Циклічні конструкції

№ уроку: 4 **Курс:** Python Starter

Засоби навчання: PyCharm

Огляд, мета та призначення уроку

Після завершення уроку учні матимуть уявлення про циклічні конструкції, можливості застосування циклічних конструкцій для оптимізації коду програми та практичного застосування для вирішення задач на практиці.

Вивчивши матеріал даного заняття, учень зможе:

- Складати циклічні алгоритми
- Користуватися операторами while та for, break та continue

Зміст уроку

1. Поняття циклічної конструкції
2. Види циклічних конструкцій
3. Цикл while
4. Ключові слова break і continue
5. Цикл while з гілкою else
6. Цикл for
7. Цикл for з гілкою else
8. Вкладені цикли

Резюме

Циклічні конструкції

Види найпростіших алгоритмів:

- **Лінійний алгоритм** – набір команд (вказівок), що виконуються послідовно в часі одна за одною.
- **Розгалужуваний алгоритм** – алгоритм, що містить хоча б одну умову, в результаті перевірки якої може здійснюватися поділ на кілька паралельних гілок алгоритму.
- **Циклічний алгоритм** — алгоритм, що передбачає багаторазове повторення однієї й тієї ж дії (одних і тих самих операцій) над новими вихідними даними. До циклічних алгоритмів зводиться більшість методів обчислень, перебору варіантів. Цикл програми – послідовність команд (серія, тіло циклу), яка може виконуватися багаторазово (для нових вихідних даних) до задоволення певної умови.
- **Цикл** – це керуюча конструкція, що багаторазово виконує блок коду.

Існують три класичні різновиди циклів:

- цикл із передумовою;

- цикл з постумовою;

- цикл із лічильником.

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Також для сучасних мов програмування характерний foreach-цикл (спільний цикл), який виконує певні дії для всіх елементів з набору.
- Перші два варіанти циклічних конструкцій виконують ділянку коду, поки зазначена умова є істинною. Вони відрізняються тим, коли перевіряється умова: на початку кожного кроку (ітерації) чи наприкінці. Цикли з передумовою та з постумовою взаємозамінні та будь-який з них можна переписати з використанням іншого, при цьому (на думку творців мови Python) цикли з постумовою використовуються значно рідше та гірше читаються через місце вказівки умови, тому в Python існують лише цикли з передумовою (Цикли "while").
- Цикл з лічильником (цикл “for”) використовується для повторення певних дій конкретну кількість разів, при цьому доступний лічильник – номер поточної ітерації.
- У Python цикл “for” є за своєю суттю циклом спільним циклом, тобто цикл із лічильником реалізується за допомогою foreach-циклу та функції, яка повертає послідовність цілих чисел. Як насправді працює цикл “for” у Python ми вивчимо пізніше (загалом розглянемо у сьомому уроці даного курсу та ґрунтовно вивчимо в курсі Python Essential при вивченні колекцій та послідовностей). Поки що ми розглядатимемо цикл “for” як цикл із лічильником.
- **Цикл while** у Python – цикл із передумовою. Синтаксис:

  while умова:
      оператори

- Ця конструкція схожа на оператор if, який був вивчений у попередньому уроці, тільки виконує блок операторів не один раз, а кілька, поки умова істинна. Все, що було сказано про умови, відступи та вкладені оператори справедливо і для оператора while (а також for, який буде розглянутий у цьому уроці пізніше).
- Цикл while не виконається жодного разу, якщо умова спочатку була хибною.
- Умова повинна залежати від якого-небудь стану, котре змінюється, інакше цикл виконуватиметься нескінченно (поки користувач не перерве виконання програми), якщо вона спочатку була істинною.

- Приклад циклу while:

```
x = 0
while x <= 0:
   x = int(input("Введіть додатнє число: "))
```

- Якщо необхідно створити нескінченний цикл, у якості умови *прийнято використовувати True:

```
print('Всі натуральні числа:')
n = 1

while True:
   print(n)
   n += 1
```

- Якщо необхідно достроково завершити виконання циклу, можна перервати його з

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

допомогою оператора break.

- Один прохід циклу через блок операторів називається ітерацією.
- Якщо необхідно перервати виконання поточної ітерації та перейти до початку наступної, використовується оператор continue.
- Оператор while також може мати гілку else (за аналогією з if). На початку кожної ітерації інтерпретатор перевіряє істинність умови виконання циклу, і якщо вона є істинною, то виконує гілку while, інакше він виконує гілку else (якщо вона присутня) і завершує виконання циклу, причому це може статися і перед першою ітерацією, якщо умова спочатку була хибною. Однак якщо цикл був перерваний оператором break, то гілка else не виконується.
- **Цикл із лічильником (цикл for)** – це цикл, у якому змінна – лічильник ітерацій, що змінює своє значення від початкового до кінцевого з певним кроком.
- **Діапазони** – незмінні послідовності чисел, які задаються початком, кінцем та кроком. Пред ставлені класом range (Python 2 – xrange; range у Python 2 – це функція, яка повертає список).
- Початок за замовчуванням дорівнює нулю, крок – одиниці. Якщо встановити нульовий крок, буде викинуто виключення ValueError.
- Параметри конструктора повинні бути цілими числами (або екземпляри класу int або будь-який об'єкт з методом __index__).
- Елементи діапазону r визначаються за формулою r[i] = start + step * i, де i>=0 і r[i]<stop для step>0 або r[i]>stop для step<0.
- Підтримує всі загальні для послідовностей операції, крім повторення, а також у версіях Python до 3.2, зрізів і негативних індексів.
- Цикл for з лічильником у Python:

```
for змінна in range(кінцеве_значення):
    оператори
```

або

```
for змінна in range(початкове_значення, кінцеве_значення):
        оператори
```

або

```
for змінна in range(початкове\_значення, кінцеве\_значення, крок):
    оператори
```

Початкове значення входить у діапазон, кінцеве – ні.

Якщо змінна не використовується в коді, але все одно потрібна, відсутня потреба у вигадуванні імені для непотрібної змінної.

```
for _ in range(початкове_значення, кінцеве_значення, крок):
```


CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

операторі

- У першому варіанті змінна-лічильник набуває значення від нуля включно до заданого значення, не включаючи його; у другому задається початкове значення замість нуля; у третьому вказується ще й крок зміни змінної-лічильника (за замовчуванням 1).
- Можна встановити початкове значення більше, ніж кінцеве, і крок, рівний -1, і отримати зворотний цикл.
- Перед початком кожної ітерації змінна-лічильник приймає наступне значення із згенерованої інтерпретатором послідовності. Таким чином, будь-які зміни змінної-лічильника будуть втрачені на наступній ітерації. Після завершення циклу змінна-лічильник зберігається і дорівнює останньому значенню, яке вона набувала.
- Оператори break і continue у циклі for мають такий самий зміст, як і у циклі while.
- Аналогічно циклу while, цикл for опціонально може використовувати блок else.
- **Вкладені цикли** – це цикли, що знаходяться всередині інших циклів. Цикл, вкладений у тіло іншого, називається внутрішнім циклом. Цикл, у тіло якого вкладено інший цикл, називається зовнішнім.

Закріплення матеріалу

- Що таке цикл?
- Які типи циклів ви знаєте?
- Які типи циклів існують у мові Python?
- Охарактеризуйте цикл із передумовою. Як реалізувати його на Python?
- Охарактеризуйте цикл із лічильником. Як реалізувати його на Python?
- Як завершити виконання циклу з його тіла?
- Як пропустити ітерацію у циклі?
- Що таке вкладені цикли?

Додаткове завдання

Завдання 1
Дано числа a і b (a < b). Виведіть на екран суму всіх натуральних чисел від a до b (включно), які є кратними середньому арифметичному цього проміжку.

Завдання 2
Створіть програму, яка малює на екрані прямокутник із зірочок заданою користувачем ширини та висоти.

Завдання 3
Створіть програму авторизації, в якій користувачеві дається 3 спроби ввести свої облікові дані (логін та пароль). Якщо користувач за меншу кількість спроб ввів вірні дані, програма достроково припиняє своє виконання та виводить на екран повідомлення: «Авторизацію успішно пройдено з «№» спроби».

Самостійна діяльність учня

Завдання 1
Дано числа a і b (a < b). Виведіть суму всіх натуральних чисел від a до b (включно).

Завдання 2
Факторіалом числа n називається число n!=1·2·3·…·n. Створіть програму, яка обчислює факторіал введеного користувачем числа.

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Завдання 3
Використовуючи вкладені цикли та функції print('*', end=''), print(' ', end='') та print() виведіть на екран прямокутний трикутник.

Рекомендовані ресурси

Документація з Python
https://docs.python.org/3/reference/compound_stmts.html#the-while-statement
https://docs.python.org/3/reference/compound_stmts.html#the-for-statement
https://docs.python.org/3/reference/simple_stmts.html#break
https://docs.python.org/3/reference/simple_stmts.html#continue

Статті у Вікіпедії про ключові поняття, розглянуті на цьому уроці
https://uk.wikipedia.org/wiki/Цикл_(програмування)

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com