

Коротенко Г.М.

Національний технічний університет «Дніпровська політехніка»

Соколова Н.О.

Національний технічний університет «Дніпровська політехніка»

Ширін А.Л.

Національний технічний університет «Дніпровська політехніка»

НАВЧАННЯ В ЗАКЛАДАХ ЗАГАЛЬНОЇ СЕРЕДНЬОЇ ОСВІТИ ОСНОВ ПРОГРАМУВАННЯ ЗА ДОПОМОГОЮ МОВИ PYTHON

Поточним часом, вектор спрямованості щодо розвитку у учнів уявлень застосування навичок алгоритмічного мислення все більше ускладнюється у зв'язку з різкою зміною прикладних завдань, які повинні вирішувати сучасні спеціалісти у ІТ-сфері. Окрім суто прикладних задач, пов'язаних з комп'ютерними або обчислювальними експериментами з цифровими двійниками та іншими цифровими моделями, надзвичайно зростає обсяг різноманітних даних різної природи. Такі дані в першу чергу стосуються наскрізних технологій і, у тому числі, завдань, пов'язаних з використанням моделей і засобів штучного інтелекту. Зрозуміло, що надання відповідних ІТ-компетентностей майбутнім студентам і фахівцям цифрових технологій є нагальною місією сучасної школи. Але тут треба мати на увазі, що не всі учні в майбутньому будуть професійними програмістами. Таким чином, виникає комплекс пов'язаних поміж собою завдань, щодо формування теоретичних, практичних, технологічних, психологічних та деяких інших комплексних процедур, спрямованих на вирішення виникаючих проблем. На тлі збільшення обсягів проблем, що стрімко ускладнюються, також вагомим елементом поліпшення методичних підходів щодо підвищення рівня навчання і учнів, і педагогічних працівників, які ведуть навчальний процес, є обрання відповідної мови програмування та її організаційного оточення.

В роботі доведено, що поступове, підкрплене відповідними теоретичними і практичними (у тому числі, комп'ютерними) засобами засвоєння основ програмування не тільки можливе, а й необхідне. Згідно аналізу сучасних підручників з інформатики для середньої школи можна резюмувати, що програмування як таке практично відсутнє у практичних завданнях. Таким чином, вчителі повинні самі обирати: мову програмування, методи викладання, засоби підтримки навчального процесу, відповідний програмний засіб та його версію.

У статті наведено приклади можливих напрямів застосування найбільш поширеної та простої у вивченні мультипарадигмальної, високорівневої мови програмування Python загального призначення з динамічною строгою типізацією та автоматичним управлінням пам'яттю, орієнтованою на підвищення продуктивності розробника, спрощення читання коду та поліпшення його якості, а також на забезпечення переносимості написаних з її застосуванням програм.

Ключові слова: програмний код, мови програмування, алгоритми, навчання, мова програмування Python, структура програмування.

Постановка проблеми. Стрімкий розвиток елементної бази та програмного забезпечення комп'ютерних технологій створюють безліч нових технологічних напрямків застосування комп'ютерних засобів. Так в даний час існують і розвиваються інформаційні, цифрові та наскрізні технології обробки різноманітних даних, а також безліч різнотипних платформ, у тому числі цифрових, для їх підтримки. Це, безумовно, викликає необхідність навчання в закладах загальної середньої освіти основ програмування. Повна загальна середня освіта має три рівні освіти: початкова

освіта тривалістю чотири роки; базова середня освіта тривалістю п'ять років; профільна середня освіта тривалістю два роки. На етапі початкової освіти добре себе зарекомендував язык програмування Scratch. Це середовище та інтерпретована динамічна візуальна мова програмування, у якій код створюється шляхом маніпулювання графічними блоками. Середовище орієнтовано, в першу чергу, на дітей та початкове знайомство з основними концепціями та ідеями програмування. Завдяки динамічності, вона дає змогу змінювати код навіть під час виконання. Мова має за

мету навчити дітей поняттю програмування і дає можливості створювати ігри, анімації чи музику. Користувачі можуть створювати онлайн-проекти, ними можна обмінюватися всередині міжнародної спільноти, яка існує в мережі Інтернет. Середовище програмування можна безкоштовно завантажити і вільно використовувати у шкільній чи позашкільній освіті [1].

На етапах базової та профільної середньої освіти потрібно вивчати вже одну з мов програмування високого рівня, тобто C++, C#, Java, JavaScript, Python та ін. [2].

Аналіз останніх досліджень і публікацій. Базовий нормативний документ для організації освітнього процесу Нової української школи (НУШ) (Держстандарт) визначає 9 освітніх галузей, які мають розвивати природні здібності, інтереси та обдарування учнів, формувати компетентності, необхідні для їх соціалізації та самореалізації, виховувати громадянина з відповідальною життєвою позицією [3].

У Держстандарті для кожної освітньої галузі визначено мету, компетентнісний потенціал (перелік умінь та ставлень й базові знання), обов'язкові результати навчання учнів та орієнтири для їхнього оцінювання.

Метою інформатичної освітньої галузі є «розвиток особистості учня, здатного використовувати цифрові інструменти і технології для розв'язання проблем, розвитку, творчого самовираження, забезпечення власного і суспільного добробуту, критично мислити, безпечно та відповідально діяти в інформаційному суспільстві» (стаття 18 ДСБСО [3]).

Інформатична освітня галузь стала однією з лідерів серед навчальних дисциплін в школі. За обсягом годин вона поступається лише математиці та українській мові. І це є нагальною вимогою часу. Її розробники формують вимоги до обов'язкових результатів навчання учнів у вигляді чотирьох груп умінь, друга з яких це – створення інформаційних продуктів і програм для ефективного розв'язання задач/проблем, творчого самовираження індивідуально та у співпраці з іншими особами за допомогою цифрових пристроїв чи без них, а саме: «Розробка та реалізація алгоритмів» (п. 2.1) та «Створення та налагодження програмних проєктів» (п. 2.2) (додаток 14 до ДСБСО [3]). Є правильним, що всередині інформатики збільшено кількість годин на змістову лінію алгоритмізації та програмування [4].

Держстандартом передбачено, що школярі мають навчитися розробляти та коригувати алго-

ритми для розв'язання задач. Орієнтирами для оцінювання результатів навчання є сформованість таких знань, умінь та навичок школярів [3]:

- складання лінійних, розгалужених і циклічних алгоритмів для розв'язання задач;
- представлення алгоритму одним чи кількома способами;
- поєднання базових структур для розв'язання задачі;
- пропонування та використання способів перевірки коректності алгоритму;
- пошук, пояснення та пропонування варіантів виправлення простих логічних помилок;
- формулювання висновку щодо відповідності алгоритму для розв'язання задачі.

Заклад освіти з урахуванням особливостей технічного забезпечення, кадрового складу, контингенту, освітніх пріоритетів учнівства тощо може обрати різні варіанти викладання курсу інформатики, створюючи власну навчальну програму на основі модельної [5]. У модельній навчальній програмі авторські колективи пропонують послідовність вивчення тем, зміст і види навчальної діяльності, які є орієнтовними. Тому з модельної навчальної програми вчитель / учителька до кожної теми навчальної програми добирає з-поміж запропонованих або ж додає ті елементи змісту й способи діяльності, що в умовах певного навчального закладу й класу є найоптимальнішими для кожного учня / учениці та для учнівської спільноти класу.

Заклад освіти обирає свій варіант викладання курсу інформатики, враховуючи свої технічні можливості, кадровий склад, контингент учнів та освітні пріоритети [5]. На основі модельної навчальної програми, розробленою авторським колективом досвідчених та визнаних фахівців-методистів, яка пропонує послідовність вивчення тем, зміст та види навчальної діяльності (рис. 1), заклад освіти створює власну навчальну програму [6]. Вчитель може обрати запропоновані елементи з модельної програми або додати власні, що найкраще підходять для конкретного навчального закладу та класу, визначаючи послідовність та орієнтовний час вивчення тем для певного класу.

У роботах [8, 9] вказується на те, що у середній школі потрібно вивчати мову програмування Java. Окрім того у роботі [4] пропонується наступна схема навчання програмуванню:

1. Scratch-програмування (2–4 класи);
2. Основи структурного (алгоритмічного) програмування (5 клас);
3. Ознайомлення з об'єктним програмуванням (6 клас);

АВТОРИ: Бондаренко О. О., Ластовецький В. В., Пилипчук О. П., Шестопапов Є. А.	АВТОРИ: Морзе Н. В., Барна О. В.	АВТОРИ: Пасічник О. В., Козак Л. З., Ворожбит А. В.
Служби інтернету	Інформація, повідомлення, дані, знання. Інформаційні процеси	Персональний цифровий простір
Комп'ютерна анімація	Електронні сервіси. Електронна пошта	Цифрове середовище для навчання та роботи
Моделі та моделювання	Векторна графіка	Візуальний контент
Опрацювання табличних даних	Текстовий процесор	Тексти та публікації
Алгоритми та програми	Програми для роботи з поліграфічною продукцією	Медіа-дизайн
Практикум з використання інформаційних технологій	Комп'ютерна анімація	Графічне програмування

АВТОРИ: Завадський І. О., Коршунова О. В., Твердохліб І. А.	АВТОРИ: Громко Г.Ю., Шевчук П.Г., Ковбаса В.М.	АВТОРИ: Ривкінд Й. Я., Лисенко Т. І., Чернікова Л. А., Шакотько В. В.
Цифрові інструменти для навчання	Інформаційні процеси та системи	Пошук в Інтернеті. Електронна пошта. Хмарні сервіси
Текстовий процесор. Основи верстки	Комунікаційні технології та мережі	Алгоритми та програми
Моделювання в електронних таблицях	Організація власного інформаційного середовища	Комп'ютерні презентації
Алгоритми та програми	Комп'ютерна графіка та анімація	Об'єкти мультимедіа
	Опрацювання табличних даних	Комп'ютерна анімація
	Основи програмування	Практикум з використання інформаційних технологій
	Захист та взаємодія в цифровому просторі	

Рис. 1. Порівняння модельних програм інформатичної освітньої галузі для 7-9 класів НУШ [7]

- Ознайомлення із програмуванням на Android (7 клас);
- Основи об'єктного програмування (8 клас);
- Основи функціонального програмування (9 клас).

На погляд авторів, такий підхід може тільки злякати більшість школярів, оскільки, згідно дослідженню програми IREX «Мріємо та діємо», приблизно 52% підлітків віком 13–15 років уже вирішили, ким мріють стати в дорослому віці: зокрема, хочуть стати програмістами (22%), лікарями (13%) і дизайнерами (9%) [9]. З іншого боку, включення у розгляд та намагання на початкових кроках охопити весь комплекс широкого спектра суто «програмістських» понять, включаючи і такі абстракції високого рівня як класи, може суттєво ускладнити освоєння самих принципів і основ програмування, не тільки учням, але й викладачам шкіл. Тому, на наш погляд, треба зосередитись на таких коцептуальних засадах:

- Кодування допомагає дітям розвивати впевненість.
- Кодування надихає дітей дивитися на світ інакше (і логічно).

- Кодування дозволяє дітям досліджувати свою творчість.

- Кодування покращує навички дітей розповідати історії.

- Кодування може сприяти покращенню кар'єрних можливостей у міру дорослішання [10].

Постановка завдання. Метою статті є обґрунтування вибору мови Python як засобу навчання учнів основам програмування та методики їх освоєння.

Виклад основного матеріалу. Сьогодні жодна галузь людського суспільства, чи медицина, чи освіта, чи економіка, чи транспортний напрям не обходиться без комп'ютерних технологій. Тому дуже важливо починати вивчення програмування з мови яка знаходиться в тренді галузі комп'ютерингу. Згідно рейтингу PYPL (Popularity of Programming Language – Популярність мов програмування) [11] перші три позиції займають мови програмування Python, Java та JavaScript (рис. 2).

Якщо казати про мову JavaScript, то це мова програмування та основна технологія Інтернету, яку можна поставити у один ряд з мовами HTML

Worldwide, Sept 2024 :

Rank	Change	Language	Share	1-year trend
1		Python	29.66 %	+1.6 %
2		Java	15.64 %	-0.2 %
3		JavaScript	8.3 %	-1.0 %

Рис. 2. Найпопулярніші мови програмування серед ІТ-спеціалістів у 2024 згідно рейтингу PYPL

і CSS. 99% веб-сайтів використовують JavaScript на стороні клієнта для формування функцій поведінки веб-сторінки відповідно до зовнішніх запитів [12].

Щоб порівняти Python та Java, що у 2024 році стабільно займають перші позиції практично усіх відомих рейтингів, зробимо наступне. Звернемось до сайту “Hello World” In 30 Different Languages («Hello World» 30 різними мовами) [13].

Згідно матеріалів цього сайту, щоб вивести рядок тексту «Hello World» мовою Python нам буде потрібен тільки один рядок:

```
print("Hello World")
```

А от, щоб вивести один такий текстовий рядок мовою Java, вже потрібно застосувати шість рядків операторів цієї мови:

```
import java.io.*;

class GFG {
    public static void main (String[] args) {
        System.out.println("Hello World");
    }
}
```

Річ у тому, що Java – це високорівнева об’єктно-орієнтована мова програмування на основі класів [14]. Зазвичай, об’єктно орієнтоване програмування пов’язане зі створенням «об’єктів», які є екземплярами класів (і це, на перших кроках засвоєння основ програмування для початківців – учнів та вчителів (педагогів), може бути досить неочевидним). Згідно основам об’єктно-орієнтованого програмування (ООП) створені програмістом об’єкти мають атрибути (дані) та методи (функції), які працюють із даними, що обробляються. Ключові концепції ООП включа-

ють інкапсуляцію (об’єднання даних та методів), успадкування (створення нових класів з існуючих) та поліморфізм (використання методів різними способами для різних об’єктів) [15]. В роботі [16] вказується, що розуміння об’єктно-орієнтованих концепцій завжди є складним завданням для студентів (!). Викладачам також важко викладати ці концепції. Тому, на погляд авторів, починати вивчення програмування з включення у розгляд достатньо складних компонентів парадигми ООП не зовсім доцільно.

Процедурне програмування – це метод написання програмного забезпечення (ПЗ), який заснований на послідовності інструкцій або кроків, які слідують один за одним для виконання завдання. Ви можете думати про це як про рецепт, у якому кожен крок потрібно виконувати в певному порядку, щоб досягти бажаного результату. Ця парадигма зосереджена навколо написання функцій або процедур, які виконують операції з даними. Процедурний код, як правило, легший для розуміння багатьма, оскільки він простий і лінійний [15]. Мова Python підтримує декілька парадигм програмування, включаючи процедурне, об’єктно-орієнтоване та функціональне програмування [17]. Це дозволяє починати вивчення основ програмування з процедурного програмування як найбільш корисного для освоєння основних принципів алгоритмізації і кодування.

Ще однією з важливих рис мови Python є так зване «Правило поза грою» [18]. Це правило описує синтаксис мови комп’ютерного програмування, у котрій межі блоку коду визначаються за допомогою відступу на відміну від Java, у якій блок визначається за допомогою фігурних дужок «{}». При наявності декількох вкладених блоків відстежування дужок складає певну проблему, як це можна побачити на вищенаведеному прикладі виведення простого текстового рядка мовою Java.

Якщо казати про галузі застосування мови Python, то для розробників тут відкривається дуже широкий простір у сферах застосування сучасних цифрових технологій [19]:

1. ІІІ (штучний інтелект) і машинне навчання;
2. Аналітика даних;
3. Візуалізація даних;
4. Програмування додатків;
5. Веб-розробка;
6. Розробка ігор;
7. Розробка мов програмування;
8. Фінанси;
9. SEO (search engine optimisation – пошукова оптимізація);
10. Дизайн.

Тут треба додати, що Python має більш ніж 137 000 бібліотек. Ці бібліотеки надають неосяжні можливості повторного використання програмного коду, що, з одного боку, суттєво прискорює створення програм, а з іншого – дозволяє швидко долучитися початківцям до вирішення складних практичних задач. Серед досить широкого кола цих програмних засобів, що розповсюджуються з відкритим кодом, найбільш поширеними є наступні десять [20].

1. Pandas (обробка та аналіз широкого кола типів даних).
2. NumPy (виконання наукових обчислень з великими матрицями та багатовимірними даними).
3. Keras (експериментування з усіма типами глибоких нейронних мереж).
4. TensorFlow (велика кількість чисельних методів для роботи у алгоритмах глибокого навчання та машинного навчання).
5. Scikit Learn (наявність засобів для виконання великої кількості алгоритмів з контрольованим або неконтрольованим навчанням).
6. Eli5 (засоби для уточнення прогнозів при дослідженні процесів машинного навчання).
7. SciPy (засоби для наукових досліджень, обробки даних і високопродуктивних обчислень).
8. PyTorch (комплекс методів обробки природної мови з використанням графічних процесорів).
9. LightGBM (пакет машинного навчання для розробки нових алгоритмів шляхом перевизначення простих моделей, таких як дерева рішень).
10. Theano (набір алгоритмів, що дозволяє визначати, оптимізувати та оцінювати математичні вирази для величезних алгоритмів нейронних мереж глибокого навчання (Deep Learning)).

Слід також додати, що багато з вище зазначених пакетів пов'язані поміж собою.

Іншим важливим фактором зниження порогу складності засвоєння даної мови може бути її об'єднання з одним з найпоширеніших додатків обчислювального напрямку – Microsoft Excel [21]. Використання цього додатку у практичних заняттях учнів є достатньо поширеним і тому, вивчення мови Python разом з ним, може суттєво розширити можливості включення і дітей, і вчителів у сферу поглиблення знань програмування.

Таким чином, спираючись на процедурну парадигму Python, при вивченні мови пропонується використовувати єдиний підхід з упором на концептуальні поняття. На думку авторів, основну увагу необхідно приділяти поняття вбудованих (базових) та агрегатних типів даних [22].

Наступним пунктом у цьому підході є наголос на структурне програмування, засноване на застосуванні структурної теореми Боме та Джакопіні [23]. За Боме та Джакопіні для побудови логічної структури програми потрібно три основні складові блоки:

- функціональний блок *СЛІДУВАННЯ*, що включає лінійну послідовність операторів (у тому числі і керуючих) (рис. 3);
- умовний оператор (конструкція прийняття двійкового чи дихотомічного рішення – *РОЗВИЛКА*) (рис. 4);
- конструкція узагальненого циклу *ЦИКЛ* (з параметром, з передумовою, з постумовою тощо) (рис. 5).

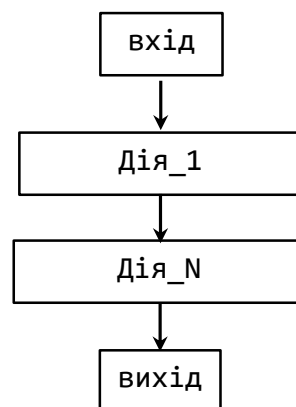


Рис. 3. Лінійна послідовність операторів (*СЛІДУВАННЯ*)

Тоді для будь-якої програми, складеної із зазначених конструкцій, завжди може бути отримано ще й доказ її правильності.

При цьому важливо те, що всі розглянуті блоки мають один вхід і один вихід. Користуючись ними, можна конструювати програми будь-якої складності.

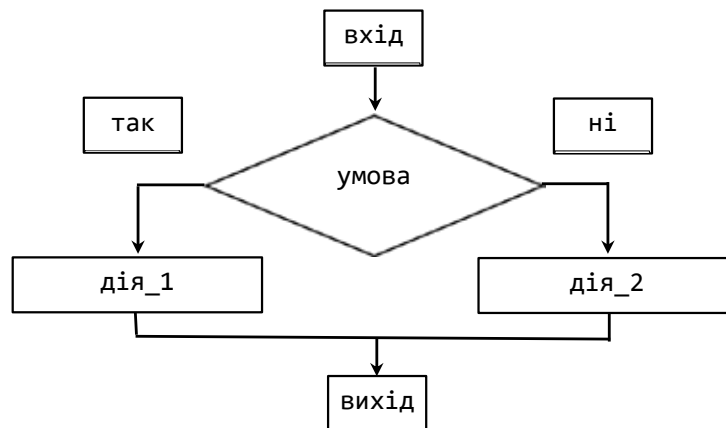


Рис. 4. Керуючий умовний оператор (РОЗВИЛКА)

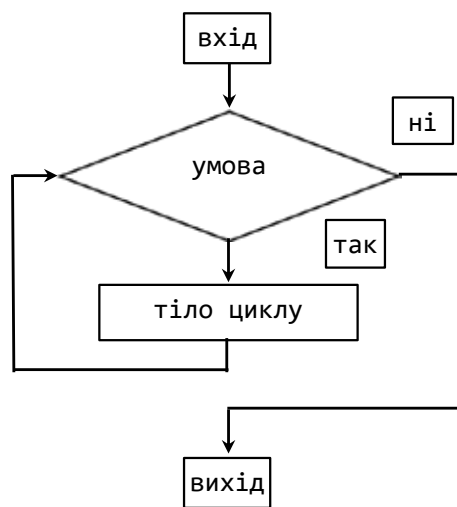


Рис. 5. Керуючий оператор циклу (ЦИКЛ)

Крім того, підхід до їхнього вивчення має бути єдиним. Надалі пропонується називати їх керуючими операторами, оскільки вони керують ходом виконання програми.

Так, у Python є три типи умовних керуючих операторів. Керуючий оператор `if` виконує (вибирає) дію (або групу дій), якщо умова істинна, або пропускає її (їх), якщо умова хибна. Керуючі оператори `if...else` та `if...elif` виконують дію (або групу дій), якщо умова є істинною, і виконують іншу дію (або групу дій), якщо умова хибна. Керуючі оператори `if...elif` та `if...elif...else` можуть перевіряти декілька умов і відповідають оператору `switch` у мові C++.

Керуючі оператори циклу `for` і `while` не виконують дію (чи групу дій), якщо умова продовження циклу хибна, або потрібну кількість дій, якщо умова істинна.

Таким чином, учень повинен чітко уявляти собі, що програма проектується і конструюється з використанням лінійної послідовності опера-

торів *СЛІДУВАННЯ* і керуючих операторів *РОЗВИЛКА* і *ЦИКЛ*, які мають один вхід і один вихід.

При створенні програм слід використовувати принципи спадної розробки (проекткування) програми методом покрокової деталізації. Спадне проектування включає розбиття великої задачі на менші підзадачі, які можуть розглядатися окремо і, у свою чергу, розбиватися на менші підзадачі до з'ясування всіх особливостей основної задачі, що розв'язується.

Основну ідею спадного проектування методом покрокової деталізації добре пояснив Edward Yourdon [24]. Розглянемо її на прикладі будь-якої (довільної) програми, з умовною назвою GLOP. Спочатку спробуємо припустити, що у нас є комп'ютер, який може відразу виконати цю програму. Тоді ми можемо просто написати команду:

GLOP

і все готове! На жаль, поки немає таких комп'ютерів, які б за Вашим бажанням виконували будь-які дії. Тому нам доведеться розбивати завдання на дрібніші елементи, тобто команди які б розумів комп'ютер. Як правило, виконання програми передбачає введення необхідних даних, їх обробку та виведення отриманих результатів. І тут Edward Yourdon упускає важливий момент. Адже перед проектуванням програми потрібно спроектувати для неї структури даних! Тому необхідно переписати нашу програму GLOP у наступному вигляді

Спроектувати структури даних GLOP!

Ввести дані GLOP

Виконати обчислення GLOP

Вивести результати GLOP

При цьому кожен рядок нашої програми ми можемо знову розбивати на простіші операції, доки вони не зможуть бути представленими в операторах мови програмування, що є в нашому розпорядженні (в даному випадку це Python).

Практика викладання показала, що використання блок-схем [25] і псевдокоду [26] дають мало користі оскільки відірвані від програм, тому найкращі результати дають коментарі у програмах. Як показав досвід викладання, процес коментування програми змушує учнів під час її проектування та кодуванні ретельніше продумувати алгоритм та його реалізацію, а згодом – краще розуміти саму програму.

Як приклад, розглянемо програму мовою Python до задачі з роботи [8] з рішення квадратного рівняння $ax^2+bx+c=0$ з дійсними коефіцієнтами a ($a \neq 0$), b , c . Умовимося символом «☞» позначати текст скрипта, який виконується у вікні редактора інтегрованого середовища розробки Python, а символом «☜» – результат роботи цього

скрипта у вікні виконання. Тоді текст програми та результати її виконання будуть виглядати, як показано на рис. 6.

Як видно з прикладу, коментарі практично відповідають алгоритму, що унеможливорює неправильне складання коду програми. Тому на чільне місце необхідно ставити освоєння учнями підходів і навичок конструювання програмних одиниць на основі читання, вивчення та розуміння чужого коду – найважливішого компонента командної розробки ПЗ. Обов'язковим є виділення кожного оператора, що управляє, відступами спереду і ззаду, а також наявність коментаря про його призначення безпосередньо перед ним.

Структура та формат коментарів у пропонуваніх для навчання учнів комп'ютерних програ-

☞	<pre> # Програма рішення квадратного рівняння # Імпортуємо модуль math import math as m # перейменовуємо модуль в m # Друкуємо інформацію про програму print("Програма розв'язує квадратне рівняння виду") print("a*x^2+b*x+c=0 \n") # Вводимо коефіцієнти рівняння a = float(input('Введіть значення a = ')) b = float(input('Введіть значення b = ')) c = float(input('Введіть значення c = ')) D = b * b - 4 * a * c # обчислюємо дискримінант D=b^2-4ac # Знаходимо корені рівняння згідно значення D if (D >= 0): # якщо D більше або рівне 0 x1 = (-b - m.sqrt(D)) / (2 * a) # обчислюємо корінь x1 x2 = (-b + m.sqrt(D)) / (2 * a) # обчислюємо корінь x2 print("Корені рівняння: x1 = ", x1, " x2 = ", x2) # виводимо результат elif (D == 0): # якщо D рівне 0 x = -b / (2 * a) # обчислюємо корінь x print("Рівняння має єдиний корінь: x = ", x) # виводимо результат else: # у інших випадках print("Рівняння не має дійсних коренів!") # виводимо попередження </pre>
☜	<pre> Програма розв'язує квадратне рівняння виду a*x^2+b*x+c=0 Введіть значення a = 1 Введіть значення b = -8 Введіть значення c = 15 Корені рівняння: x1 = 3.0 x2 = 5.0 </pre>

Рис. 6. Програма для розв'язування квадратного рівняння та результат її роботи

мах мають бути засновані на відомому стандарті PEP 8 [27], котрий роз'яснює основні вимоги до оформлення програми мовою Python:

1. Використовуйте 4 пробіли для відступів замість табуляції.
2. Обмежте довжину рядка 79 символами.
3. Розділяйте функції та класи двома порожніми рядками.
4. Використовуйте порожній рядок навколо операторів привласнення та після блоків функцій та класів.
5. Робіть зрозумілі імена змінних та функцій і таке інше.

Також, на наш погляд було б доцільним для учнів програмувати задачі з математики і фізики для кращого поєднання цих знань з інформатикою. Тим більш, що у навчальній програмі з математики (рівень стандарту) для 10–11 класів загальноосвітніх шкіл [28] та фізики [29] однією з ключових є інформаційно-цифрова компетентність!

Висновки. Розглянувши та провівши дослідження сучасних напрямів використання мови Python автори вважають її одною з перспективніших засобів не тільки навчання основам алгоритмізації та програмування, але й ознайомлення учнів з найсучаснішими прикладними напрямками розвитку цифрових технологічних трендів:

- простота мови є одним з головних чинників вважати її однією з доступніших для проникнення до її основ та розширення кола користувачів, включаючи, як учнів різних вікових категорій, так і вчителів різних предметів (математики, фізики, хімії, інформатики тощо);
- стрімкий розвиток поточним часом і широке розповсюдження застосування технологій штучного інтелекту практично в усіх можливих сферах поточного професійного розвитку людини: медицина, бізнес, інженерія, програмування і т.і., розкриває нові напрями застосування мови Python завдяки її спеціалізованим можливостям;
- велика кількість бібліотек (більше 137 000), з якими постачається мова Python, надає багато додаткових можливостей з точки зору як використання, так і вивчення цієї мови;
- велика кількість безкоштовних освітніх курсів, а також масових відкритих онлайн-курсів

(MOOC, Massive Open Online Courses) з вивчення мови Python (Python MOOC and Free Online Courses), що надає широкі можливості розширення та поглиблення знань програмування на цій мові не тільки учням, а й вчителям;

– додатковою важливою можливістю при використанні мови є її мобільність, завдяки якій учні можуть створювати програмний код у мобільних пристроях – смартфонах, які є постійними супутниками сучасної молоді, використовуючи безкоштовні онлайн компілятори;

– сучасним важелем зрушення ситуації навколо вивчення мови Python може надати нещодавнє об'єднання можливостей даної мови з популярним і достатньо розповсюдженим компонентом Microsoft Office – Microsoft Excel, який достатньо широко використовується у навчальному процесі завдяки своїй простоті;

– спираючись на процедурну парадигму Python, при вивченні мови пропонується використовувати єдиний підхід з упором на концептуальні поняття, тобто на структурне програмування й принципи спадної розробки (проектування) програми методом покрокової деталізації;

– практика викладання показала, що використання блок-схем і псевдокоду дають мало користі оскільки відірвані від програм, тому найкращі результати дають коментарі у програмах, оскільки процес коментування програми змушує учнів під час її проектування та кодуванні ретельніше продумувати алгоритм та його реалізацію;

– структура та формат коментарів у пропонованих для навчання учнів комп'ютерних програмах мають бути засновані на відомому стандарті PEP 8, котрий роз'яснює основні вимоги до реалізації програми мовою Python;

– також, на наш погляд було б доцільним для учнів програмувати задачі з математики і фізики для кращого поєднання цих знань з інформатикою, для оволодіння інформаційно-цифровою компетентністю.

Подальші дослідження потрібно направити у бік створення методичних розробок та посібників, не тільки для учнів, а й вчителів! Бо все описане не можна вмістити у підручник з інформатики, де розглядається багато тем суто інформаційної спрямованості.

Список літератури:

1. Скретч (мова програмування). URL: [https://uk.wikipedia.org/wiki/Скретч_\(мова_програмування\)](https://uk.wikipedia.org/wiki/Скретч_(мова_програмування))
2. Мова програмування високого рівня. URL: https://uk.wikipedia.org/wiki/Мова_програмування_високого_рівня
3. Державний стандарт базової середньої освіти. URL: <https://surl.li/glhimb>
4. Вчити програмування треба в школі чи університеті? URL: <https://osvita.ua/school/54063/>

5. Інформатична освітня галузь. Інструктивно методичні рекомендації щодо викладання навчальних предметів / інтегрованих курсів у закладах загальної середньої освіти у 2024/2025 навчальному році (Розділ 9). Додаток до листа МОН від 30.08.2024 1.1/15776-24
6. Інформатична освітня галузь. Модельні навчальні програми. Інститут модернізації змісту освіти. <https://surl.li/bjzejw>
7. Громко Г. Методичні рекомендації щодо викладання інформатики у 2024/2025 навчальному році. <https://surl.li/cjbbmn>
8. Сікора О.В., Кобильник Т.П. Java як засіб навчання учнів основ програмування. *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. 2023. Том 34 (73), № 5. С. 224-230
9. Половина підлітків знає, ким хоче бути в майбутньому. У топі професій – програміст, лікар, дизайнер. URL: <https://www.village.com.ua/village/city/city-news/353791-polovina-pidlitkiv-znae-kim-hoche-buti-u-maybutnomu-u-topi-profesiy-programist-likar-dizayner>
10. Does Coding Help You Think Logically? URL: <https://funtech.co.uk/latest/does-coding-help-you-think-logically>
11. PYPL Popularity of Programming Language. URL: <https://pypl.github.io/PYPL.html>
12. JavaScript. URL: <https://en.wikipedia.org/wiki/JavaScript>
13. “Hello World” In 30 Different Languages. URL: <https://www.geeksforgeeks.org/hello-world-in-30-different-languages/>
14. Java (programming language). URL: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
15. What are the differences between procedural and object-oriented programming? URL: <https://surl.li/baxeod>
16. Difficulties in Understanding Object Oriented Programming Concepts. URL: https://www.researchgate.net/publication/289775172_Difficulties_in_Understanding_Object_Oriented_Programming_Concepts
17. Python (programming language). URL: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
18. Off-side rule. URL: https://en.wikipedia.org/wiki/Off-side_rule
19. What is Python used for? 10 practical Python uses. URL: <https://www.futurelearn.com/info/blog/what-is-python-used-for>
20. Top 10 Python Libraries. URL: <https://www.interviewbit.com/blog/python-libraries/>
21. Felix Zumstein. Python for Excel. A Modern Environment for Automation and Data Analysis. O'Reilly Media. 2021. – 275 p.
22. Korotenko G, Korotenko L. Formation of a Programming Languages Stack and a methodology of teaching to students specialized in Computer Science at Technical Universities in the context of interdisciplinarity. *Technium Sustainability*. Vol. 1, No. 1 (2021): – P. 21 33. Sustainability. Published: 2021-10-07. Retrieved from URL <https://techniumscience.com/index.php/sustainability/article/view/4944> DOI: <https://doi.org/10.47577/sustainability.v1i1.4944>
23. Flow diagrams, turing machines and languages with only two formation rules. URL: <https://dl.acm.org/doi/10.1145/355592.365646>
24. Yourdon E. Techniques of Program Structure and Design. – Prentice Hall, 1975. – 409 p.
25. Block diagram. URL: https://en.wikipedia.org/wiki/Block_diagram
26. Deitel P., Deitel H. C++11 for Programmers. 2nd Edition. – Prentice Hall, 2013. – 1280 p.
27. PEP 8 – Style Guide for Python Code. URL: <https://peps.python.org/pep-0008/>
28. МАТЕМАТИКА (рівень стандарту). Програма для 10-11-х класів ЗНЗ. URL: <https://osvita.ua/school/program/program-10-11/58878/>
29. ФІЗИКА. Навчальні програми для загальноосвітніх навчальних закладів у 10 та 11 класах. URL: <https://surl.li/htqoky>

Korotenko G.M., Sokolova N.O., Shirin A.L. TEACHING IN GENERAL SECONDARY EDUCATION INSTITUTIONS OF PROGRAMMING BASICS USING THE PYTHON LANGUAGE

Currently, the vector of orientation towards the development of students' ideas about the application of algorithmic thinking skills is becoming increasingly complicated due to the sharp change in applied tasks that modern specialists in the IT sphere must solve. In addition to purely applied tasks related to computer or computational experiments with digital twins and other digital models, the volume of various data of various nature is growing tremendously. Such data primarily concerns end-to-end technologies and, including, tasks related to the use of models and artificial intelligence tools. It is clear that providing relevant IT competencies to future students and specialists in digital technologies is an urgent mission of a modern school. But it must be borne in mind that not all students will be professional programmers in the future. Thus, a complex of interrelated tasks arises regarding the formation of theoretical, practical, technological, psychological and some other complex procedures aimed at solving emerging problems. Against the backdrop of an increasing

volume of rapidly complex problems, an important element of improving methodological approaches to improving the level of learning of both students and teachers conducting the educational process is the choice of an appropriate programming language and its organizational environment.

The work proves that gradual, supported by appropriate theoretical and practical (including computer) means of mastering the basics of programming is not only possible, but also necessary. According to the analysis of modern computer science textbooks for middle school, it can be summarized that programming as such is practically absent from practical tasks. Thus, teachers must choose themselves: programming language, teaching methods, means of supporting the educational process, appropriate software tool and its version.

The article provides examples of possible directions for applying the most widespread and easy-to-learn multi-paradigm, high-level general-purpose programming language Python with dynamic strong typing and automatic memory management, aimed at increasing developer productivity, simplifying code reading and improving its quality, as well as ensuring the portability of programs written using it.

Key words: *program code, programming languages, algorithms, learning, Python programming language, structured programming.*