

Сумський державний педагогічний університет імені А.С.Макаренка

Фізико-математичний факультет

Кафедра інформатики

**Шумейко Володимир Юрійович**

**МЕТОДИЧНІ ОСОБЛИВОСТІ НАВЧАННЯ  
МОВИ ПРОГРАМУВАННЯ PYTHON УЧНІВ 5-7 КЛАСІВ**

Спеціальність: 014 Середня освіта (Інформатика)

Галузь знань: 01 Освіта/Педагогіка

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра

Науковий керівник:

\_\_\_\_\_ Н.В. Дегтярьова,  
кандидат педагогічних наук, доцент,  
доцент кафедри інформатики

« \_\_\_\_ » \_\_\_\_\_ 2021 року

Виконавець:

\_\_\_\_\_ В.Ю. Шумейко

« \_\_\_\_ » \_\_\_\_\_ 2021 року

Суми 2021

## ЗМІСТ

ВСТУП .....	3
РОЗДІЛ I. Обґрунтування доцільності використання мови Python в навчанні школярів основам програмування .....	5
1.1. Про мову Python .....	5
1.2. Вивчення алгоритмізації та програмування в 5-7 класах загальноосвітньої школи .....	8
1.3. Порівняльний аналіз мов програмування, використовуваних в навчанні школярів програмування .....	11
1.4. Python як основна мова програмування в середній школі.....	14
РОЗДІЛ II. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ ЩОДО ВИКОРИСТАННЯ МОВИ ПРОГРАМУВАННЯ Python В НАЧАННІ ПРОГРАМУВАННЮ УЧНІВ 5-7 КЛАСІВ.....	17
2.1.Методика викладання теми «Алгоритми і програми» в 6-7 класах класі .....	17
2.1. Методика викладання теми «Основні поняття алгоритмізації».....	19
2.3. Умовний оператор на мові Python... <b>Ошибка! Закладка не определена.</b>	
2.4. Цикли в мові Python..... <b>Ошибка! Закладка не определена.</b>	
2.5. Практичні аспекти вивчення основ алгоритмізації на прикладі мови програмування Python .....	25
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	37

## ВСТУП

Тема основ алгоритмізації та програмування незамінна в курсі інформатики. Вивчення цієї теми забезпечує сформованість основ логічного, алгоритмічного і математичного мислення. У цьому у вчителя є безліч інструментів - мов програмування. Однією з найбільш викладацьких мов програмування в школах і педагогічних вузах став Паскаль і його варіації. Однак Паскаль не найдосконаліший засіб, Паскаль як мова застаріла і більше не здатна конкурувати з сучасними мовами програмування. На цей момент існує безліч альтернатив Паскалю, однією з таких може виступити мова програмування Python.

В учнів часто виникають проблеми з розумінням типів даних, і з їх оголошенням, написанням і читаністю коду програми. Python дозволяє вирішити дані прогалини, адже дана мова відрізняється простим синтаксисом і «низьким порогом входження». Крім того, в даний час Python користується високим попитом на ринку інформаційних технологій, на відміну від Паскаля Python розвивається та є затребуваною мовою, займаючи четверте в списку найбільш популярних мов програмування. Знання Python стане в нагоді тим учням, що збираються продовжити навчання в сфері інтернет-технологій. Простота мови допоможе його освоєння слабким учням.

Об'єктом даного дослідження є: процес навчання програмуванню в курсі інформатики в загальноосвітній школі.

Предметом дослідження є методика навчання мови програмування Python учнів 5-7 класів

Метою написання даної роботи є вивчення методичних особливостей навчання мови програмування Python учнів 5-7 класів.

Завдання дослідження:

1. Вивчити структуру та зміст вивчення теми «Алгоритми і програми» в 5- 7 класах.

2. З'ясувати проблеми вибору мови програмування для вивчення в шкільному курсі інформатики.

3. Виконати порівняльний аналіз мови програмування Python з мовою програмування Паскаль найбільш використовуваним в навчальному процесі в загальноосвітній школі.

4. Розглянути особливості методики навчання мови програмування Python.

5. При вивченні основ алгоритмізації встановити практичні аспекти мови програмування Python.

## **РОЗДІЛ I. Обґрунтування доцільності використання мови Python в навчанні школярів основам програмування**

### **1.1. Про мову Python**

Python-популярна мова програмування. Вона була створена Гвідо ван Россумом і випущена в 1991 році.

Вона використовується для:

веб-розробки (на стороні сервера),  
розробка програмного забезпечення,  
математики,  
системних сценаріїв.

Python працює на різних платформах (Windows, Mac, Linux, Raspberry Pi тощо).

Python має простий синтаксис, схожий на англійську мову. Ця мова має синтаксис, який дозволяє розробникам писати програми з меншою кількістю рядків, ніж деякі інші мови програмування.

Працює в системі інтерпретатора, що означає, що код може бути виконаний, як тільки він буде написаний. Це означає, що прототипування може бути дуже швидким.

Найновішою основною версією Python є Python 3.

Python була розроблений для зручності читання і має деяку схожість з англійською мовою з впливом математики.

Використовує нові рядки для завершення команди, на відміну від інших мов програмування, які часто використовують крапки з комою або круглі дужки.

Також, ця мова використовує відступи, використовуючи прогалини, для визначення області, такої як область циклів, функцій і КЛАСІВ. Інші мови програмування часто використовують для цієї мети фігурні дужки.

Щоб почати працювати з Python, потрібно отримати доступ до інтерпретатора Python. Існує кілька загальних способів зробити це:

Python можна отримати на сайті Python Software Foundation [python.org](http://python.org). Як правило, це означає завантаження потрібного інсталяційного файлу для вашої операційної системи і запуску його на вашому комп'ютері.

В якості альтернативи, існує ряд сайтів, які дозволяють вам отримати доступ до інтерпретатора Python онлайн, без необхідності в установці чогось на вашому комп'ютері. Перш ніж почати вивчати основи мови програмування Python ви повинні мати стабільну версію інтерпретатора Python.

### Операції зі змінними і числами на пітоні

У програмуванні ми завжди взаємодіємо з даними, які не є якоюсь абстрактною субстанцією. Всі дані поділяються за певним типом. На попередніх уроках ми дізналися, про рядки, списки, словники і про логічний тип даних. Сьогодні, поговоримо про самому базовому типі даних в Python-числах.

Чому не можна звалити всі дані в одну загальну купу і не морочитися з кожним типом окремо? Припустимо, ми присвоїли змінної `a` цифру 5: `a = 5`. А тепер уявіть, що ніякого поділу за типом даних немає. Так що знаходиться в змінній: число або рядок? Якщо 10-це число, то з ним можна зробити математичні дії. А якщо це рядок, то ми маємо справу з текстом і тоді Python задіє зовсім інші методи.

```
# об'єднання рядків (конкатенація)
```

```
d = '10'
```

```
f = 'негрят'
```

```
d + ' ' + f
```

```
'10 негрят'
```

Ключовий момент: у кожного типу даних свої методи.

Цифра 100 написана без лапок, відноситься до числового типу даних. А цифра '100' в лапках - до рядкового типу. За допомогою синтаксису, ми

повідомляємо Python, який у нас тип даних. Числа в Python діляться на два види: ціле число і дійсне.

#### Ціле число

До цілих чисел (int) відносяться всі позитивні і негативні числа без дробової частини. Всі позитивні цілі числа називаються натуральними.

-80, -10, -4, 0, 1, 2, 20

#### Дійсне число

У речових чисел (float) завжди присутня дробова частина, а ще їх називають числами з плаваючою точкою. Оскільки дробова частина відділяється від цілої частини, точкою.

-5.2, -3.1, 7.8, 9.33

#### Математичні операції з числами

Заради математичних обчислень в Python існує числовий тип даних.

#### Додавання чисел

```
k = 13 + 155
```

```
print(e)
```

168

```
h = 5.7 + 4
```

```
print(h)
```

9.7

#### Віднімання чисел

```
d = 10 - 7
```

```
print(d)
```

3

#### Множення чисел

```
j = 3 * 7
```

```
print(j)
```

21

#### Розподіл чисел

У результаті ділення цілого числа на ціле, завжди буде дробова частина. Така особливість обумовлена тим, що в більшості випадків числа не діляться без залишку.

```
y = 30/10  
print(y)  
3.0
```

Цілочисельний поділ

В результаті цілочисельного ділення, завжди буде ціле число. Ми просто відкидаємо залишок. Число 15 поміститься цілих 2 рази.

```
m = 40 // 15
```

## **1.2. Вивчення алгоритмізації та програмування в 5-7 класах загальноосвітньої школи**

Відповідно до Державного стандарту базової і повної загальної середньої освіти курс «Інформатика» будується за такими змістовими лініями:

- інформація, інформаційні процеси, системи;
- комп'ютер як універсальний пристрій для опрацювання даних;
- комп'ютерні мережі, інформаційні технології створення та опрацювання
  - текстових документів, графічних зображень, числових даних, об'єктів
  - мультимедіа, комп'ютерних презентацій;
  - комп'ютерне моделювання;
  - основи алгоритмізації та програмування.

Програма побудована лінійно-концентрично. Зміст понять поступово розширюється і доповнюється. Лінійність реалізується шляхом ознайомлення

учнів з поняттями інформації, інформаційних процесів, систем і технологій, інформаційної моделі та комп'ютерним моделюванням, операційною системою й прикладними програмами захисту та архівування даних, графічним редактором, текстовим і табличним процесорами, засобами створення та опрацювання публікацій, редакторами презентацій й об'єктів мультимедіа, системою управління базами даних, сервісами Інтернету, поняттям алгоритму, базовими структурами алгоритмів, навчальним середовищем виконання алгоритмів. Змістова лінія «Основи алгоритмізації та програмування» є наскрізною для всього курсу.

Концентричність реалізує ознайомлення учнів з поняттями інформатики і інформаційно-комунікаційними технологіями за всіма змістовими лініями на різних рівнях складності, поступово доповнюючи і розширюючи їх зміст залежно від рівня сформованості загальнонавчальних навичок, вивченого навчального матеріалу з інших предметів і вікових особливостей розвитку учнів відповідних класів. Таким чином забезпечується поступове нарощування складності матеріалу, його актуалізація, повторення, закріплення, що сприяє формуванню предметної ІКТ-компетентності та ключових компетентностей і способів діяльності на більш високому рівні:

➤ перший рівень (5 – 7 класи) – ознайомлення з базовими поняттями курсу, формування орієнтувальної основи дій щодо роботи з персональним комп'ютером, комп'ютерними мережами, інформаційними технологіями, навчальним середовищем виконання алгоритмів, формування предметної ІКТ-компетентності та ключових компетентностей під час виконання репродуктивних і проблемних завдань, зокрема індивідуальних навчальних проєктів та компетентнісних задач, виконання яких передбачає використання однієї з інформаційних технологій або програмного середовища.

За новою програмою вивчення алгоритмізації та програмування повертається у шкільний курс інформатики і розглядається у кожній паралелі, починаючи з другого класу.

Вивчення алгоритмізації у шкільній інформатиці може мати два цільових напрями:

- розвивальний напрям: розвиток алгоритмічного мислення учнів;
- програмістський напрям: вивчення технології створення програм.

Останній напрям можна розділити на два цільових аспекти:

Перший аспект пов'язаний з посиленням фундаментальної компоненти курсу інформатики. Учням дається уявлення про те, що таке мови програмування, що представляє собою мова програмування високого рівня, та як вона створюється в середовищі системи програмування. Другий аспект носить профорієнтаційний характер. Вивчення програмування в рамках шкільного курсу дозволяє учням випробувати свої здібності до такого роду діяльності і, при бажанні, вибрати у майбутньому відповідний професійний шлях.

Алгоритмізація у школі відповідає методу структурного програмування і є підготовчим етапом до вивчення об'єктно-орієнтованого програмування, актуального на сучасному етапі розвитку програмування.

Стандартна програма вивчення основ алгоритмізації передбачає наступну послідовність тем:

- складання лінійних алгоритмів;
- складання циклічних алгоритмів;
- використання розгалужень в алгоритмах;
- опис і використання допоміжних алгоритмів.[6]

Таб.1 - Розподіл навчальних годин на вивчення розділу «Основи алгоритмізації та програмування» в 5-7 класах

Назва розділу	Клас і кількість годин			Всього
	5	6	7	
Основи алгоритмізації та	-	7	9	16

програмування				
---------------	--	--	--	--

### 1.3. Порівняльний аналіз мов програмування, використовуваних в навчанні школярів програмування

Перед тим як робити порівняння мов програмування, покажемо, що з себе представляє мова програмування Python.

Python - інтерпретована високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читаності коду. Створена Гвідо ван Россум і вперше випущена в 1991 році, Python підтримує структурне, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування.

Python поширюється безкоштовно і є кросплатформною. У більшості дистрибутивів операційної системи Linux міститься інтерпретатор Python.

Розберемо головні відмінні особливості мови програмування Python і зробимо порівняння з Паскалем.

#### 1. Простий синтаксис.

Python замість знаків пунктуації або ключових слів (в Паскалі такими словами є «begin» і «end»), використовує відступи для позначення виконання блоку. Програми, написані в один рядок або з іншими порушеннями в структурі, не зможуть бути виконаними в Python. Така особливість дозволить скоротити розмір коду і збільшити читаність програми. Синтаксис Python привчить школярів писати «красивий код», що поліпшить написання і розуміння коду. Так, наприклад, відрізнитися запис циклу на Паскалі і Python (Таблиця 1).

Таблиця 1 Порівняння синтаксису циклу з передумовою в Паскаль і Python

<i>Паскаль</i>	<i>Python</i>
----------------	---------------

<pre>while s + n &lt;150 do begin s: = s + 15; n := n - 5 end; writeln (n)</pre>	<pre>while s + n &lt;150: s = s + 15 n = n - 5 print (n)</pre>
--	--

## 2. Динамічна типізація.

Python має динамічну типізацію. Це означає що змінна зв'язується з типом під час надання значення, а значить немає необхідності заздалегідь оголошувати змінну. Це спрощує розуміння типів даних і знімає плутанину в різних довжинах цілочисельних і речових, строкових і символічних типах, а також скорочує розмір коду. Порівняння оголошення змінних в Паскаль і Python (Таблиця 2).

Таблиця 2 Порівняння синтаксису оголошення змінних в Паскаль і Python

<i>Паскаль</i>	<i>Python</i>
<pre>var s, n: integer; begin s: = 0; n := 75; end.</pre>	<pre>s = 0 n = 75</pre>

## 3. Компактний код.

Однією з очевидних переваг мови Python - компактність програмного коду. Наприклад, рішення задачі - поміняти місцями значення двох змінних - на мові Паскаль вирішується в три оператора, в Python в один рядок (Таблиця 3):

Таблиця 3 Порівняння синтаксису перепривласнення змінних в Паскаль і Python

<i>Паскаль</i>	<i>Python</i>

<pre>c := a; a := b; b := c;</pre>	<pre>a, b = b, a</pre>
------------------------------------	------------------------

#### 4. Високорівневі типи даних.

Python, будучи мовою дуже високого рівня, має вбудовані типи даних високого рівня, такі як динамічні масиви (списки) і словники.

У мові Python немає масивів в звичному розумінні цього терміна, але зате є списки, які можна вважати розширенням поняття «динамічний масив». Ми можемо окремо працювати з кожним елементом списку, а можемо виконувати операції зі всім списком, наприклад, додавати та видаляти елементи, копіювати частини списку, сортувати. Розглянемо приклад на заповнення масиву однаковими значеннями (Таблиця 4). Python справляється з цим завданням в один рядок, продублював масив, що складається з одного нуля.

Таблиця 4 Порівняння синтаксису заповнення масиву в Паскаль і Python

<i>Паскаль</i>	<i>Python</i>
<pre>const n = 100; var a: array [0..n - 1] of integer; for i: = 0 to n - 1 do a [i] := 0;</pre>	<pre>n = 100 a = [0] * n</pre>

#### 5. Інтерактивний режим в Python.

Python може значно заощадити час при розробці програми, оскільки не вимагає компіляції. Для трансляції програми в машинний код використовується інтерпретатор. Інтерпретатор може використовуватися в інтерактивному режимі, що дозволяє легко експериментувати з функціями мови, писати одноразові програми або тестувати функції у час розробки

програм. Інтерактивний режим підійде для самих перших уроків вивчення Python, так як забезпечує наочність процесу програмування (Таблиця 5).

Таблиця 5 Інтерактивний режим в Python

<i>Python</i>
>>> 50 - 5 * 6 20

На підставі даних відмінностей можна зробити висновок, що синтаксис і структура основних алгоритмічних конструкцій в Python багато в чому схожа з Паскалем. Однак вони наведені в досконалий вид, код став чистішим і коротшим, крім того, Python підтримує сучасні типи даних і необхідні функції для роботи з ними. Це робить мову відповідною для першого знайомства з програмуванням, особливо для школярів, і легким для вивчення самим учителем.

#### 1.4. Python як основна мова програмування в середній школі

Нині програмування стає невід'ємним інструментом діяльності широкого кола осіб: від фахівців у сфері послуг до ІТ-спеціальностей. Найпопулярнішими мовами програмування є Java, C, C++, Python, C#. Їх вивчення здійснюється, як правило, зі школи: учні освоюють мінімальний рівень володіння будь-якою мовою і можуть застосовувати наявні знання при вирішенні завдань. В силу того, що актуальними є професії, пов'язані з інформатикою, робототехнікою і програмнуванням, то важливим аспектом навчання в школі є актуальність отриманих знань, їх відповідність вимогам роботодавців, і запитам суспільства, тому одним з ключових питань стає вибір підходу інструменту навчання. На наш погляд, таким інструментом може стати мова Python.

Вона проста у вивченні і багатогранна у використанні. На відміну від мов програмування сімейства C, Python більш зрозуміла школярам в синтаксисі. Basic і Pascal, хоч і добре підходять для знайомства і

відточування навичок, але незважаючи на свою популярність, їх складно назвати сучасними. Ці мови витісняються продуктами більш високої цінності. Мови Java і Ruby орієнтовані на досвідчених користувачів, які можуть прогнозувати і знаходити помилки при написанні програмного коду, створювати програми, не розглядаючи їх в шкільному курсі. Таким чином, проміжне місце, на нашу думку, займає Python яку цілеспрямовано вивчати в шкільному курсі інформатики. Також, додаткове і більш глибоке вивчення мови Python, допоможе підготувати учнів до вивчення мов більш високого рівня, якщо дитина буде проявляти інтерес при її вивченні.

Звичайно, ставлення до Python різне. Існує думка, що ті люди, які почали вивчати мови програмування з Python, не хочуть переходити на низько рівневі мови, або не розуміють їх. На наш погляд, якщо дитина зацікавлена в програмуванні, то такої проблеми може не виникнути в силу того, що можливість вивчення нових мов сприяє розширенню рівня знань і дає можливість подивитися на програмний код з різних сторін, оцінити всі плюси і мінуси синтаксису мов і їх можливостей, встановити загальне і різне.

Python досить простий у вивченні, багатогранний і різноманітний у використанні. На Python активно пишуть різні програми, ігри, веб-сайти, а в школах реалізуються індивідуальні проекти. Відзначимо також той факт, що участь школярів в деяких олімпіадах передбачає наявність знань даної мови. В силу того, що час участі обмежений, школярам потрібно витратити якомога менше часу на написання програмного коду, в чому знову ж таки, виграє Python. Так, наприклад, знаходження мінімального елемента в рядку в Python можна знайти за допомогою написання одного слова `min`, в той час як в Pascal, наприклад, потрібно порівнювати елементи один з одним, вводячи додаткову змінну.

Важливо під час вивчення мови програмування школярам говорити про те, що в різних мовах при написанні програм використовуються різні конструкції, як у випадку з прикладом вище, але реалізуються одні і ті ж алгоритми. Використання таких «скорочених» конструкцій в мові Python

зеконмить час роботи учнів із завданням, і дає можливість вирішення і написання більшої кількості програм. Взагалі, за весь час навчання доцільно промовляти, звідки що береться, і чому використовується такий запис. Це одне з основних питань методики викладання мови Python. Завдання вчителя якраз-таки полягає в тому, що, перш ніж давати будь-який метод, необхідно розповісти про нього, і переконатися, що школярі розуміють, як він працює. Ось саме тоді і не виникне проблема переходу до мов нижчого рівня! Використання де - яких конструкцій типу max, min, sort і т.д. навпаки, полегшує роботу школяра, позбавляючи його від написання можливих помилок.

У школі можна працювати на Python як онлайн на сайті repl. it, так і встановивши спеціальну програму. Для роботи з Python досить мати будь-яку операційну систему Windows. Також корисною буде робота з сайтом <https://pythonguide.rozh2sch.org.ua/>, на якому покроково можна відстежити, що відбувається в тілі програми. Особливо важливо це показувати на перших уроках, щоб школярам було зрозуміло, що відбувається всередині програми. На сайті є як теоретична інформація, так і надається можливість вирішення деяких практичних завдань для тренування.

Головним недоліком Python, як зізнаються самі розробники, є те, що програми, написані на ньому, працюють повільніше, ніж створені на мовах (C, C++). Але часом важлива саме швидкість розробки, а не швидкодія програми. І як вже було згадано вище, писати на Python виходить швидше, ніж на багатьох інших мовах. Учнів необхідно з перших занять привчати до грамотного написання програмного коду, так як у великій і об'ємній задачі при виявленні помилки може знадобитися багато часу для її виправлення. До недоліків також можна віднести той факт, що наявність різних версій мови тягне за собою несумісність деяких програм.

## РОЗДІЛ II. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ ЩОДО ВИКОРИСТАННІ МОВИ ПРОГРАМУВАННЯ Python В НАЧАННІ ПРОГРАМУВАННЮ УЧНІВ 5-7 КЛАСІВ

### 2.1.Методика викладання теми «Алгоритми і програми» в 6-7 класах

У минулому навчальному 2020/2021 році учні 5-7 класів вивчали інформатику за навчальними програмами для учнів 5-9 класів, які можна знайти на офіційному веб-сайті Міністерства освіти і науки України (посилання: <http://mon.gov.ua/>).

Вчитель має право на власний розсуд розподілити кількість навчальних годин за темами, але за будь-якої траєкторії навчання на опанування теми «Алгоритми та програми» має приділятися не менше 40% навчального часу у 5–7 класах.

Враховуючи специфіку школи з поглибленим вивченням інформатики, в 7 класі виділено 2 години інформатики на тиждень (70 годин на рік).

Таблиця 6. – Орієнтовне календарно-тематичне планування з тематичної лінії «Основиалгоритмізації» в 7 класі

Номер уроку в темі	Тема уроку
Тема 1. Основні поняття алгоритмізації. Лінійні алгоритми (12 год.)	
1.	Правила поведінки і безпеки життєдіяльності (БЖ) в комп'ютерному класі.
2.	Алгоритм. Властивості алгоритмів. Способи подання алгоритмів. Базові алгоритмічні структури.
3.	Мови програмування. Елементи мови програмування. Структура програми.
4.	Середовище програмування та його основні елементи.
5.	Змінні. Типи даних. Дії з типами даних. Визначення типу результату обчислення арифметичного виразу.
6.	Введення та виведення даних.

7.	Арифметичні операції. Правила запису арифметичних виразів.
8.	Ідентифікатори. Правила запису імен ідентифікаторів. Цілочисельне ділення.
9.	Складання лінійних програм.
10.	Побудова лінійних алгоритмів та їх реалізація у вигляді програм.
11.	Розв'язування задач.
12.	Цікаві задачі.
Тема 2. Алгоритми з розгалуженням (10 год.)	
1.	Логічні вирази та логічні операції.
2.	Таблиці істинності.
3.	Команда розгалуження.
4.	Повна та скорочена форма умовного оператора мовою програмування.
5.	Складання програм.
6.	Побудова алгоритмів з послідовними розгалуженнями у вигляді програм.
7.	Оператор вибору.
8.	Складання програм.
9.	Вкладені розгалуження.
10.	Складання програм.
Тема 4. Алгоритми з повторенням (8 год.)	
1.	Базові алгоритмічні структури повторення. Команди повторення. Оператор циклу з умовою.
2.	Складання програм.
3.	Складання та виконання алгоритмів із повторенням у середовищі програмування.
4.	Оператор циклу з лічильником.
5.	Складання програм.
6.	Розв'язування задач.
7.	Розв'язування задач.
8.	<i>Практична робота №7.</i> Складання та виконання алгоритмів із повторенням та розгалуженням у середовищі <i>програмування</i> . Інструктаж з БЖД.

## 2.2. Методика викладання теми «Основні поняття алгоритмізації та програмування» на прикладі мови Python

Щоб розпочати роботу з інтерактивною оболонкою IDLE, яка є у дистрибутиві, її потрібно скачати з сайту: <http://python.org/download/>. Для учнів її буде достатньо.

Після запуску python.exe запрошення для введення буде мати вигляд:

```
>>>
```

Розглянемо основні практичні аспекти вивчення даної мови

Операції зі змінними і числами на Python

У програмуванні ми завжди взаємодіємо з даними, які не є якоюсь абстрактною субстанцією. Всі дані поділяються за певним типом. На попередніх уроках ми дізналися, про рядки, списки, словники і про логічний тип даних. Сьогодні, поговоримо про самому базовому типі даних в Python-числах.

Чому не можна звалити всі дані в одну загальну купу і не морочитися з кожним типом окремо? Припустимо, ми присвоїли змінної а цифру 5:  $a = 5$ . А тепер уявіть, що ніякого поділу за типом даних немає. Так що знаходиться в змінній: число або рядок? Якщо 10-це число, то з ним можна зробити математичні дії. А якщо це рядок, то ми маємо справу з текстом і тоді Python задіє зовсім інші методи.

```
# об'єднання рядків ( конкатенація)
```

```
d = '10'
```

```
f = 'негрят'
```

```
d + ' ' + f
```

```
'10 негрят'
```

Ключовий момент: у кожного типу даних свої методи.

Цифра 100 написана без лапок, відноситься до числового типу даних. А цифра '100' в лапках - до рядкового типу. За допомогою синтаксису, ми

повідомляємо Python, який у нас тип даних. Числа в Python діляться на два види: ціле число і дійсне.

#### Ціле число

До цілих чисел (int) відносяться всі позитивні і негативні числа без дробової частини. Всі позитивні цілі числа називаються натуральними.

-80, -10, -4, 0, 1, 2, 20

#### Дійсне число

У речових чисел (float) завжди присутня дробова частина, а ще їх називають числами з плаваючою точкою. Оскільки дробова частина відділяється від цілої частини, точкою.

-5.2, -3.1, 7.8, 9.33

#### Математичні операції з числами

Заради математичних обчислень в Python існує числовий тип даних.

#### Додавання чисел

```
e = 45 + 55
```

```
print(e)
```

100

```
j = 4.5 + 5
```

```
print(j)
```

9.5

#### Віднімання чисел

```
z = 15 - 4
```

```
print(z)
```

11

#### Множення чисел

```
i = 3 * 2
```

```
print(i)
```

6

#### Розподіл чисел

У результату ділення цілого числа на ціле, завжди буде дробова частина. Така особливість обумовлена тим, що в більшості випадків числа не діляться без залишку.

```
k = 9/3
```

```
print(k)
```

```
3.0
```

Цілочисельний поділ

В результаті цілочисельного ділення, завжди буде ціле число. Ми просто відкидаємо залишок. Число 15 поміститься цілих 2 рази.

```
m = 40 // 15
```

```
print(m)
```

```
2
```

Залишок від ділення

Відповіддю буде залишок від ділення. При звичайному розподілі, відповідь була б 15.1. Але нам потрібен тільки залишок. Прибираємо ціле число і залишаємо 1.

```
n = 16 % 3
```

```
print(n)
```

```
1
```

```
o = 12 % 3
```

```
print(o)
```

```
0
```

Зведення числа в степінь

Число перед двома зірочками-це об'єкт, який потрібно звести в ступінь. Цифра після зірочок позначає, в який ступінь зводимо: 4 зводимо в другу ступінь.

```
l = 4 ** 2
```

```
print(l)
```

```
16
```

У Python є вбудовані математичні функції.

## Модуль числа

Функція `abs ()` знаходить модуль числа. Передаємо в параметрах одне значення. Якщо передане значення негативне, то `abs ()` поверне позитивне число. Модуль числа не може бути негативним.

```
>>> abs(-5)
```

```
5
```

```
>>> abs(5)
```

```
5
```

## Найменше число

Функція `min()` в Python повертає найменше число.

```
>>> min(1,8,9)
```

```
1
```

## Максимальне число

Функція `max()` поверне найбільше число.

```
>>> max(25, 8, 57) 57
```

## Округлення до цілого числа

Функція `round ()` округлює до цілого числа.

```
>>> round(2.33)
```

```
2
```

```
>>> round(4.5)
```

```
4
```

## Вивести число в Python

Функція `print()` виводить числа на екран.

```
print(2,3,4)
```

```
2 3 4
```

## Ввести число в Python

Функція `input()` зчитує вводитьися значення користувачем в консоль і зберігає його в змінній. Але що станеться, якщо ми введемо якесь число і потім звернемося до змінної?

```
>>> r = input()
```

33

```
>>> r
```

```
'33' # це рядок
```

Python вивів рядок, так як число стоїть в лапках.

Для введення цілого числа, слід обернути функцію `input ()` в іншу функцію `int ()`.

```
>>> s = int(input())
```

```
22
```

```
>>> s
```

```
22
```

Для дійсного числа, відповідно в `float ()`.

```
>>> s
```

```
t = float(input())
```

```
11.9
```

```
>>> t
```

```
11.9
```

Як порахувати суму введених чисел?

У команді `input ()` можна передавати підказки.

```
w = int (input ( «введіть перше число: «))
```

```
q = int (input ( «введіть друге число: «))
```

```
summa=w+q
```

```
print(summa)
```

```
Введіть перше число: 6
```

```
Введіть друге число: 7
```

```
13
```

Як говорив один розумний програміст «Код повинен бути написаний так, щоб він сам був документацією до себе.

Форматування може виконуватися в так званому старому стилі або за допомогою строкового методу `format`. Старий стиль також називають Сі-

стилем, так як він схожий з тим, як відбувається висновок на екран в мові С. розглянемо приклад:

```
>>> pupil = «Ben «
>>> old = 16
>>> grade = 9.2
>>> print( «It's %s, %d. Level: %f « % (pupil, old, grade))
It's Ben, 16. Level: 9.200000
```

Тут замість трьох комбінацій символів

%s

,

%d

,

%f

підставляються значення змінних pupil, old, grade. Букви s, d, f позначають типи даних – рядок, ціле число, дійсне число. Якби було потрібно підставити три рядки, то у всіх випадках використовувалося б поєднання %s.

Хоча в якості значення змінної grade було вказано число 9.2, на екран воно вивелося з додатковими нулями. Однак ми можемо вказати, скільки потрібно знаків після коми, записавши перед буквою f точку з бажаним числом знаків у дробовій частині:

```
>>> print( «It's %s, %d. Level: %.1f « % (pupil, old, grade))
It's Ben, 16. Level: 9.2
```

Тепер подивимося на метод format():

```
>>> print( «This is a {0}. It's {1}. «.format( «ball «, «red «))
```

This is a ball. It's red.

```
>>> print( «This is a {0}. It's {1}. «.format( «cat «, «white «))
```

This is a cat. It's white.

```
>>> print( «This is a {0}. It's {1} {2}. «.format(1, «a «, «number «))
```

This is a 1. It's a number.

У рядку в фігурних дужках вказані номери даних, які будуть сюди підставлені. Далі до рядка застосовується метод `format ()`. У його дужках вказуються самі дані (можна використовувати змінні). На нульове місце підставиться перший аргумент методу `format ()`, на місце з номером 1 – другий і т. д.

Округлення до сотих

У функції

`round()` є ще один аргумент. Він показує до якої кількості знаків після коми слід округляти. Таким чином, якщо нам треба в Python округлити до сотих, цьому параметру слід задати значення 2.

Приклад округлення до потрібного знака:

```
round(3.555, 2) # = 3.56
```

Безумовно, вивчення школярами Python, в якості першої мови програмування, може викликати деякі закономірні побоювання: до таких будуть ставитися, перш за все, динамічна типізація і високорівневність мови. Однак безсумнівні і переваги вивчення Python в якості першої мови програмування в школі. Програми на Python істотно лаконічніше Pascal, що істотно полегшують завдання знайомства з мовою початківцям, так як пошук помилок і налагодження вимагає значно менших витрат.

### **2.3. Розробка циклу уроків**

Урок №1

Тема: Елементи мови Python і тии даних

Мета: сформувані знання в учнів про елементи та типи даних мови Python

Тип уроку: засвоєння нових знань, умінь та навичок

Обладнання: комп'ютер, зошит, проектор.

ПЗ: Python

Хід уроку

1. Організаційний момент:

Вчитель вітає учнів, перевіряє готовність учнів до уроків.

## 2. Вивчення нового матеріалу:

Сьогодні ми починаємо вивчення мови програмування Python. Чи чули ви про нього? Це сучасний і затребуваний мову, крім того, дуже простий і легкий у вивченні. У підручнику вивчається Паскаль, НЕ дивлячись на деяку схожість мов, між ними безліч відмінностей. Ці самі відмінності будемо записувати в зошит.

Мови програмування Python, як і іншим мовам властива система основних понять, в неї входять складу програми на Python, то є елементи мови, коментарі; типи даних, прості типи, структурні типи. Перед тим почати програмувати розглянемо цю систему.

Алфавіт. Алфавіт мови Python складається з безлічі символів, що включають в себе букви, цифри і спеціальні символи.

Латинські букви: від А до Z прописні) і від а до z. (рядкові). Цифри: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Спеціальні символи і комбінації символів: + - \* / = < > [ ] . , ( ) ; { } ^ @ \$ #, Пробіл. Ідентифікатори.

Ідентифікатором називається символічне ім'я певного програмного об'єкта. Такими об'єктами є імена констант, змінних, типів даних, процедур і функцій, програм. Ідентифікатори формуються по наступним правилам: ім'я може складатися з букви або послідовності букв, цифр і символу підкреслення, починатися така послідовність не повинна починатися з цифри; бажано, щоб ім'я відображало сенс величини; ім'я не має збігатися ні з одним з зарезервованих слів. Назва змінних, як і службові слів залежить від регістра.

Для того, щоб відзначити, що робить ту чи іншу ділянку коду можна застосувати коментарі. Інтерпретатор ігнорує коментарі, що не надає вплив на працездатність програми перед коментарем необхідно поставити знак решітки «#». Приклад коментаря «# Введення ступеня числа».

Букви українського алфавіту вживаються тільки в коментарях і строковому типі.

Концепція типів даних є однією з центральних в будь-якому мовою програмування. З типом величини пов'язані три її властивості: форма внутрішнього уявлення, безліч прийнятих значень і безліч допустимих операцій. Python має безліччю типів даних.

Логічний тип (Boolean) - bool. Представлений зумовленими об'єктами True і False, БРЕХНЯ і ПРАВДА. Важливо зауважити, що ці два значення є службовими, і їх написання завжди має бути з великою літери.

Числа представлені двома типами, цілочисельним (Integer) - int, і речовими (Float) - float для чисел з плаваючою точкою.

Також існує ще один чисельний тип complex для комплексних чисел, однак не станемо вдаватися в подробиці.

Строковий тип даних (String) - str. Крім чисел, Python також може маніпулювати рядками, які можуть бути виражені декількома способами. Вони можуть бути укладені в одинарні лапки ('...') або подвійні лапки ("...") з однаковим результатом.

Python включає кілька складових типів даних, використовуваних для угруповання інших елементів. Найбільш універсальним є список (list), іноді ще званий масивом, який можна записати в вигляді списку елементів, розділених комами, в квадратних дужках. Списки можуть містити елементи різних типів, але зазвичай все елементи мають однаковий тип. Приклад списку: «fibonacci = [0, 1, 1,2, 3, 5, 8] ».

Словники (Dictionary) - dict. Словник являє з себе набір послідовностей «ключ: значення» укладені в фігурні дужки, і розділені комами. Словники зручно використовувати завжди, коли виникає необхідність зв'язати значення з ключами, наприклад, щоб описати властивості чого-небудь. Словники можна перешкодити всередині інших словників. Наприклад, словником можна уявити шкільний журнал: «journal = { 'Смирнов': 4, 'Попов': 5 }».

Кортеж - tuple. У загальних рисах нагадує список, який неможливо змінити - кортежі є послідовностями, як списки, але вони є незмінними, як рядки. Синтаксично літерал кортежу полягає в круглій, а не в квадратній дужці. Приклад кортежу:  $T = (1, 2, 3, 4)$ .

Безлічі - set. Безлічі - це невпорядковані колекції унікальних і незмінних об'єктів. Безлічі створюються фігурними дужками, як словник, однак перераховуються запитом як список. Приклад безлічі:  $Y = \{ 'h', 'a', 'm' \}$ .

### 3. Підсумок уроку.

Чому навчилися на уроці? Є чи питання до вчителя?

### 4. Домашнє завдання:

Повторити зміст елементів мови програмування Python; структуру алфавіту; визначення ідентифікатора

### 5. Рефлексія:

Велике всім спасибі за роботу. Дуже радий, що ви уважно слухали, допомагали мені в роботі. І тепер головне, добре підготуватися вдома до наступного уроку.

## Урок №2

Тема: Операції, функції, вирази

Мета: сформувані знання в учнів про операції, функції, вирази мови Python

Тип уроку: засвоєння нових знань, умінь та навичок

Обладнання: комп'ютер, зошит, проектор.

ПЗ: Python

### Хід уроку

#### 1. Організаційний момент:

Вітає учнів, перевіряє готовність учнів до уроків

## 2. Вивчення нового матеріалу

Арифметичні операції. До числовим типам даних ставляться групи речових і цілочисельних типів. До них застосовані арифметичні операції і операції відносин.

Операції над даними бувають бінарними, застосовні до одного операнду, і бінарні, застосовні до двох операндам. Для перших кроків в програмуванні обмежимося вивченням однієї унарною арифметичної операцією в Python. Це операція зміни знака. Її формат: - <Величина>.

Перерахуємо бінарні арифметично операції.

Додавання

Віднімання

Множення

Ділення

Цілочисельне ділення

Стандартні функції. В Python існують велика кількість стандартних функцій, до яких програміст може звертатися в своїх програмах. Найбільш часто використовуються математичні функції, наприклад: `math.sqrt (x)` - квадратний корінь, `abs (x)` - абсолютна величина, `math.sin (x)` і ін. Часто використовувани стандартні функції: `input (...)` - функції введення, `print (...)` - процедура виведення даних.

Стандартні функції є зовнішніми підпрограмами по відношенню до викликає їх програмі. Вони об'єднані в модулі, які підключаються до основної програми і стають доступними для використання. Найбільш часто використовуваний модуль - Модуль `math`, один з найважливіших в Python. Цей модуль надає великий функціонал для роботи з числами.

Для підключення інших модулів необхідно в початку програми написати рядок: «`import`». Модуль `math` підключається наступним чином: `import math`.

Арифметичні вирази.

Арифметичний вираз задає порядок виконання дій над числовими величинами. Арифметичні вирази містять числові константи і змінні, арифметичні операції, функції, круглі дужки.

Наприклад, розглянемо математичний вираз.

$$2a + \sqrt{0,5 \sin(x + y)}$$

В Python воно виглядає так:

$$(2 * a + \text{math.sqrt}(0.5 * \text{math.sin}(x + y)))$$

Для того щоб правильно записувати арифметичні вирази, потрібно дотримуватися наступні правила.

1. Всі символи пишуться в рядок на одному рівні. Проставляються всі знаки операцій, то є знак множення пропускати не можна.

2. Чи не допускаються два наступних поспіль знака операцій, за винятком знака мінус, який можна поставити перед числом, яке хочемо зробити негативним.

3. Операції з більш високим пріоритетом виконуються раніше операцій з меншим пріоритетом.

Порядок зменшення пріоритетів: обчислення функції; унарна операція зміни знака (-); \*, /, //, %; +, -.

4. Кілька записаних підряд операцій однакового пріоритету виконуються послідовно зліва направо.

5. Частина виразу, укладена в дужки, обчислюється в першу чергу.

Чи не слід записувати вирази, НЕ мають математичного сенсу, наприклад розподіл на нуль, логарифм негативного числа і т. П.

3. Підсумок уроку.

Чому навчилися на уроці? Є чи питання до вчителя?

4. Домашнє завдання:

Повторити зміст елементів мови програмування Python; структуру алфавіту; визначення ідентифікатора

## 5. Рефлексія:

Велике всім спасибі за роботу. Дуже радий, що ви уважно слухали, допомагали мені в роботі. І тепер головне, добре підготуватися вдома до наступного уроку.

### Урок №3

Тема: Оператор присвоювання, введення та виведення даних

Мета: сформувані знання в учнів про оператор присвоювання, введення та виведення даних на мові Python

Тип уроку: засвоєння нових знань, умінь та навичок

Обладнання: комп'ютер, зошит, проектор.

ПЗ: Python

#### Хід уроку

##### 1. Організаційний момент:

Вітає учнів, перевіряє готовність учнів до уроків

##### 2. Вивчення нового матеріалу

Присвоєння - це дія, внаслідок якого змінна величина отримує певне значення. оператор присвоювання має наступний формат:

<Мінлива> = <Вираз>. наприклад:

1.  $x = 2 * a + \text{math.sqrt}(b)$

2.  $b = (x > y) \text{ and } (k \neq 0)$

Спочатку обчислюється вираз, потім отримане значення.

Присвоєння - це дія, внаслідок якого змінна величина отримує певне значення. оператор присвоювання має наступний формат:

<Мінлива> = <Вираз>. наприклад:

1.  $x = 2 * a + \text{math.sqrt}(b)$

2.  $b = (x > y) \text{ and } (k \neq 0)$

Спочатку обчислюється вираз, потім отримане значення присвоюється змінній. У першому прикладі наведено арифметичний оператор

присвоювання. У другому прикладі логічний оператор присвоювання. При присвоєнні типи змінної і виразу можуть НЕ збігатися, при наступному присвоєнні, тип змінної підлаштуватися під нового значення, дане властивість типів називається динамічної типизацією.

Python володіє динамічною типізацією, це означає, що тип даних присвоюється разом зі значенням, а значить якщо переприсвоїть тип деякої змінної, значить переприсвоїть її тип. Однак це правило стосується лише присвоєння, існують функції, які мають на увазі певний тип даних.

Введення і висновок даних. Під введенням розуміється передача даних з зовнішнього пристрою комп'ютера, з клавіатури в оперативну пам'ять. При виведенні дані передаються з оперативної пам'яті на зовнішні пристрій, на монітор.

Операція введення називається читанням і виконується з допомогою оператора `input ()`. Висновок називається записом, і для його виконання використовується `print ()`. Всередину дужок поміщається аргумент функції.

Введення даних з клавіатури. Дано деякої змінної запис з клавіатури, для цього скористаємося функцією `input ()`. Цю функцію слід привласнити деякої змінної. Як аргумент функції введемо то, повідомлення, яке виведеться при введенні з клавіатури, наприклад таким повідомленням може бути: «Як вас звати?». Наприклад так, `name = input ("Як вас звать?")`.

При виконанні цього оператора відбувається переривання виконання програми, після чого користувач повинен набрати на клавіатурі значення змінної. При цьому вводиться значення висвічується на екрані. За закінченню введення натискається клавіша `Enter`. Якщо операторів `input` кілька, значить кожен з них буде при виконанні відділятися клавішою `Enter`.

Висновок даних. Виведемо повідомлення "Hello World!", Для цього використовуємо функцію `print ()`.

Аргументом даної функції є запис деякого типу або змінна, повний запис буде виглядати так: `print ('Hello World!')` або так

```
a = 'Hello World!' print (a).
```

Аргументи функцій `input ()` і `Print ()` може приймати кілька значень. Відокремлюються змінні запитом. наприклад:

```
name = "Сергій" print ( 'Привіт, ', name)
```

Лінійна програма. дотримання

- найпростіша алгоритмічна структура. Програма, що реалізує проходження, називається лінійною програмою.

Одним з обов'язкових умов гарного стилю програмування є організація діалогу між комп'ютером і користувачем. Таке діалогове взаємодія називається інтерактивним інтерфейсом.

### 3. Самостійна робота

Запустимо інтерпретатор Python, програма називається IDLE і виконаємо два простих завдання на створення лінійної програми.

Приклад 1. Складемо лінійну програму, по якій в діалозі будуть вводитися два цілих числа з клавіатури і обчислюватися їх твір.

```
A = input ( 'A = ')
B = input ( 'B = ')
print (A, ' * ', B, ' = ', A * B)
```

Приклад 2. Дано натуральне тризначне число. Потрібно обчислити суму його цифр. Наприклад, якщо дано число 325, то в результаті повинно вийти:  $3 + 2 + 5 = 10$ .

Спочатку складемо програму, а потім її прокоментуємо.

```
N = input ( 'Введіть тризначне число: ')
Sum = 0
Sum = Sum + N % 10 N = N // 10
Sum = Sum + N % 10 N = N // 10
Sum = Sum + N print ( 'Сума цифр = ', Sum)
```

У цій програмі використані дві операції цілочисленної арифметики: «//» - цілочисельне поділ і % - залишок від цілочисельного ділення. залишок

від ділення 10 виділяє молодшу цифру числа, а целочисленне розподіл на 10 відкидає молодшу цифру.

### 3. Підсумок уроку.

Чому навчилися на уроці? Є чи питання до вчителя?

### 4. Домашнє завдання:

Повторити зміст елементів мови програмування Python; структуру алфавіту; визначення ідентифікатора

### 5. Рефлексія:

Велике всім спасибі за роботу. Дуже радий, що ви уважно слухали, допомагали мені в роботі. І тепер головне, добре підготуватися вдома до наступного уроку.

## **2.3. Методичні рекомендації щодо проведення розроблених уроків**

1. Поставлені цілі і завдання знайшли своє відображення в плані-конспекті, вони простежуються по ходу уроку і відповідають поставленим темі.

2. Навчальний матеріал побудований як лекційний матеріал, адже підручники для 6 і класів з інформатики не мають теоретичного матеріалу щодо Python, грамотно вибудований хід уроку.

3. План-конспект має на увазі постійне спілкування і взаємодія вчителя і учнів.

4. На уроці планується використання мультимедійної презентації, що безсумнівно робить урок більш привабливим, а матеріал легким для сприйняття учнями.

## ВИСНОВКИ

Програмування - складна глава в курсі інформатики, вибір мови програмування дозволяє визначити успішність в освоєнні даного курсу. Запропонована нами для використання в навчанні мова програмування Python легша у вивченні, дозволяє вирішити ряд складнощів в програмуванні, з якими стикаються учні. Крім того, Python є затребуваною і оновлюваною мовою програмування.

Вивчивши науково-методичну, нормативно-правову, навчальну літературу з питань навчання школярів основам програмування в базовому курсі інформатики, слід зауважити, що державний стандарт не встановлює єдину мову програмування. Визначено лише вміння і знання, якими учень повинен оволодіти. Проаналізувавши навчальні програми, зауважимо, що учнів переважно навчають мови програмування Паскаль. Схожість основних структур Паскаля і Python, а також, робить їх придатними для вивчення на їх основі мови програмування Python.

Порівняльний аналіз мов програмування Python і Паскаль дозволив показати гідності Python, такі як простий синтаксис, сучасні типи даних і динамічна типізація, компактний код, цікаві функції і наявність інтерактивного режиму.

Мета даної роботи: вивчення методичних особливостей навчання мови програмування Python учнів 5-7 класів, була досягнута

Також було досліджено:

Структуру та зміст вивчення теми «Алгоритми і програми» в 5- 7 класах.

З'ясували проблеми вибору мови програмування для вивчення в шкільному курсі інформатики.

Виконали порівняльний аналіз мови програмування Python з мовою програмування Паскаль найбільш використовуваним в навчальному процесі в загальноосвітній школі.

Розглянули особливості методики навчання мови програмування Python.

Встановили практичні аспекти вивчення основ алгоритмізації на прикладі мови програмування Python

В даний час програмування стає широко поширеним, а мови програмування — простими і зручними у використанні. Потреба в людях, які добре знають Python буде тільки зростати, отже ми вважаємо, що Python може стати першою мовою в області вивчення програмування в середній школі.

В результаті можна відзначити, що не тільки немає необхідності відмовлятися від ідеї вивчення високорівневих мов програмування в школі, а навпаки, вивчення Python, при правильному підході і обліку методичних особливостей, відкриє перед учнем нові горизонти і можливості, так як сучасні мови програмування, вдосконалюючись, стають все більш універсальними, гнучкими і простими, зручними для сприйняття і налагодження.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. В. В. Зубик. Python. Практикум програмування для початківців. – 56 с. URL: <https://drive.google.com/file/d/0B3GWrzARFojMLU9jMGEwQjNUUE0/> (дата звернення: 17.05.2021).
2. Гаєвський О. Ю. Інформатика : 7-11 кл. Навч. Посіб. – К.: Видавництво А.С.К., 2004. – 512 с. : іл.
3. Златопольський Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
4. Інформатика (Морзе, Барна) 7 клас 2020. Електронний підручник Інформатика 7 клас Морзе по новій програмі, НУШ 2020 рік. Рік: 2020 Підручник за 7 клас » Інформатика Автор: Морзе, Барна. URL: <https://pidruchnyk.com.ua/1494-informatyka-7-klas-morze-2020.html> (дата звернення: 19.05.2021).
5. Караванова Т.П. Інформатика: методи побудови алгоритмів та їх аналіз. Необчислювальні алгоритми: Навч. посіб. для 9-10 кл. із поглибл. вивч. інформатики. – К.: Генеза. – 2007.- 216 с.: іл.
6. Навчальна програма з інформатики для 5-9-х класів для загальноосвітніх навчальних закладів затверджена наказом МОН від 07.06.2017 № 804
7. Програмування числових методів мовою Python : підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр «Київський університет», 2014. – 640 с.
8. Юрченко І.В., Сікора В.С. Інформатика та програмування. Частина 2.– Чернівці: Видавець Яворський С.Н., 2015.– 210 с