'
```

В квадратних дужках потрібно вказати номери першого та останнього символів, а третє число означає, що буде вилучений кожен другий символ.

Запитання для учнів

1. Який режим роботи називається інтерактивним?
2. Які середовища програмування Python ви знаєте?
3. Що таке операція та оператор?
4. Які математичні оператори ви знаєте?
5. Як позначаються оператори додавання, віднімання, множення, ділення, піднесення до степеня та знаходження остачі?
6. Які операції можна виконувати над рядками?

Завдання для самостійного виконання

- 1.1 Обчислити значення виразу $24560+5678$.
- 1.2 Обчислити значення виразу $24560-5678$.
- 1.3 Обчислити значення виразу $2456*5678$.
- 1.4 Обчислити значення виразу $2456/5678$.
- 1.5 Піднести до степеня 2 число 24650.
- 1.6 Знайти остачу від ділення числа 24560 на 3.

- 2.1 Обчислити свій вік в годинах.
- 2.2 Обчислити свій вік в секундах.
- 2.3 Обчислити кількість навчальних днів, якщо на навчання ми витрачаємо 9 місяців, з них 5 місяців мають по 31 дню, 3 місяці – 30 днів і один місяць – 28 днів. Ще потрібно врахувати, що 28 днів тривають всі канікули.
Скласти числовий вираз для розв'язання цієї задачі.
- 3.1 Скласти програму, яка об'єднає три рядки (Прізвище, ім'я, по батькові) в один рядок.
- 3.2 Скласти програму, яка дублює на екрані слово 'Hello' десять разів.
- 3.3 Скласти програму, яка визначає та виводить на екран довжину вашого імені.
- 3.4 Скласти програму, яка вилучає з вашого імені третій символ.
- 3.5 Скласти програму, яка вилучає з імені зріз з другого по четвертий символ.

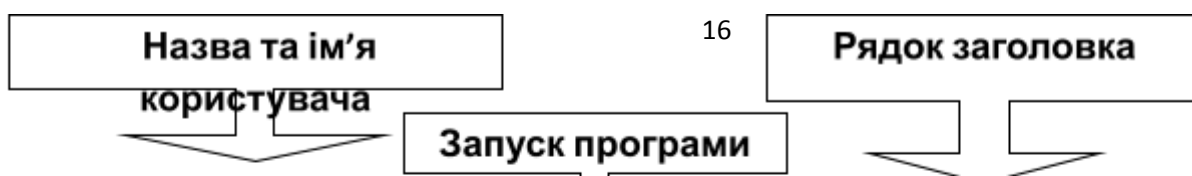
§ 8. Середовище програмування для Python

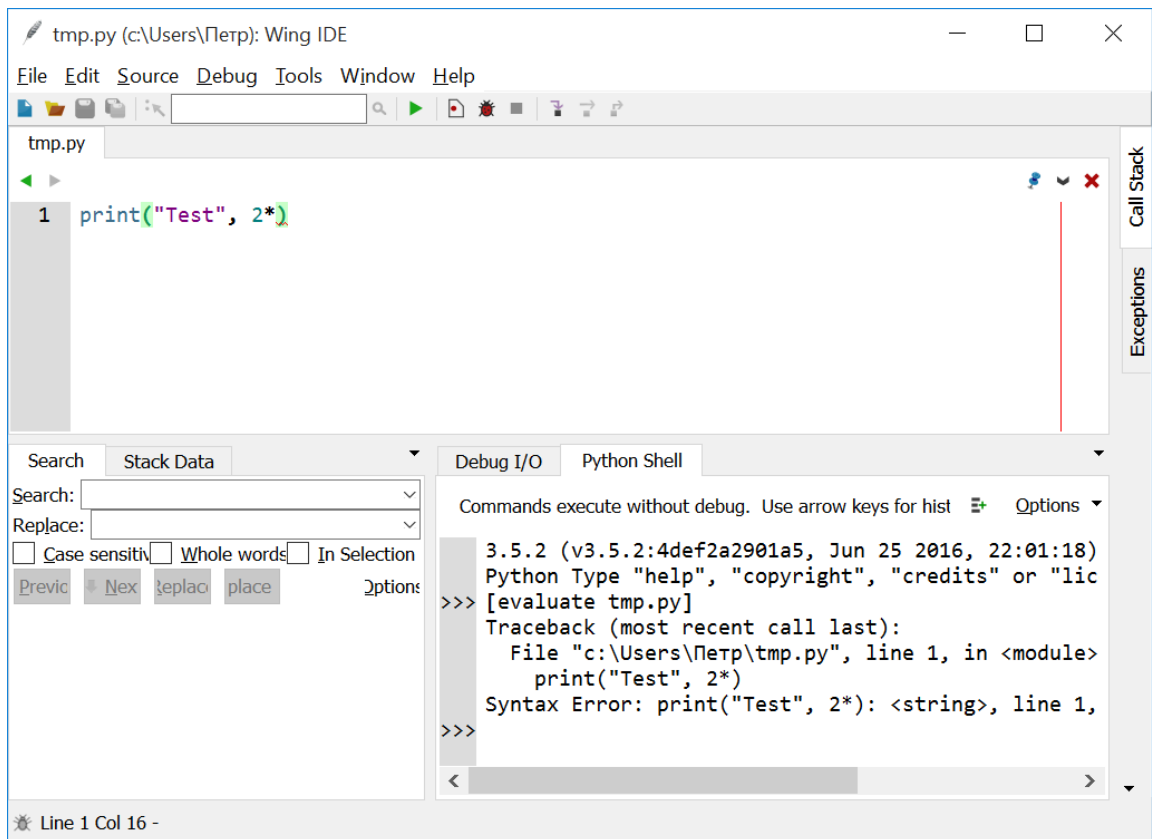
Для написання програм використовують **текстові редактори** або **інтегровані середовища розробки**, які включають в себе різні інструменти для роботи з кодом: засіб для написання коду (текстовий редактор), інтерактивний інтерпретатор, відлагоджувач тощо.

Текстові редактори та інтегровані середовища програмування для Python

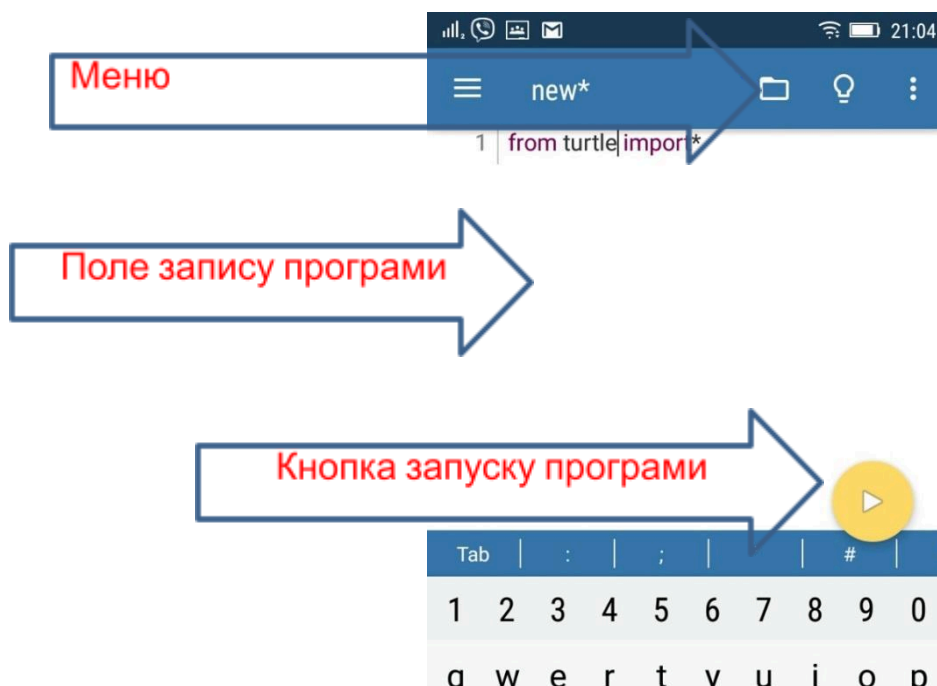
- **IDLE** — стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим пакетом для користувачів Linux.
- **Notepad++** - безкоштовний текстовий редактор вихідного коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.
- **PyScripter** - інтегроване середовище розробки для мови програмування Python, працює під Windows. Поширюється безкоштовно.
- **Wing IDE 101** - вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і Mac OS X. Поширюється безкоштовно.
- **Geany** - вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, Mac OS X і Windows.
- **Sublime Text 3** - кросплатформенний текстовий редактор вихідних текстів програм та інтегроване середовище розробки. Підтримує плагіни, розроблені за допомогою мови програмування Python. Sublime Text не є вільним чи відкритим програмним забезпеченням, але деякі його плагіни розповсюджуються з вільною ліцензією, розробляються і підтримуються спільнотою розробників. Проте, можна використовувати вільно, хоча часто з'являється повідомлення про придбання ліцензії. Для користувачів Windows, Mac OS X, Linux.
- **PyCharm** — інтегроване середовище розробки для мови програмування Python. Підтримує веб-розробку на Django. PyCharm є власницьким програмним забезпеченням. Присутні безкоштовна версія **Community** з усіченим набором можливостей і безкоштовна версія **Edu** для навчальних закладів. PyCharm працює під операційними системами Windows, Mac OS X і Linux.

Інтерфейс середовища програмування Wing IDE 101





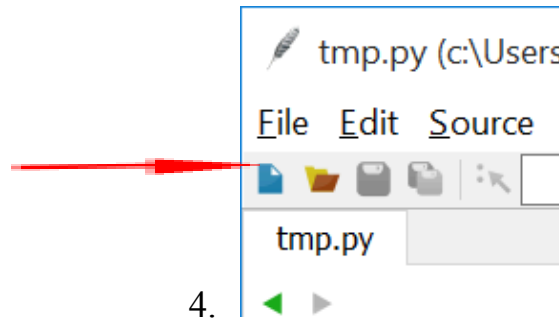
Якщо ви працюєте зі смартфоном, вікно має трохи інший вигляд.



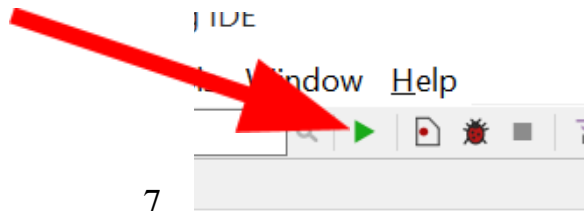
Меню містить вказівки для створення нового файлу та збереження створеного.

§ 9. Як працювати з програмами?

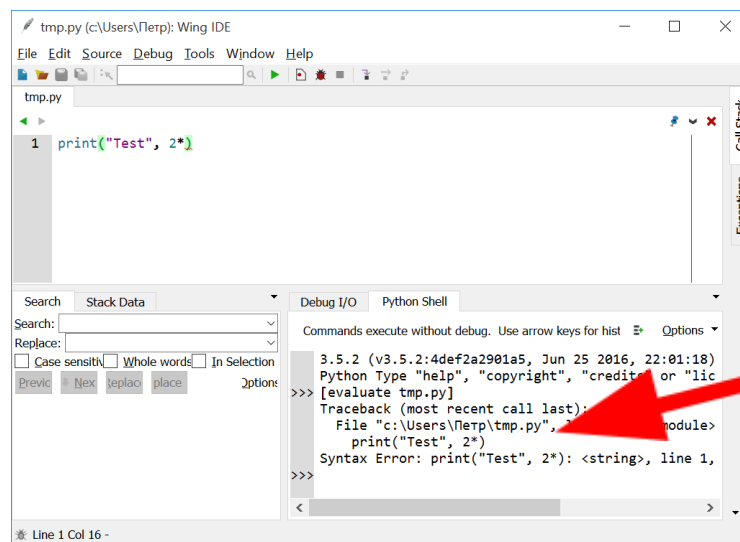
1. Виконувати практичні завдання слід в середовищі програмування.
2. Перед початком роботи запустіть середовище Wing IDE 101.
3. Роботу завжди починайте зі створення нового файлу.



4. В полі запису програми запишіть потрібні команди, приклади, програми.
6. Перевірте виконання програми, натиснувши Запуск, що знаходиться на Панелі інструментів.



7. Якщо програма записана правильно, в полі виконання ви побачите результат.
9. Якщо в програмі є помилки, в полі виконання буде прописано назву помилки та рядок, в якому вона знаходиться.



§ 10. Введення та виведення величин

Операції та операнди

Ви вже знаєте, що операція - це виконання певних дій над даними (операндами). Для виконання конкретних дій потрібні спеціальні інструменти — оператори. Наприклад, символ "+" по відношенню до чисел виконує операцію додавання, а по відношенню до рядків - конкатенацію (з'єднання).



Команди введення та виведення

Для введення та виведення величин використовують певні команди. У мові Python це команди:

- **input** - введення величини (те, що дано в задачі);
- **print** - виведення (відповідь).

Розглянемо на прикладі, як використовуються дані команди.

Проект Сума

Напишемо програму, за допомогою якої можна отримати результати знаходження суми чисел.

Пояснення

У нас є два числа, які ми позначимо літерами **a** та **b**. Отже, нам потрібно ввести два числа за допомогою команди **input**. Рядків програмного коду буде два (для кожного числа окремий рядок). Далі позначимо літерою **s** суму чисел і запишемо математичну дію. Потім виведемо на екран нашу відповідь за допомогою команди **print**.

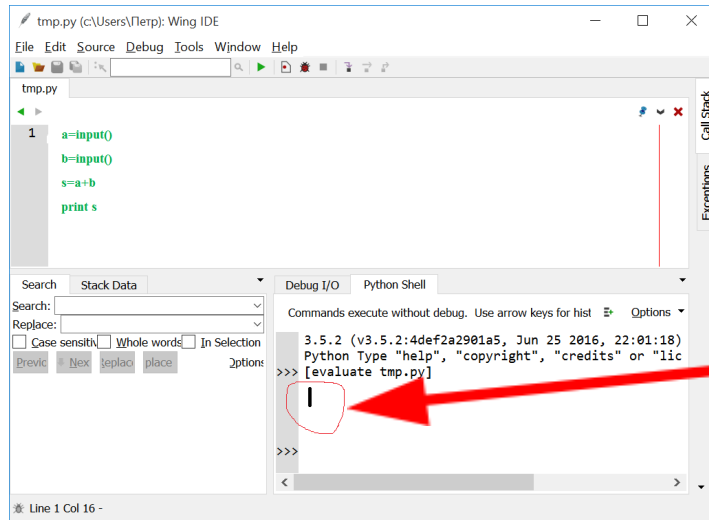
Програма буде мати вигляд:

```
a=input()
b=input()
s=a+b
print s
```

Важливо пам'ятати, що **великі літери ми не використовуємо!!!**

Порожні дужки в програмі означають, що після запуску програми ми зможемо виконати дію для будь-якої пари чисел. Ні в якому разі **не записуйте числа в ці дужки**.

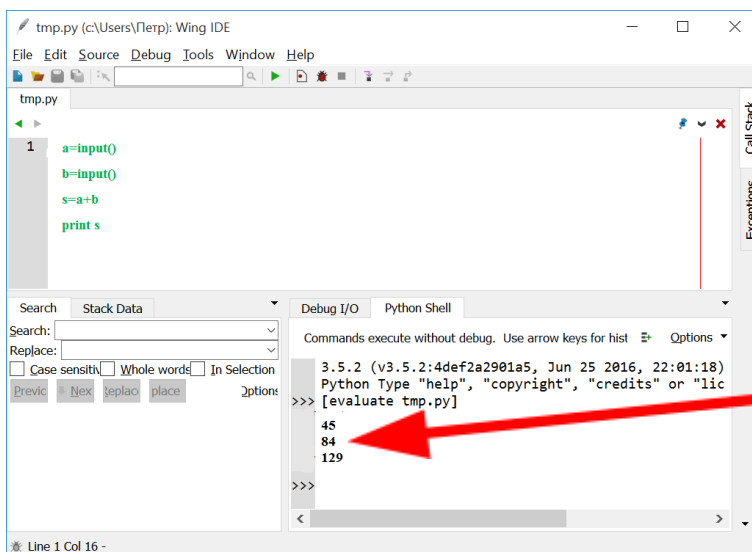
Коли програма створена, можна записати її в середовище програмування.



Після запуску програми в полі виконання відповідь не з'явиться, а буде блимати курсор. Це означає, що програма чекає від нас введення конкретних чисел. Вводимо їх прямо в полі виконання так:



Після цього в полі виконання програми з'явиться відповідь.



Завдання для самостійного виконання

- 1.1 Написати програму, яка обчислює різницю двох чисел.
 - 1.2 Написати програму, яка обчислює добуток двох чисел.
 - 1.3 Написати програму, яка обчислює частку двох чисел.
 - 1.4 Написати програму, яка обчислює суму трьох чисел.
 - 1.5 Написати програму, яка обчислює значення виразу a^2 .
 - 1.6 Написати програму, яка обчислює значення виразу $(a+b) \cdot c$.
 - 1.7 Написати програму, яка обчислює значення виразу $a^2 + b^2$.
-
- 2.1 Написати програму, яка обчислює периметр трикутника.
 - 2.2 Написати програму, яка обчислює периметр прямокутника.
 - 2.3 Написати програму, яка обчислює периметр квадрата.
 - 2.4 Написати програму, яка обчислює площу прямокутника.
 - 2.5 Написати програму, яка обчислює площу квадрата.

Завдання для розумників

- 3.1 Дано довжину ребра куба a . Написати програму, яка обчислює об'єм куба $V = a^3$ і площу його поверхні $S = 6 \cdot a^2$.
- 3.2 Дано довжини ребр a , b , c прямокутного паралелепіпеда. Написати програму, яка обчислює його об'єм $V = a \cdot b \cdot c$ і площу поверхні $S = 2 \cdot (a \cdot b + b \cdot c + a \cdot c)$.
- 3.3 Дано два числа a і b . Написати програму, яка обчислює їх середнє арифметичне: $(a + b) / 2$.

§ 11. Графічні можливості модуля Turtle

Основні алгоритмічні конструкції, зокрема цикли і багато іншого варто вивчати, використовуючи візуалізацію виконання програм. Такі можливості вперше були реалізовані в мові програмування LOGO. Це мова управління візком ("черепашкою"), яка має хвіст, що може залишати за собою слід. Python має всі функції такої "черепашкової" графіки, які зібрані в окремий модуль **turtle**.

Розглянемо, що таке модуль.

Модуль - це самостійна програма, що розширює функціонал вже існуючої програми.

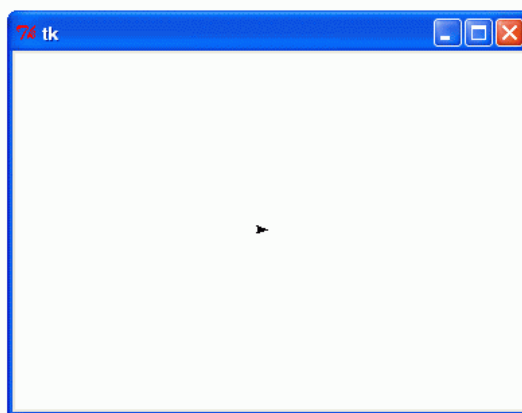
Функціонал - це перелік можливостей, що здатна виконувати програма.

Підключення модуля turtle у Python відбувається за допомогою команди:

```
from turtle import *
```

Ця стрічка дає можливість працювати з графічними елементами.

Тепер ми можемо створювати різні картинки. Після запуску такого роду програм відкривається окреме вікно (з назвою **tk**) з вказівником, що має вигляд трикутника. Будемо називати його «черепашкою» (Мал.1).

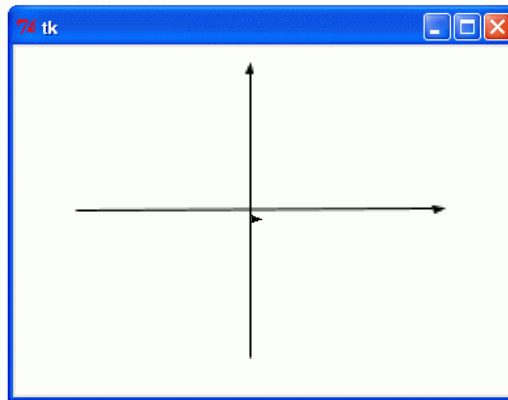


Мал. 1

Черепашка дивиться строго **праворуч** і готова виконувати команди. Малює вона ніби пером. За замовчуванням на початку виконання перо в неї опущене і при пересуванні буде залишати **чорний** слід.

Черепашка орієнтується у вікні як у системі координат. На початку виконання програми вона знаходиться у точці (0, 0) і готова перейти у будь-яку точку з заданими координатами. Її система координат нічим не відрізняється від звичної для

нас та від тієї, яка використовувалася в середовищі Scratch. Ця система координат невидима (Мал.2).



Мал. 2

Для керування «черепашкою» використовують певні команди.

1. **forward(60)** - проповзти вперед 60 кроків (пікселів).
2. **left(50)** - повернути наліво на 50 градусів.
3. **right(90)** - повернути направо на 90 градусів.
4. **circle(30)** - намалювати коло радіуса 30, яке повністю знаходиться зліва від положення черепашки.
5. **circle(-30)** - намалювати коло радіуса 30, яке повністю знаходиться справа від положення черепашки.
6. **goto(50, 90)** - переміщення черепашки у координати (50, 90)
7. **down()** - опустити перо. Після цієї команди черепашка почне залишати слід при будь-якому своєму пересуванні.
8. **up()** - підняти перо. Після цієї команди при пересуванні черепашка не буде залишати слід.
9. **width(3)** - встановити товщину сліду (пера) черепашки 3 пікселів.
10. **color('red')** - встановити червоний колір пера (сліду) черепашки. Колір має бути текстовим рядком з назвою кольору англійською мовою, наприклад, 'red', 'yellow', 'green', 'blue', 'brown'.
11. **fill()** - використовується для малювання зафарбованих областей. Починаючи
12. **begin_fill()** – початок заливки,
13. **end_fill()** – кінець заливки.
14. **shape('turtle')** – показати “черепашку”.

Тестові завдання для перевірки знань

1. В який бік спрямована черепашка перед виконанням команд?
 - a) Праворуч;
 - b) Ліворуч;
 - c) Вгору;
 - d) Вниз.
2. Як називається графічний модуль?
 - a) Turtle;
 - b) Python;
 - c) Paint;
 - d) Forward.
3. Обрати команду, яка задає товщину лінії 5 пікселів.
 - a) forward(5);
 - b) width(5);
 - c) left(5);
 - d) right(5).
4. Обери правильний запис команди:
 - a) goto(50, 90);
 - b) goto(50);
 - c) width(-3);
 - d) circle(red).
5. Обрати команду, яка дозволяє черепашці повернути праворуч на 30.
 - a) forward(30);
 - b) left(30);
 - c) right(30);
 - d) circle(30).
6. Обрати команду, яка дозволяє черепашці при пересуванні не залишати слід.
 - a) down();
 - b) up ();
 - c) width();
 - d) fill().
7. Який слід залишає черепашка за замовчуванням?
 - a) білий;
 - b) чорний;
 - c) невидимий;
 - d) прозорий.
8. В записі **from turtle import** пропущено знак. Який?
 - a) ?
 - b) *
 - c) :
 - d) .

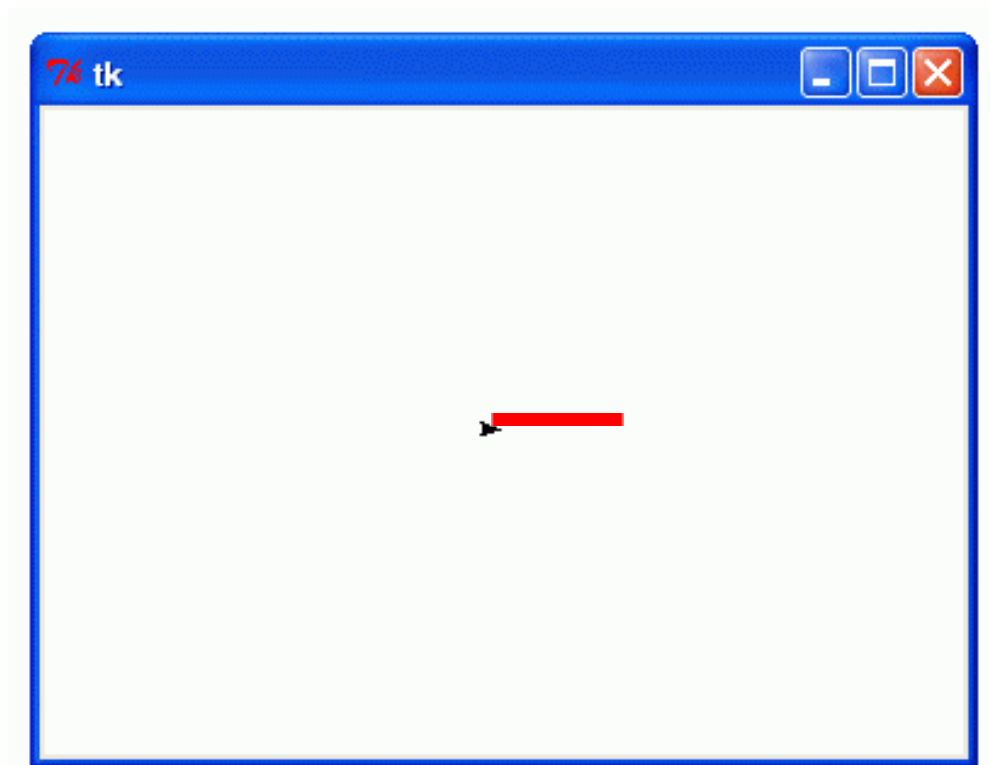
§ 12. Створення найпростіших програм

Проект 1.

Створимо програму, яка малює червону лінію довжиною 100 пікселів.

1. `from turtle import*` підключення графічного модуля
2. `width(10)` товщина лінії 10 пікселів
3. `color('red')` колір лінії червоний
4. `forward(100)` довжина лінії 100 пікселів

Після запуску програми на виконання отримаємо таке зображення (Мал.3):



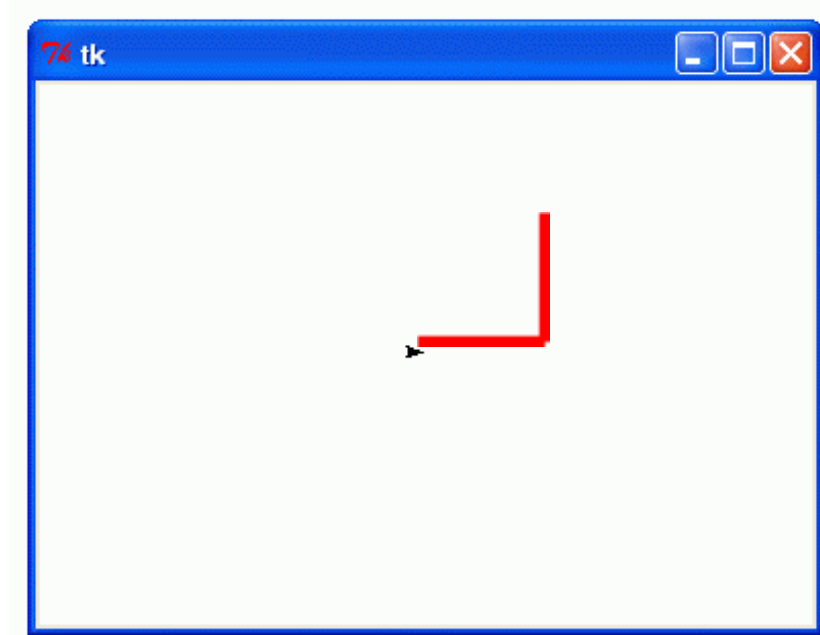
Мал.3

Проект 2.

Ускладнимо програму. Додамо елементи так, щоб утворився прямий кут.

1. `from turtle import*` підключення графічного модуля
2. `width(10)` товщина лінії 10 пікселів
3. `color('red')` колір лінії червоний
4. `forward(100)` довжина лінії 100 пікселів
5. `left(90)` поворот ліворуч на 90 градусів
6. `forward(100)` довжина лінії 100 пікселів

Після запуску програми на виконання отримаємо таке зображення (Мал.4):



Мал.4

Щоб малюнок був зафарбований, використовують групу команд **fill()**

begin_fill() – початок заливки,

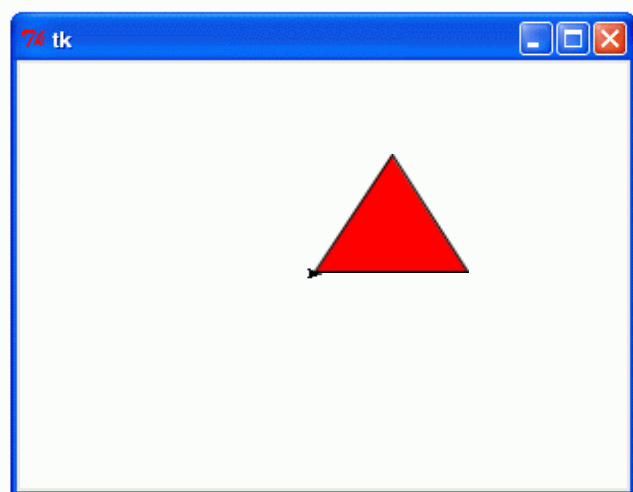
end_fill() – кінець заливки.

Перед початком запису команд вводиться команда **begin_fill()**, далі записуємо команди, необхідні для створення малюнка. І останньою записуємо команду **end_fill()**.

Проект 3.

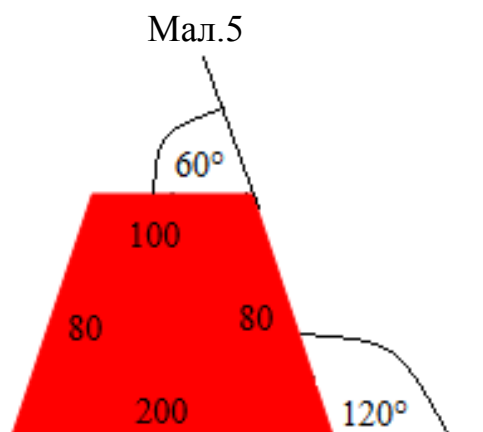
Ускладнимо програму. Додамо елементи так, щоб утворився прямий кут.

1. *from turtle import**
2. *width(10)*
3. *color('red')*
4. *begin_fill()*
5. *forward(100)*
6. *left (120)*
7. *forward(100)*
8. *left (120)*
9. *forward(100)*
10. *end_fill()*



Завдання для самостійного виконання

- 1.1 Написати програму, яка малює лінію синього кольору довжиною 200 пікселів і товщиною лінії 15 пікселів.
- 1.2 Написати програму, яка малює кут 60 градусів жовтого кольору довжиною лінії 100 пікселів і товщиною лінії 10 пікселів.
- 1.3 Написати програму, яка малює квадрат рожевого кольору зі стороною 150 пікселів і товщиною лінії 10 пікселів.
- 1.4 Написати програму, яка малює прямокутник зеленого кольору зі сторонами 150 і 80 пікселів і товщиною лінії 10 пікселів.
- 2.1 Написати програму, яка малює квадрат зі стороною 200 пікселів і товщиною лінії 10 пікселів, кожна сторона якого має інший колір.
- 2.2 Написати програму, яка малює зафарбований квадрат оранжевого кольору зі стороною 100 пікселів і товщиною лінії 5 пікселів.
- 2.3 Написати програму, яка малює трикутник фіолетового кольору зі стороною 150 пікселів і товщиною лінії 10 пікселів.
- 2.4 Написати програму, яка малює ромб рожевого кольору зі стороною 150 пікселів і товщиною лінії 10 пікселів.
- 2.5 Написати програму, яка малює трапецію синього кольору зі сторонами, вказаними на малюнку 5.

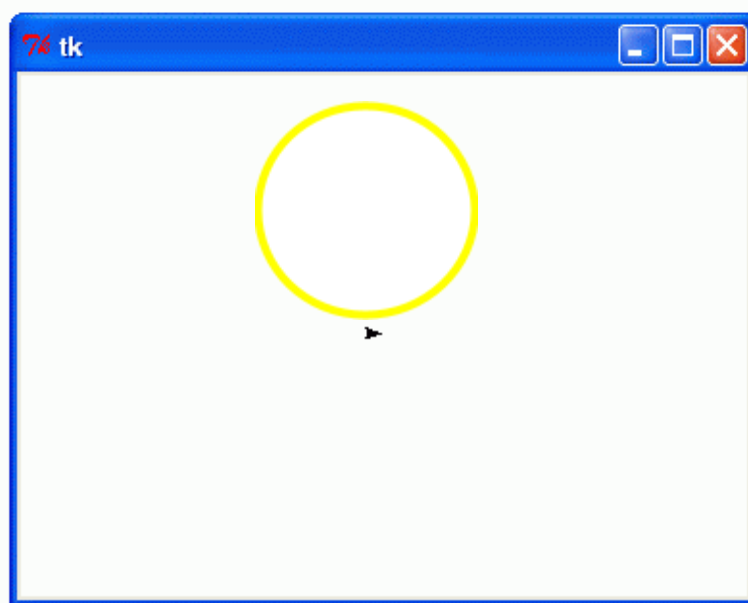


§ 13. Програми, що малюють КОЛО

Напишемо програму, яка малює жовте коло радіуса 100 і товщиною лінії 10

- | | | |
|----|-----------------------------|--------------------------|
| 1. | <i>from turtle import *</i> | підключити модуль turtle |
| 2. | <i>color('yellow')</i> | встановити колір сліду |
| 3. | <i>width(10)</i> | встановити товщину лінії |
| 4. | <i>circle(100)</i> | намалювати коло |

Результат виконання програми (Мал.6):



Мал.6

Залежно від того, куди дивиться черепашка, отримуємо різні положення кола.

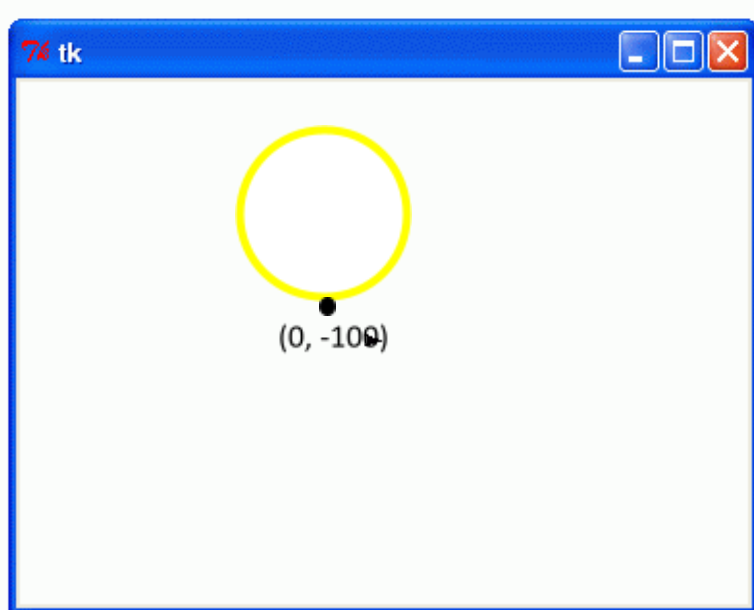
Для малювання кола в різних місцях екрану використовуємо команду *goto(x,y)*.

Зверніть увагу, що спочатку ми переносимо черепашку, а потім малюємо з нової позиції коло, а не навпаки.

- | | | |
|----|-----------------------------|---|
| 1. | <i>from turtle import *</i> | |
| 2. | <i>up()</i> | піднімаємо перо |
| 3. | <i>goto(0,-100)</i> | переміщуємо черепашку в точку з координатами (0,-100) |
| 4. | <i>down()</i> | опускаємо перо |
| 5. | <i>color('yellow')</i> | |
| 6. | <i>width(10)</i> | |

7. *circle(100)*

Результат виконання програми (Мал.7):



Мал.7

Завдання для самостійного виконання

- 1.1 Написати програму, яка малює коло червоного кольору радіусом 50 пікселів.
 - 1.2 Написати програму, яка малює коло червоного кольору радіусом 50 пікселів з точки $(-50, -50)$.
 - 1.3 Написати програму, яка малює два кола однакового кольору та однакових радіусів, що перетинаються.
 - 1.4 Написати програму, яка малює два кола однакового кольору та різних радіусів, що перетинаються.
 - 1.5 Написати програму, яка малює два кола різного кольору та різних радіусів, що перетинаються.
 - 1.6 Написати програму, яка малює два кола однакового кольору та однакових радіусів, що не перетинаються.
 - 1.7 Написати програму, яка малює два кола однакового кольору та різних радіусів, що не перетинаються.
 - 1.8 Написати програму, яка малює два кола різного кольору та різних радіусів, що не перетинаються.
-
- 2.1 Написати програму, що малює квітку, яка складається з 4 кіл одного кольору.
 - 2.2 Написати програму, що малює квітку, яка складається з 6 кіл одного кольору.
 - 2.3 Написати програму, що малює квітку, яка складається з 4 кіл різних кольорів.
 - 2.4 Написати програму, яка малює олімпійські кільця.
 - 2.5 Написати програму, яка малює «Хвіст павича».

Параметри:

1. Коло **фіолетове**, радіус 240.
2. Коло **синє**, радіус 210.
3. Коло **блакитне**, радіус 180.
4. Коло **зелене**, радіус 150.
5. Коло **жовте**, радіус 120.
6. Коло **оранжеве**, радіус 90.
7. Коло **червоне**, радіус 60

