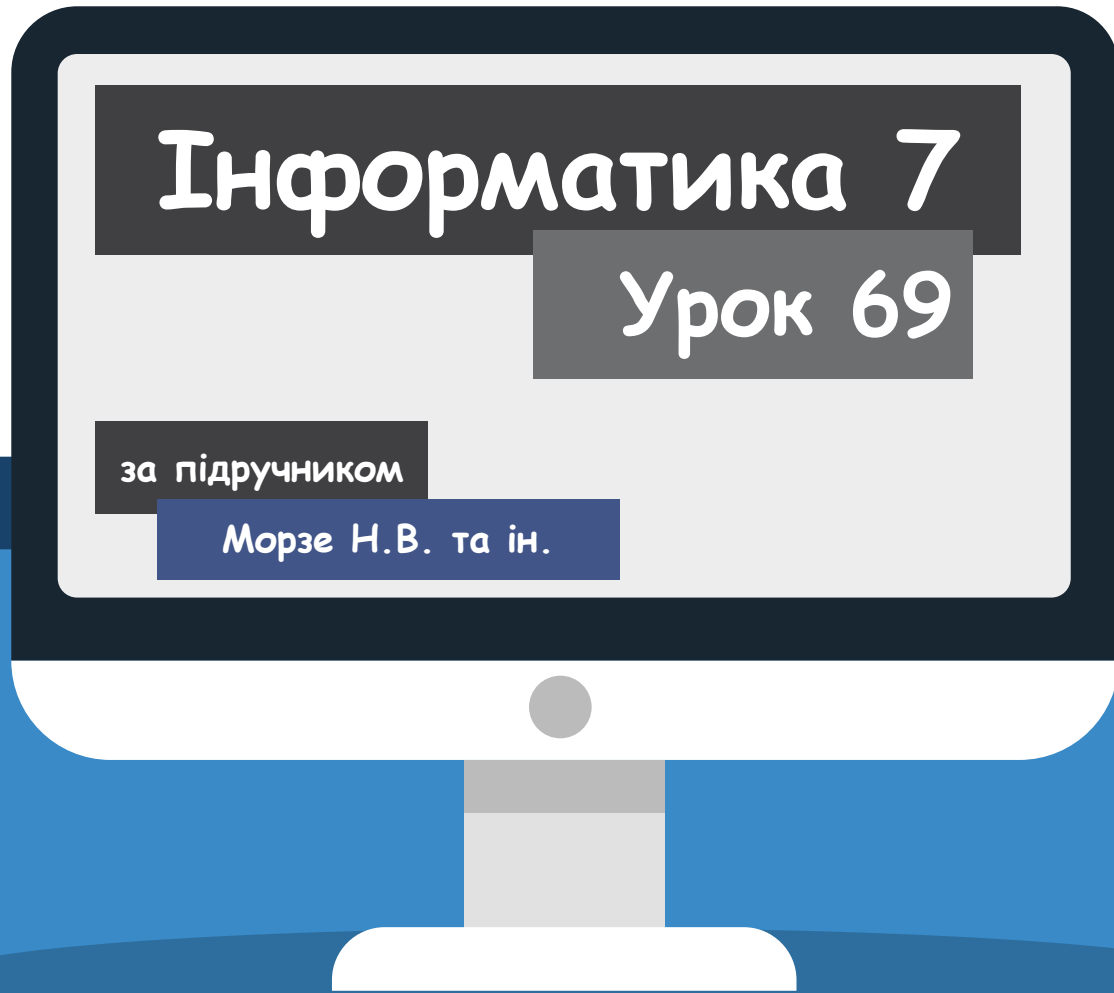


Виконання навчального проєкту



Нова українська школа



*Згадаймо
алгоритм роботи
з програмою у
вікні середовища
Thonny.*



python™

*Введення тексту
програми*

*Збереження
файлу
програми*

*Виконання
програми*

*Перегляд
результату*

Як описати алгоритмічну структуру неповного розгалуження мовою програмування Python?

Для опису алгоритмічної структури **розгалуження** мовою програмування, як і в середовищі **Скретч**, використовують оператор **неповного розгалуження**

```
if <логічний вираз>:
```

```
    блок команд
```

обов'язковий відступ від
лівого краю.

Після запису умови слід поставити двокрапку **:**, яка показує, що далі має бути розташований блок дій.



Блок дій записується з обов'язковим однаковим відступом від лівого краю.

Як описати алгоритмічну структуру неповного розгалуження мовою програмування Python?

Оператор **if** перевіряє істинність зазначеної умови.

Якщо умова набуває значення **True** (Істина)

Якщо ж умова приймає значення **False** (Хибність)

програма виконає блок дій,
зазначених у частині
<Оператор>

то блок **<Оператор>**
пропускається, і керування
передається оператору, що
йде після оператора **if**

```
if <Умова>:  
    <Оператор>  
<Оператор>
```

Як описати алгоритми із декількома умовами мовою програмування Python?

У задачах, які передбачають виконання альтернативних дій, використовують **оператор повного розгалуження**. Для запису оператора повного розгалуження додатково використовують службове слово **else**:


```
if <логічний вираз>:  
    блок команд 1  
else:  
    блок команд 2
```

Як описати алгоритми із декількома умовами мовою програмування Python?

*Якщо є потреба врахувати значення декількох логічних виразів, то використовують **вкладені розгалуження**:*

```
if <логічний вираз1>:  
    блок команд 1  
elif <логічний вираз2>:  
    блок команд 2  
...  
elif <логічний вираз n>:  
    блок команд n  
else:  
    <блок команд інакше>
```

Як описати алгоритми із декількома умовами мовою програмування Python?



*Якщо вкладених умовних операторів декілька, то до якого з них належить **else**, можна зрозуміти за відступом.*

*Відступ у **else** має бути такий само, як в **if**, до якого він належить.*



Як мовою програмування Python описують цикл із передумовою?

У мові програмування **Python** цикл із передумовою описується оператором **while...**

Скорочена форма

Опис

Приклад

```
while <логічний вираз>:  
    <команда>
```

```
while a<5:  
    a+=1
```

Як мовою програмування Python описують цикл із передумовою?

Повна форма

Опис

```
while <логічний вираз>:  
    <команда1>  
    <команда2>  
    ...  
    <команда n>  
else:  
    <команда інакше>
```

Приклад

```
while a<5:  
    s=s+a  
    a+=1  
else:  
    print('Сума не  
змінюється')
```

Як виконується оператор циклу з передумовою?

Скорочена форма оператора **while** працює таким чином. Визначається значення логічного виразу.

Якщо значення логічного виразу **true**

то виконується команда, що слідує після рядка зі знаком «**:**» у рядках, які мають відступ 4 позиції.

Якщо значення логічного виразу **false**

то цикл завершується й управління передається команді, яка записана у програмному коді після тіла циклу (з рядка, який перебуває на тому ж рівні, що і команда **while**).

Як мовою програмування Python описують цикл із визначеною кількістю повторень?

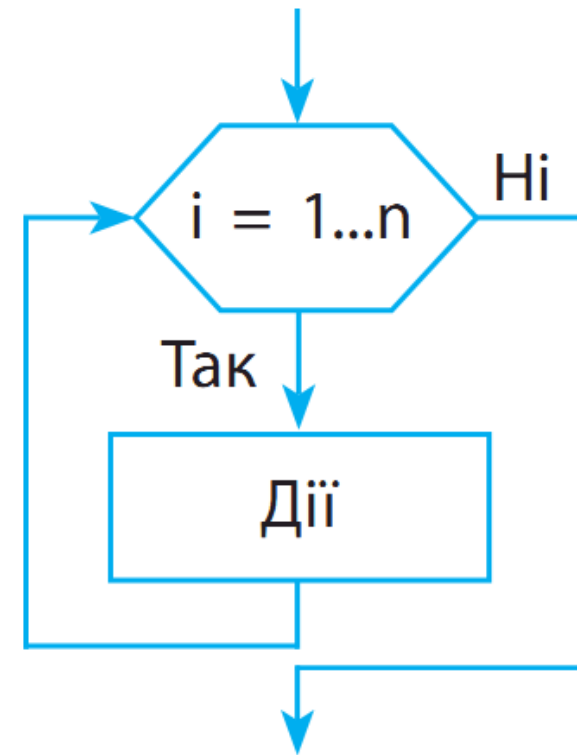
Для того, щоб описати повторення визначену кількість разів, використовують оператор **циклу із параметром**.

У мові програмування **Python** цикл із параметром описується оператором **for...**

Параметр циклу

```
for i in <діапазон>:  
    <команда>
```

Тіло циклу



Як мовою програмування Python описують цикл із визначеною кількістю повторень?

Числові послідовності можна задавати функцією **range()**:

```
range([<початок>[, <кінець>[, <крок>]])
```

Відлік починається зі значення **початок**

Завершується на 1 раніше за значення **кінець**

Із заданим кроком **крок**

Чергове число генерується при кожному звертанні до такого об'єкта, тому навіть для довгих послідовностей не займає багато пам'яті.

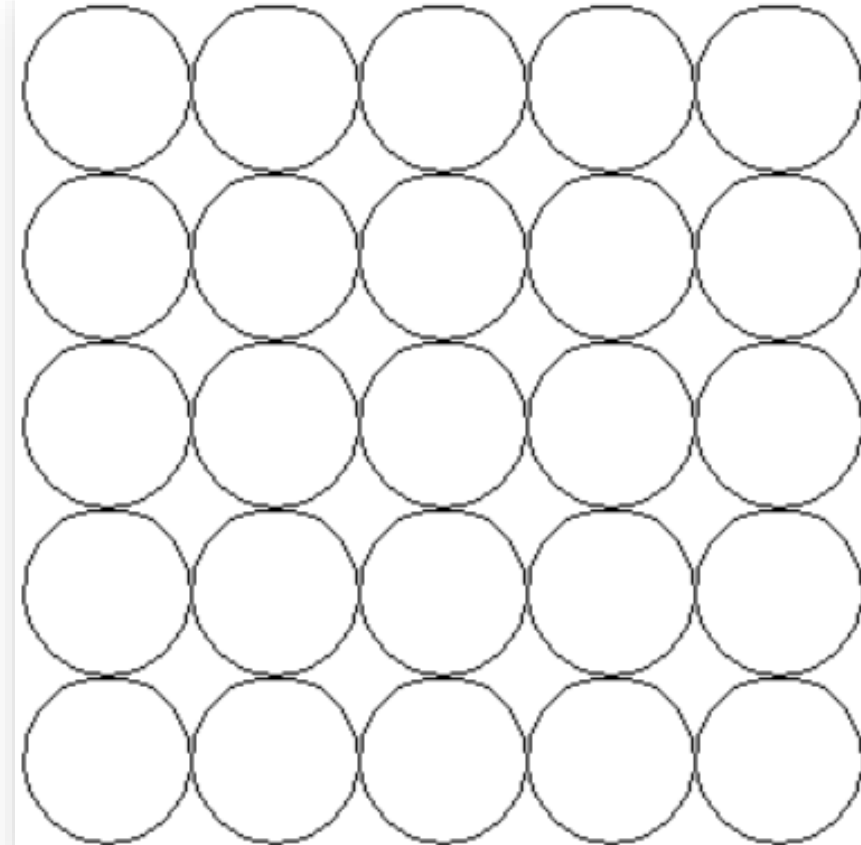
Завдання 1. Створи програму розв'язання задачі в середовищі *Python*.

Задача.

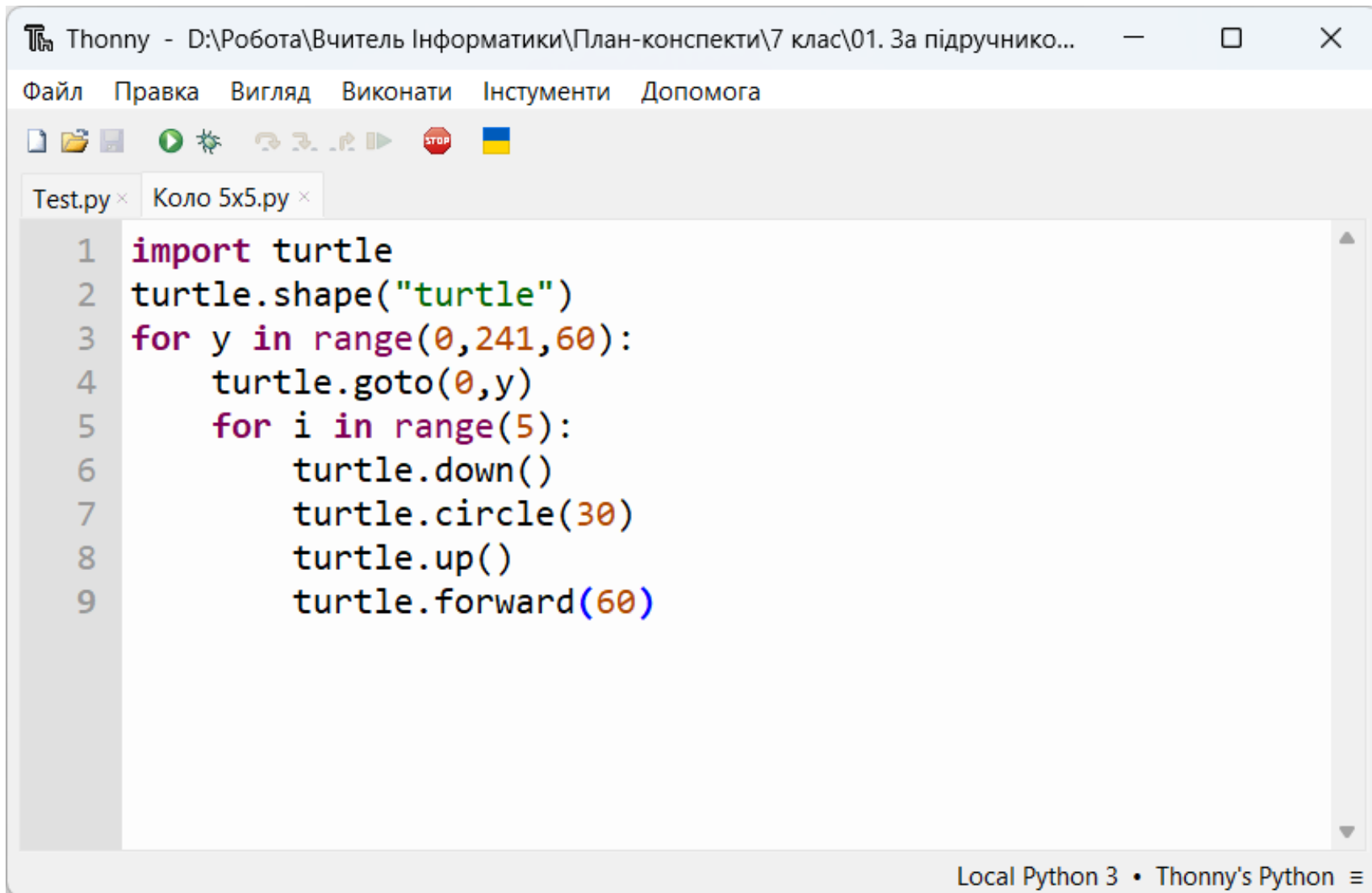
Створи програму побудови зображення згідно зразка.

Радіус кола = 30

Збережіть файл програми з іменем **Коло 5x5.**



**Можливий варіант
розв'язку:**



```
Thonny - D:\Робота\Вчитель Інформатики\План-конспекти\7 клас\01. За підручничко...
Файл  Правка  Вигляд  Виконати  Інструменти  Допомога
Test.py ×  Коло 5x5.py ×
1  import turtle
2  turtle.shape("turtle")
3  for y in range(0,241,60):
4      turtle.goto(0,y)
5      for i in range(5):
6          turtle.down()
7          turtle.circle(30)
8          turtle.up()
9          turtle.forward(60)
Local Python 3 • Thonny's Python
```

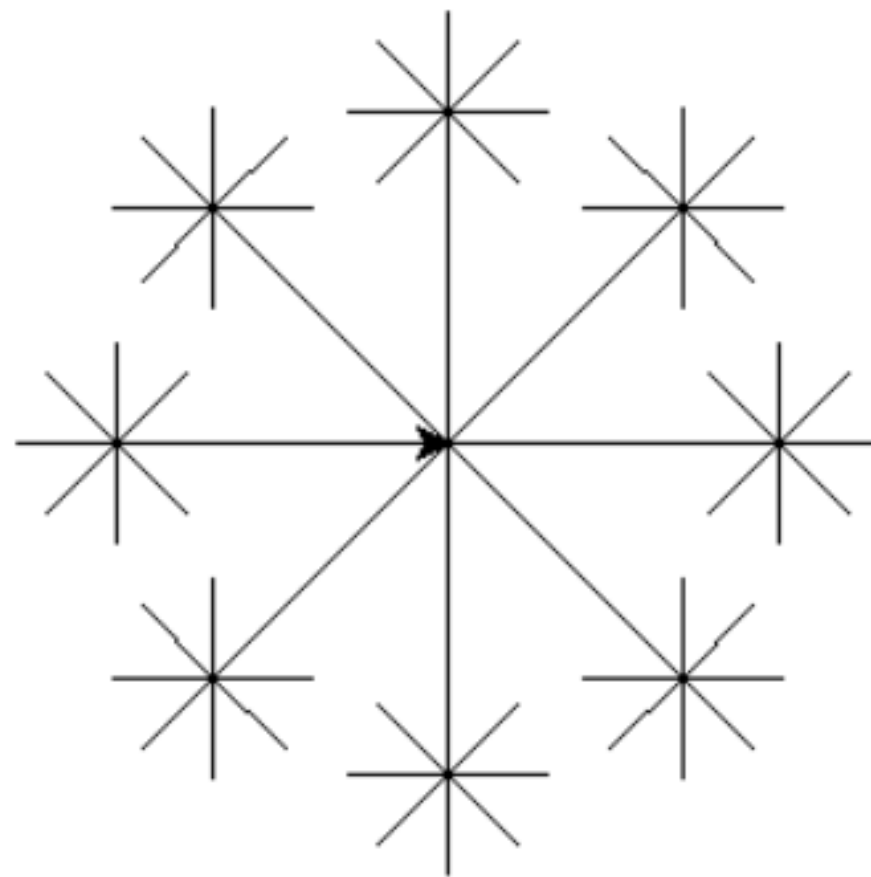
Завдання 2. Створи програму розв'язання задачі в середовищі *Python*.

Задача.

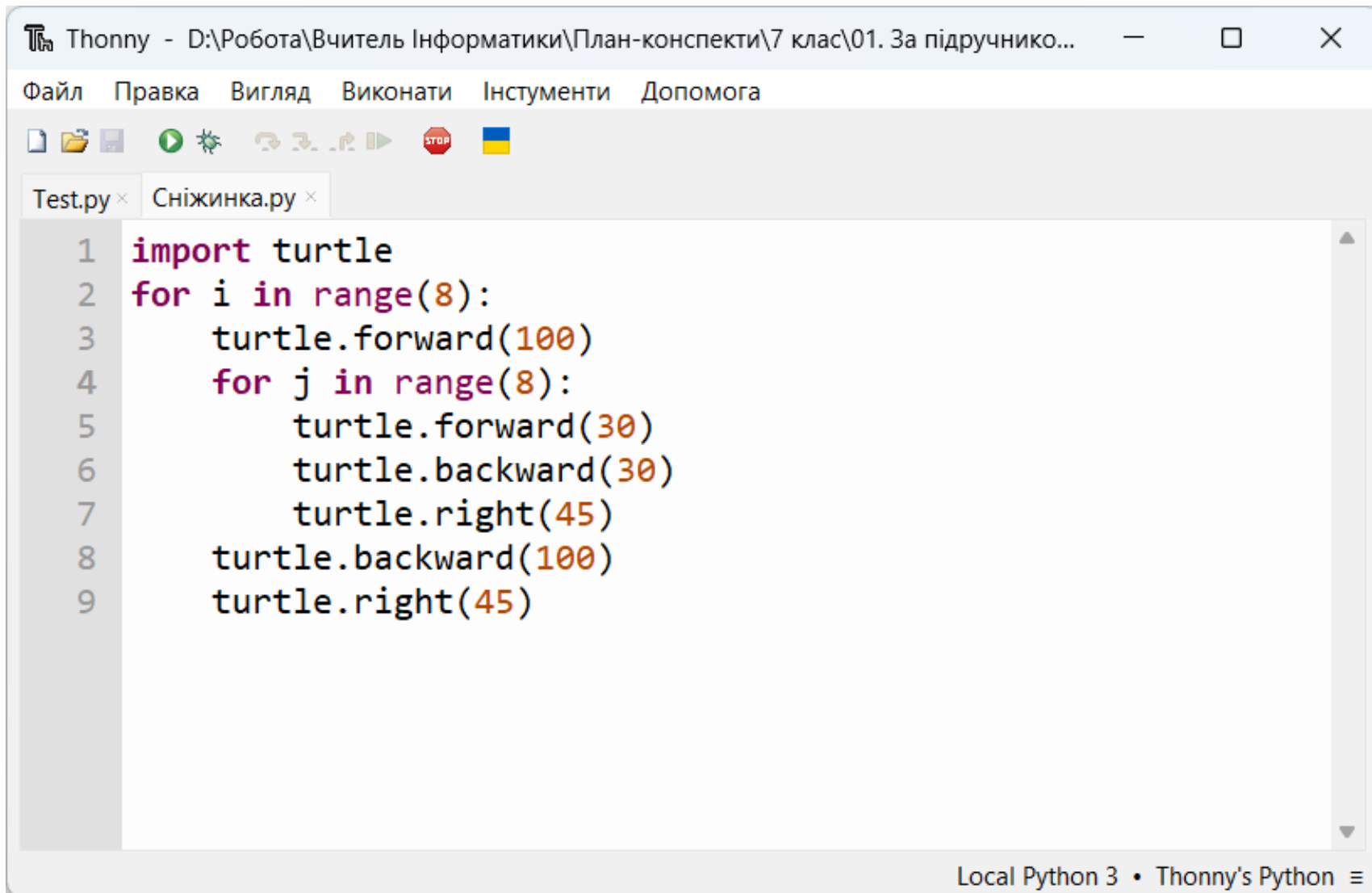
Створи програму побудови зображення сніжинки.

Довжина променю великої сніжинки дорівнює 100, маленької сніжинки - 30.

Збережіть файл програми з іменем **Сніжинка.**



**Можливий варіант
розв'язку:**



```
Thonny - D:\Робота\Вчитель Інформатики\План-конспекти\7 клас\01. За підручничко...
Файл  Правка  Вигляд  Виконати  Інструменти  Допомога
Test.py ×  Сніжинка.py ×
1  import turtle
2  for i in range(8):
3      turtle.forward(100)
4      for j in range(8):
5          turtle.forward(30)
6          turtle.backward(30)
7          turtle.right(45)
8      turtle.backward(100)
9      turtle.right(45)
Local Python 3 • Thonny's Python
```

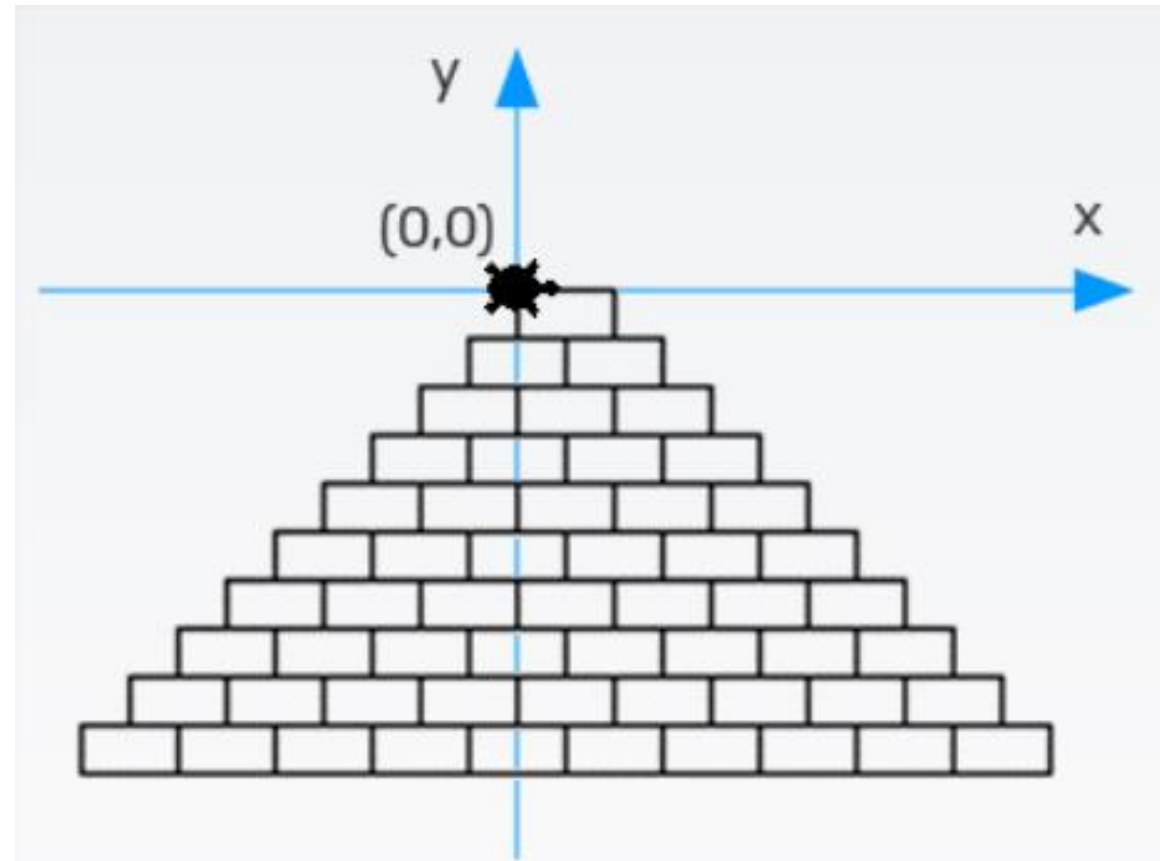
Завдання 3. Створи програму розв'язання задачі в середовищі *Python*.

Задача.


Створи програму побудови піраміди з цеглин.

Розмір цеглинки – 20 x 40

Збережіть файл програми з іменем **Пірамідка**.



**Можливий варіант
розв'язку:**



The screenshot shows the Thonny Python IDE interface. The window title is "Thonny - D:\Робота\Вчитель Інформатики\План-конспекти\7 клас\01. За підручничко...". The menu bar includes "Файл", "Правка", "Вигляд", "Виконати", "Інструменти", and "Допомога". The toolbar contains icons for file operations, running, and stopping. Two tabs are open: "Test.py" and "Пірамідка.py *". The code in the "Пірамідка.py" tab is as follows:

```
1 import turtle
2 for i in range(1,11):
3     turtle.up()
4     turtle.goto(-20*i,-20*i)
5     turtle.down()
6     for j in range(i):
7         for k in range(2):
8             turtle.forward(40)
9             turtle.right(90)
10            turtle.forward(20)
11            turtle.right(90)
12            turtle.forward(40)
```

The status bar at the bottom right indicates "Local Python 3 • Thonny's Python".

Складання програм

Побудуємо формульну модель для задачі «Вклади з капіталізацією».

Припустимо, ти поклав 1000 грн в банк на депозит на 1 рік під 10% річних. «Депозит» означає, що в кінці року ти зможеш забрати в банку всі вкладені тобою гроші плюс обумовлений відсоток від вкладеної суми.



Оскільки 10% від 1000 грн становить 100 грн, то, вклавши 1000 грн під 10%, через рік ти отримаєш 1100 грн.

Складання програм

*А якщо через рік гроші не забрати,
то скільки ти отримаєш через **2 роки**?*

*«Капіталізація відсотків» означає, що
тепер відсотки нараховуватимуться
не на початкову суму вкладу, а на ту,
яку отримано наприкінці першого
року.*



*Тобто через **2 роки** ти отримаєш:*

$$1100 + 1100 * 10\% = 1100 + 110 = 1210 \text{ грн}$$

Вправа 4. Створи комп'ютерну модель розв'язання задачі «Вклади з капіталізацією відсотків» у середовищі Python.

Задача.

Для введеної початкової суми вкладу та річного відсотку програма має виводити накопичену суму вкладу протягом 10 років.

Збережіть файл програми з іменем **Депозит**.



Складання програм

Ось позначення і формули, необхідні для розв'язання задачі.

Програма "**Депозит**"

S

Початкова сума

n

Річний відсоток
 $S=S*(1+n/100)$



**Можливий варіант
розв'язку:**



The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - C:\Users\ProMa\Desktop\Test.py @ 5:12". The menu bar includes "Файл", "Правка", "Вигляд", "Виконати", "Інструменти", and "Допомога". The toolbar contains icons for file operations, running, and debugging. The main editor window shows a Python script named "Test.py" with the following code:

```
1 S=int(input('Початкова сума '))
2 n=int(input('Річний відсоток '))
3 for y in range(1,11):
4     S=S*(1+n/100)
5     print(y,'рік - ',round(S),'грн')
```

The status bar at the bottom right indicates "Local Python 3 • Thonny's Python".

Складання програм

Тепер припустимо, що умови депозиту змінилися.

Клієнт банку одержав дві новини:

1) гарну - кожні три роки банк підвищує відсоток на 0,5%;

2) погану - кожного року необхідно сплачувати податок на одержаний прибуток, тобто прибуток скорочується на певну кількість відсотків (яку вводить користувач).

Необхідно визначити суму, яку одержить клієнт банку на кінець кожного року.

Складання програм

*Додай до створеної у **Python** моделі код підвищення відсотку кожні три роки на 0.5%.*

Код перевірки, чи рік кратний 3, є таким:

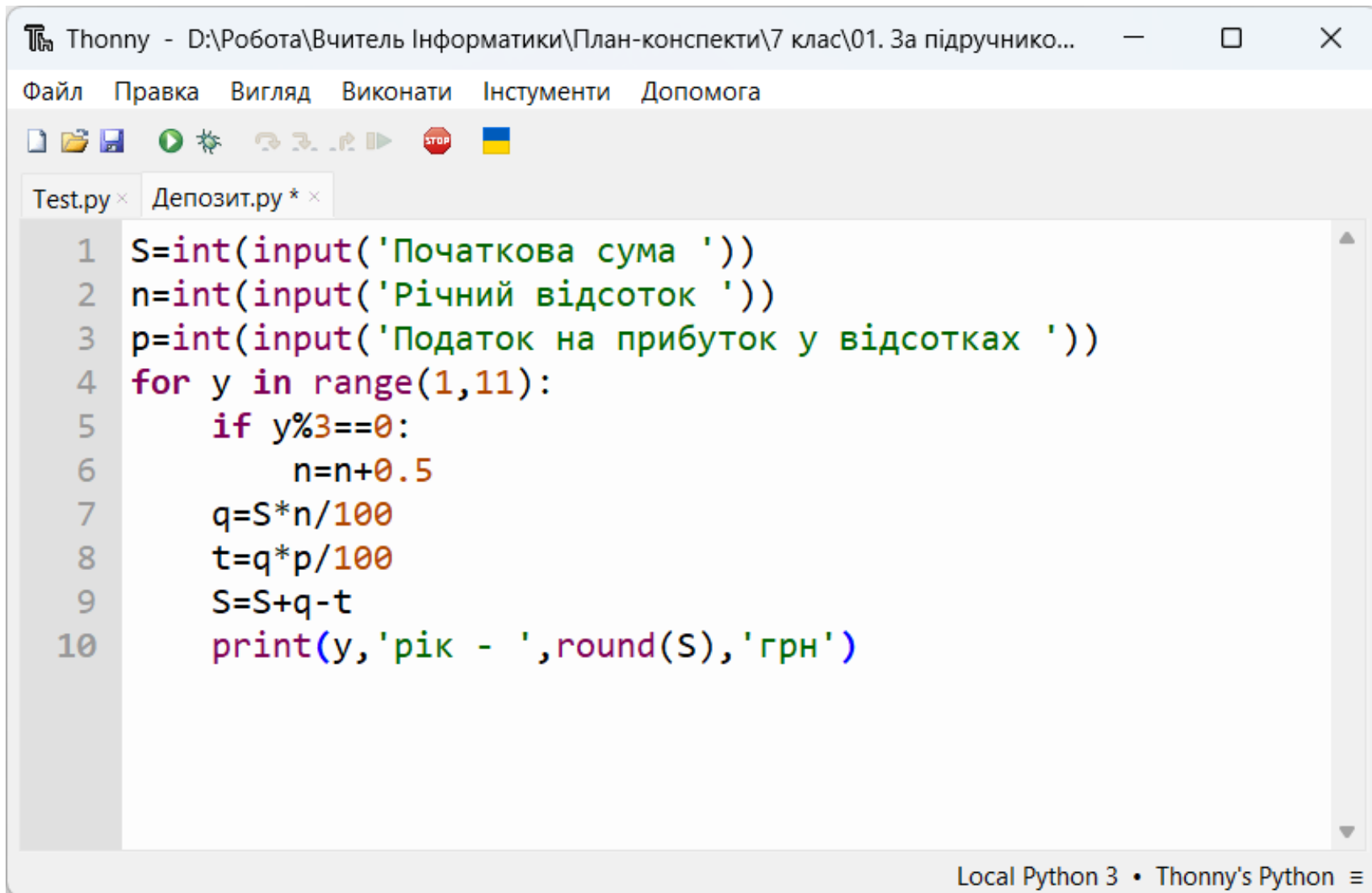
```
if y%3==0:
```

Додай формули обчислення прибутку, податку та зміни формулу обчислення суми.

p

Податок на прибуток у відсотках

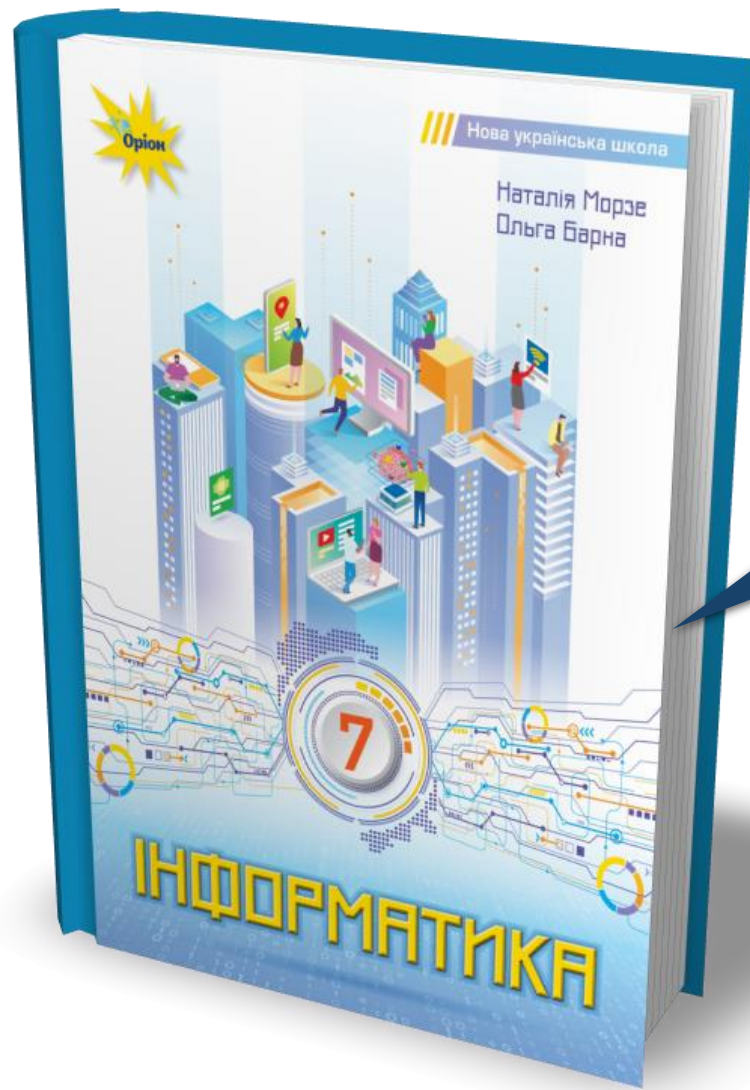
**Можливий варіант
розв'язку:**



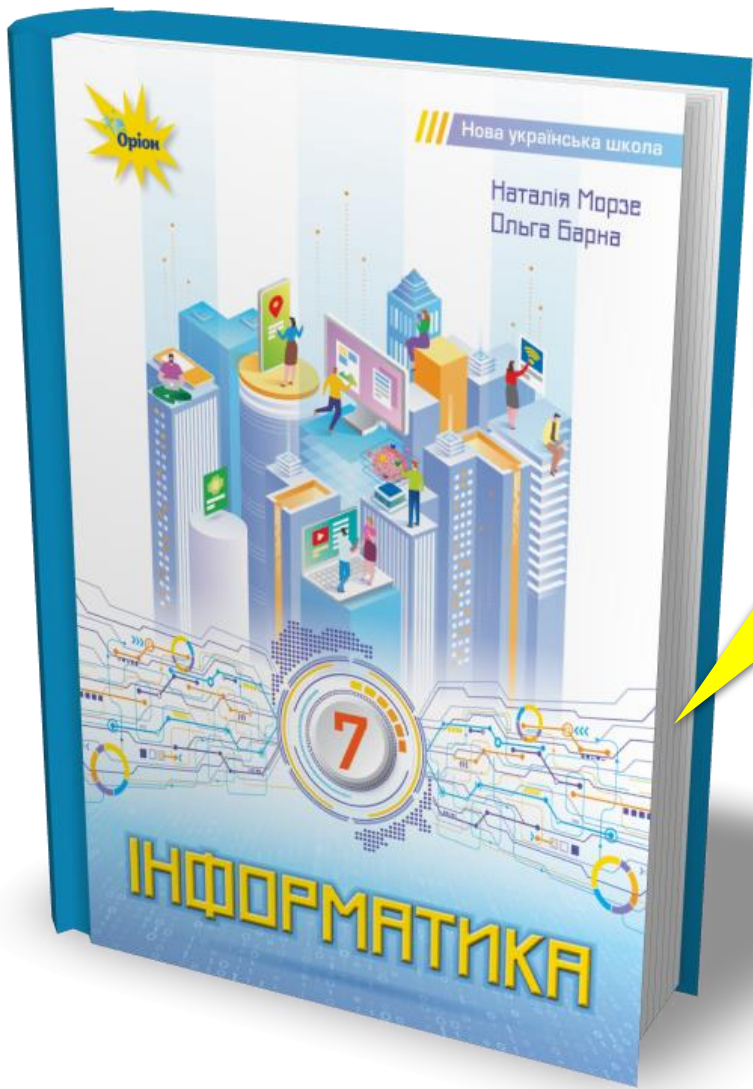
```
Thonny - D:\Робота\Вчитель Інформатики\План-конспекти\7 клас\01. За підручничко...
Файл  Правка  Вигляд  Виконати  Інструменти  Допомога
Test.py ×  Депозит.py * ×
1  S=int(input('Початкова сума '))
2  n=int(input('Річний відсоток '))
3  p=int(input('Податок на прибуток у відсотках '))
4  for y in range(1,11):
5      if y%3==0:
6          n=n+0.5
7      q=S*n/100
8      t=q*p/100
9      S=S+q-t
10  print(y,'рік - ',round(S),'грн')
```

Local Python 3 • Thonny's Python

Домашнє завдання



Проаналізувати
с. 252-254



**Сторінка
252-254**



Інформатика 7

Урок 69

за підручником

Морзе Н.В. та ін.

Дякую за увагу!

Нова українська школа

