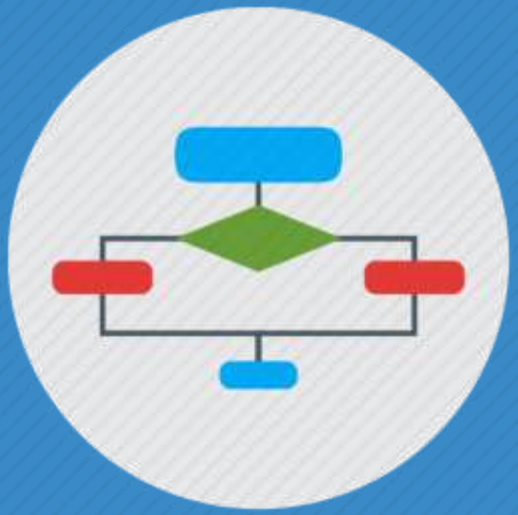
6 ІНФОРМАТИКА

Вкладені алгоритмічні структури розгалуження

За навчальною програмою 2017 року



Урок 25

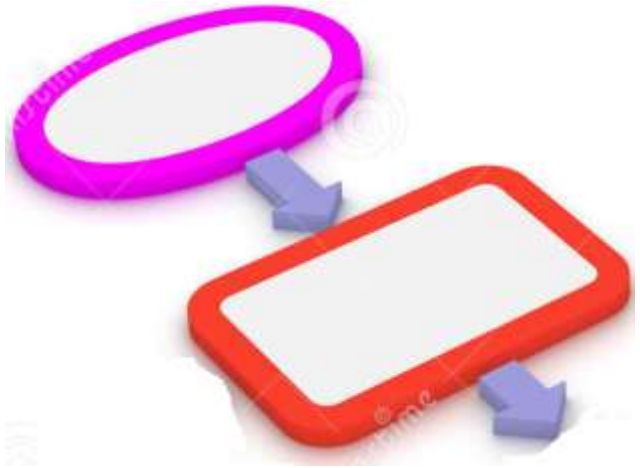
Розділ 3
§ 20

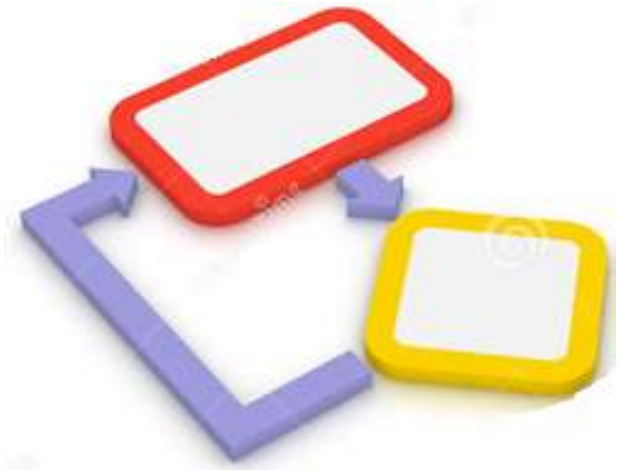
Повторюємо

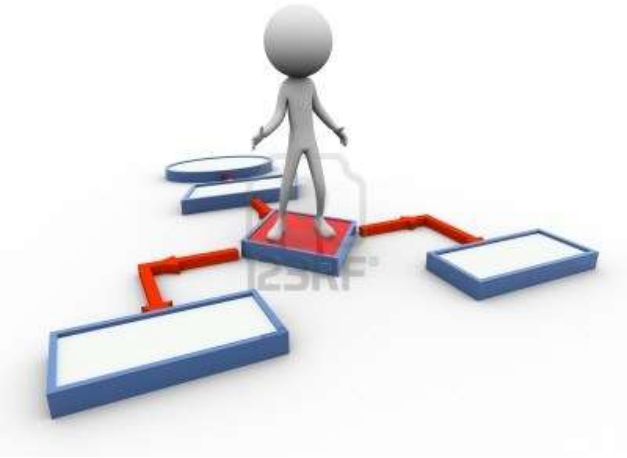
6

Ви знаєте, що під час конструювання алгоритмів використовуються три базові алгоритмічні структури:

Слідування	Повторення	Розгалуження







У 5 класі ви навчилися складати програми мовою Python з використанням операторів розгалуження та операторів повторення.

Розділ 3
§ 20

Повторюємо

Ви дізналися, що оператор розгалуження можна використовувати в:

повній формі ***і*** ***неповній формі***

А для опису дій, які мають виконуватися кілька разів, можна використовувати оператори циклу двох видів:

із параметром ***з передумовою***

Розділ 3
§ 20

Запитання

6

1. *Що таке умова в операторах розгалуження та повторення?*
2. *Яке значення мають відступи команди від лівого краю вікна програми?*
3. *Поясніть структуру й правила виконання циклу з параметром; циклу з передумовою.*
4. *У чому полягає відмінність у використанні циклу з параметром і циклу з умовою?*

Розділ 3
§ 20

Оператори розгалуження

Для реалізації розгалуження є умовні оператори if і if...else.

Згадаємо синтаксис оператора розгалуження:

Неповна форма розгалуження	***Повна форма розгалуження***
if <умова> : ***___<оператор***	***if <умова> :*** ***___<оператор 1>*** ***else:*** ***___<оператор 2>***

___ — обов'язковий відступ від лівого краю.

Розділ 3
§ 20

Оператори розгалуження

Умова *— це логічний вираз, значенням якого є True (Істина) або False (Хибність).*

True (Істина)

або

False (Хибність)

Розділ 3
§ 20

Оператори розгалуження

Перевіримо, чи є число x додатним.

Неповна форма розгалуження	Повна форма розгалуження

`if x > 0:`
`  print ('Число додатне')`
	`if x > 0:`
	`  print ('Число додатне')`
	`else:`
	`  print ('Число не додатне')`

Якщо умова хибна (x 0), то керування передається оператору, наступному за **if**	Якщо умова хибна (x 0), то виконується гілка **Hi** і виводиться повідомлення 'Число не додатне'

Розділ 3
§ 20

Оператори розгалуження

Проста умова утворюється за допомогою операцій відношення, складена умова — з кількох простих умов за допомогою логічних операцій:

AND	OR	NOT
(логічне І)	(логічне АБО)	(логічне заперечення)

Визначимо значення складеної умови
x >= 10 and x <= 18.

Значення x	5	20	10	15
Значення складеної умови	Хибне	Хибне	Істинне	Істинне

Розділ 3
§ 20

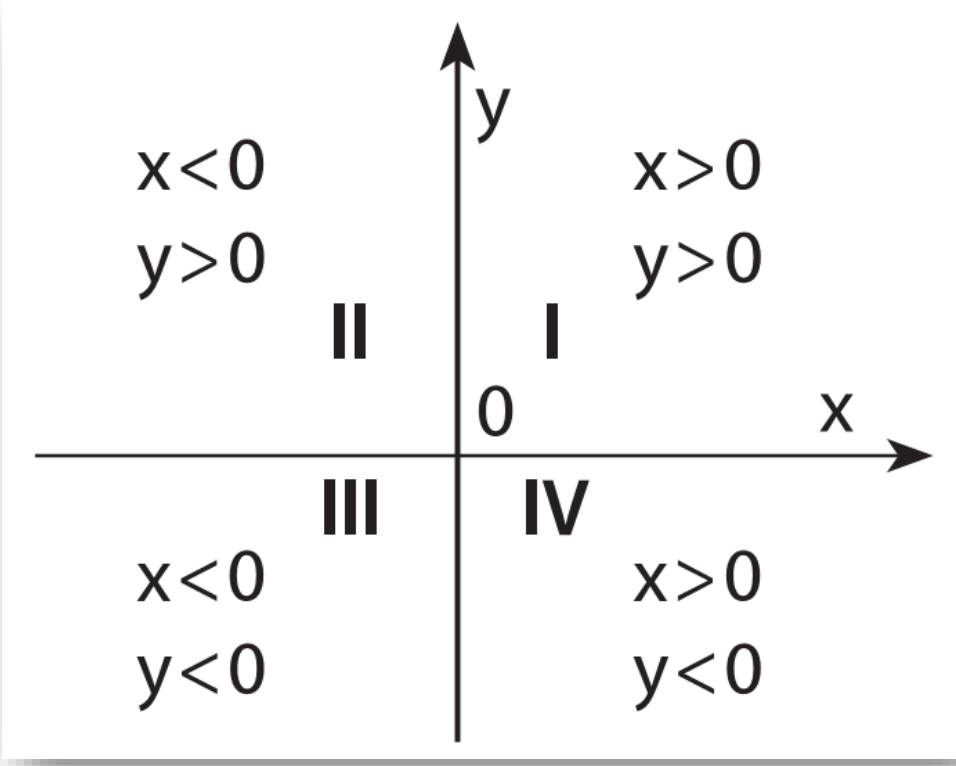
Вкладені розгалуження

6

Якщо після перевірки деякої умови виникає потреба знову робити вибір, застосовують вкладені розгалуження: в умовному операторі if по гілці Так або гілці Ні знову використовують оператор if.

Розглянемо задачу.

Визначити N — номер координатної чверті, в якій міститься точка з координатами x, y (x ≠ 0, y ≠ 0).



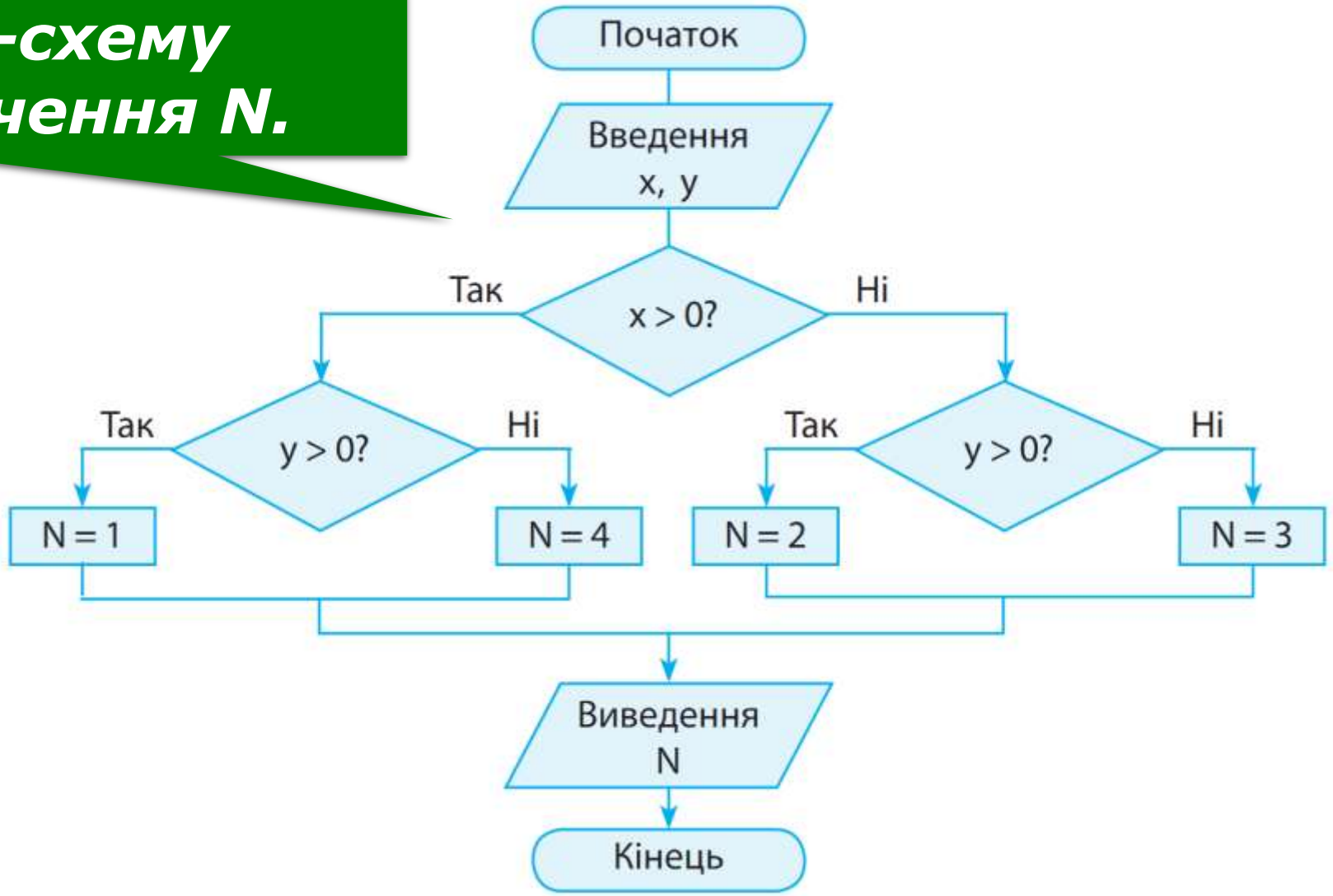
Розділ 3
§ 20

Вкладені розгалуження

6

Складемо блок-схему алгоритму визначення N.

Після визначення знака x знову з'являється потреба робити вибір залежно від знака y, тобто два наступних розгалуження «вкладаються» в перше.



Розділ 3
§ 20

Вкладені розгалуження

За цим алгоритмом запишемо програмний код:

```
x = int(input('x = ?'))
y = int(input('y = ?'))
if x>0:
    if y>0: n = 1
    else: n = 4
else:
    if y>0: n = 2
    else: n = 3
print('N =', n)
```

Розділ 3
§ 20

Вкладені розгалуження

Якщо x = 5, y = –2, то вираз x > 0 набуває значення True. По гілці Так перевіряється умова y > 0, яка при y = –2 набуває значення False, тому змінна n набуває значення 4.

Команди, вкладені в гілки оператора if, об'єднуються в блоки за величиною відступів. Відступ може бути будь-яким, головне, щоб у межах одного вкладеного блоку він був однаковий.

Розділ 3
§ 20

Множинне розгалуження

Якщо залежно від значення тієї чи іншої змінної може виконуватися одна з трьох (або більше) гілок програми, код стає громіздким і вкладати оператори if незручно.

Для таких випадків у Python є структура множинного розгалуження, яка реалізується оператором (скорочення від else if — «ще якщо»).

elif

У гілці elif обов'язково повинен бути логічний вираз, як у заголовку if. У кінці, після всіх elif, може використовуватися одна гілка else для обробки випадків, які не відповідають умовам гілки if і всіх elif.

Розділ 3
§ 20

Множинне розгалуження

Запрограмуємо поведінку гравця в лабіринті, якщо x — кількість монстрів, які зустрічаються на шляху.

```
x = int(input('x = ?'))
if x == 0:
    print('Монстрів немає. Шлях вільний!')
elif x<3:
    print('Стільки монстрів я легко подолаю’)
elif x<5:
    print('Доведеться позмагатися')
else:
    print('Час рятуватися втечею')
```

Розділ 3
§ 20

Множинне розгалуження

Інструкція if-elif-else припиняє перегляд наступних гілок, як тільки логічний вираз у поточній гілці буде True. Так, якщо вираз при if (перша гілка) буде True, то після виведення рядка 'Монстрів немає. Шлях вільний!' виконання програми завершиться.

Таким чином, використання вкладених розгалужень допоможе запрограмувати вибір із великої кількості варіантів.

Розділ 3
§ 20

Питання для самоперевірки

6

1. *Як записується і виконується умовний оператор у повній формі; неповній формі?*

2. *Поясніть схему виконання оператора if, у якому застосовані вкладені оператори розгалуження.*

3. *Напишіть програму, яка за кількістю правильних відповідей K визначає оцінку, яку учень отримав за виконання тесту, за правилом: якщо K > 9, то оцінка «Відмінно»; якщо 8 K 9 — оцінка «Добре»; якщо 6 K 7 — оцінка «Задовільно»; якщо K < 6 — повідомлення «Слід повторити роботу»).*

Розділ 3
§ 20

Домашнє завдання

Проаналізувати
§ 20, ст. 128-132

Розділ 3
§ 20

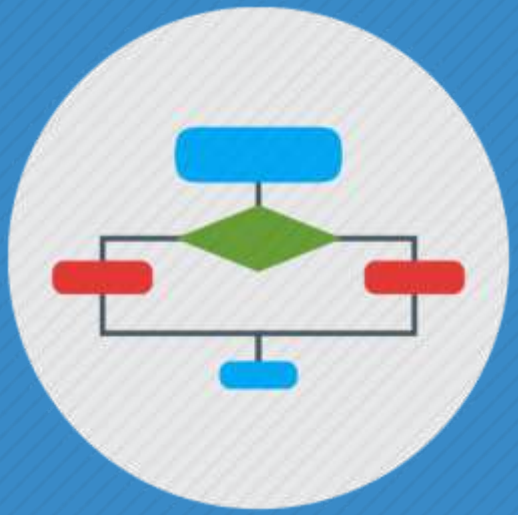
Працюємо за комп’ютером

Сторінка 132

6 ІНФОРМАТИКА

Дякую за увагу!

За навчальною програмою 2017 року



Урок 25