

УДК 37.016:004.43

DOI: 10.31652/2412-1142-2025-76-45-54

Оніщенко Сергій Миколайович

здобувач третього рівня вищої освіти

Український державний університет імені Михайла Драгоманова,

м. Київ, Україна

ORCID ID: 0009-0008-9253-2974

s.m.onishchenko@npu.edu.ua

ПОРІВНЯЛЬНИЙ АНАЛІЗ МОВ ПРОГРАМУВАННЯ ДЛЯ ВИВЧЕННЯ В ШКОЛІ

Анотація. Сучасне суспільство характеризується широким використанням інформаційних технологій у різних галузях діяльності людини. У зв'язку з цим висувуються особливі вимоги до підготовки фахівців з програмування у закладах вищої освіти, і, звичайно, до вивчення програмування у закладах загальної середньої освіти. Важливим компонентом інформатичної підготовки учнів є розвиток операційного, алгоритмічного мислення, засвоєння основних понять мови програмування, структур даних, етапів створення програмних засобів. Мова програмування є основним засобом вивчення елементів програмування. Тому, успішність оволодіння учнями основами алгоритмізації і програмування великою мірою визначається тим, наскільки вдало вчителем обрано мову програмування із запропонованого в чинних методичних рекомендаціях МОН України переліку мов для вивчення програмування. Здійснено аналіз науково-методичної літератури з проблеми дослідження, шкільних підручників і навчальних посібників з інформатики; використано систематизацію й узагальнення вітчизняного і зарубіжного досвіду, узагальнення власного досвіду. Розглянуто й досліджено перелік основних загальних та синтаксичних характеристик мови програмування, які варто врахувати при обранні мови для вивчення програмування в закладах загальної середньої освіти. Наведено програмні реалізації різними мовами програмування демонстраційного прикладу з детальними поясненнями і уточненнями структурних особливостей і застосованих мовних засобів. Розроблено рекомендації щодо обрання мови програмування для вивчення основ програмування в класах другого циклу базової середньої освіти, а також для поглибленого вивчення програмування в класах профільної середньої освіти. Проведений аналіз загальних характеристик, синтаксичних конструкцій структури програм і основних управляючих інструкцій вказує на те, що мови Python, C++, C# і Java можуть бути обрані для вивчення програмування. Але, з огляду на те, що мова Python має суттєві відмінності в синтаксисі, можна рекомендувати її для вивчення основ програмування в класах другого циклу базової середньої освіти, а для поглибленого вивчення програмування в класах профільної середньої освіти може бути використана мова програмування C#.

Ключові слова: програмування; мова; характеристики; вибір мови; навчання; освітній процес; середня загальна освіта; школа.

1. ВСТУП

Постановка проблеми. На вирішення питання про вибір мови програмування для вивчення в закладах загальної середньої освіти впливають багато факторів, зокрема характеристики мов програмування. В класах другого циклу базової середньої освіти (7-9 класи) за методичними рекомендаціями МОН України для вивчення програмування пропонуються на вибір кілька мов програмування: Object Pascal, Visual Basic, Python, Java, C#, C++. Запропонований перелік мов програмування базується на тому, що «основною вимогою до вибору мови програмування у 8 класі є підтримка в ній базових концепцій об'єктно-орієнтованого програмування». В 10-11 класах «рекомендується опановувати одну з професійних мов програмування, наприклад C++, Java або Python, незалежно від того, на якій мові учні вчилися програмувати в основній школі» [1]. У затверджених МОН України підручниках матеріал з алгоритмізації й програмування базується на вивченні мов програмування Object Pascal (середовище Lazarus) або Python (середовище IDLE).

Таким чином, перед учителем постає важлива проблема вибору мови програмування, що має ґрунтуватися на певних факторах, що формують вимоги до обраної мови програмування. Для подальшого розгляду та аналізу характеристик мов програмування слід

обрати наступний перелік рекомендованих МОН мов програмування: Python, C++, C#, Java. Мови Object Pascal та Visual Basic на сьогодні є малопоширеними, і не користуються попитом на ринку праці [2].

Аналіз останніх досліджень і публікацій. Над дослідженням основних рис та характеристик сучасних мов програмування, що пропонуються як засіб для вивчення програмування у закладах загальної середньої освіти, працювали багато українських та зарубіжних вчених, зокрема: Біляй Ю. П., Вернигоренко С. А., Глинський Я. М., Горошко Ю. В., Гришко Л. В., Жалдак М. І., Жуковський В. С., Коротун О. В., Костюченко А. О., Лапінський В. В., Мельник В. І., Мінтій І. С., Міхеєв В. В., Міца О. В., Морзе Н. В., Палюшок Л. В., Ракута В. М., Рамський Ю. С., Руденко В. Д., Сейдаметова З. С., Семеріков І. О., Страуструп Б., Теплицький О. І., Троелсен Е., Умрик М. А., Цибко Г. Ю., Шевчук П. Г., Шилдт Г., Ящик О. Б. та інші.

Основні змістові лінії шкільного курсу інформатики та вимоги до компетентностей учнів, зокрема з програмування, наведено в Державному стандарті загальної середньої освіти [3] та в модельних навчальних програмах [4, 5]. В 2024 році було затверджено державний стандарт профільної середньої освіти [6]. В ньому окреслено вимоги до підготовки учнів в профільній школі. Тенденції становлення інформатики як шкільного предмета та місце програмування в ньому розглянуто в праці [7]. Проте, залишаються недостатньо розробленими питання методики навчання окремих тем шкільного курсу інформатики в сучасній українській школі в умовах інформаційного суспільства.

З урахуванням тенденцій розвитку програмної інженерії, процесу реформування шкільної освіти, розробки та прийняття нових державних стандартів повної загальної середньої освіти, потребують подальших досліджень питання якісного навчання основ програмування в школі. При цьому дуже важливим питанням залишається вибір мови для навчання програмування у закладах загальної середньої освіти.

Мета статті є визначення та обґрунтування переліку основних характеристик мов програмування, що полегшить вчителю процес вибору мови для вивчення програмування в школі.

2. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У мовах програмування, що можуть бути використані для навчання, слід виокремити для аналізу такі основні загальні характеристики: призначення мови, засоби мови для підтримки сучасних парадигм програмування, поширеність мови, особливості синтаксичних конструкцій. Розкриваючи при цьому питання призначення мови, варто наголосити на умовному поділі на мови загального (універсальні) і спеціального призначення.

Універсальні мови програмування можна використовувати для широкого спектру завдань і областей програмування. Такі мови мають різноманітні функції та засоби, що дають змогу ефективно використовувати їх у багатьох сферах.

Поширеність та рейтинг мови програмування визначається кількістю програмістів [8], які використовують цю мову для розробки програмного забезпечення, а також кількістю проєктів та обсягом ринку, в яких вона застосовується.

Мови програмування, що проєктувалися і створювалися для навчання програмування (Basic, Logo, Pascal, Scratch, Blockly, НАМ, Рапіра), мають доступніший для сприйняття і розуміння синтаксис [9]. Але це досягається за рахунок обмеження засобів мови та зменшення ефективності алгоритмічних конструкцій, що можуть бути реалізовані мовою програмування.

З часом розробники мов програмування продовжують їх розвивати та вдосконалювати. І цей процес передбачає розробку нових засобів, успадковування та привнесення засобів з покращеними характеристиками, синтаксичними особливостями з інших мов програмування. Аналізуючи мови програмування, що традиційно використовуються для навчання, можна виокремити такі поширені різновиди синтаксису:

– синтаксис мови Basic (QuickBasic, Visual Basic, VB.NET);

- синтаксис мови Pascal (ObjectPascal, Delphi);
- синтаксис мови C (C++, C#, Java, Kotlin, JavaScript).

Мовами програмування, певною мірою, можна вважати набори та правила використання команд управління виконавцями в навчальних середовищах «Сходинки до інформатики» та «Скарбниця знань». Кожна з навчальних мов, зазвичай, має свій специфічний синтаксис та набір команд.

Відаючи перевагу тому чи іншому синтаксису, варто враховувати, що, здебільшого, синтаксис мов подібний до синтаксису мови C дещо складніший. Проте основним аргументом на користь використання у навчанні C-подібного синтаксису є його розповсюдженість, решта різновидів синтаксису мов програмування не настільки широко використовується в професійній діяльності. Проблема переходу з навчальної на професійну мову програмування є актуальною для більшості фахівців з розробки програмного забезпечення [10]. Використання в навчанні мов програмування з C-подібним синтаксисом дозволяє практично уникнути проблем такого переходу.

Виокремимо в таблицю загальні характеристики сучасних мов програмування Python, C++, C#, Java (див. Таблицю 1).

Таблиця 1.

Загальні характеристики мов програмування

Характеристика	Python	C++	C#	Java
Кросплатформність	Так	Так	Так	Так
Тип системи програмування	Динамічна типізація	Сильна статична типізація	Сильна статична типізація	Сильна статична типізація
Об'єктно-орієнтоване програмування	Повна підтримка	Повна підтримка	Повна підтримка	Повна підтримка
Функціональне програмування	Має підтримку, але не є основною парадигмою	Часткова підтримка	Часткова підтримка	Часткова підтримка
Асинхронне програмування	Підтримка засобів, таких як асинхронні функції та бібліотеки	Часткова підтримка	Повна підтримка	Повна підтримка
Інтерфейси	Неявні інтерфейси (протоколи)	Чисті віртуальні функції та інтерфейси	Інтерфейси та делегати	Інтерфейси та абстрактні класи
Спосіб виконання коду	Інтерпретація або компіляція до байт-коду (з використанням інтерпретатора або JIT-компіляції)	Компіляція до машинного коду, найбільше розповсюджена форма виконання	Компіляція до мови CLR	Компіляція до байт-коду (з використанням JVM)
Механізм керування пам'яттю	Автоматично (збірник сміття)	Вручну (з використанням вказівників)	Автоматично (збірник сміття)	Автоматично (збірник сміття)
Підтримка багатозадачності	Так	Так	Так	Так
Використання	Веб-розробка, аналітика даних, штучний інтелект, наукові обчислення, системне програмування	Системне програмування, ігри, вбудовані системи, наукові дослідження	Веб-розробка, мобільні додатки, настільні додатки, графіка	Веб-розробка, мобільні додатки, настільні додатки, корпоративні додатки

Для обрання мови для вивчення програмування суттєве значення мають такі загальні характеристики: тип системи програмування (сильна статична типізація), механізм керування пам'яттю, підтримка парадигм програмування (об'єктно-орієнтована парадигма за методичними рекомендаціями МОН), і сфери використання (для розробки веб-, мобільних та

настільних додатків з графічним інтерфейсом). Сильна статична типізація даних вимагає перед записом алгоритму мовою програмування передбачити та заздалегідь описати й оголосити потрібні програмні об'єкти (типи даних користувача, класи, константні та змінні величини) перед їхнім застосуванням. Такі вимоги до побудови програми дисциплінують процес розробки програмних засобів та забезпечують від прийняття необачних рішень.

Керування пам'яттю в об'єктно-орієнтовних мовах Python, C#, Java покладається на віртуальні машини, що мають збирачі «сміття». Це спеціальні підпрограми, які періодично сканують оперативну пам'ять і вивільнюють ті її фрагменти, на які не має посилань. Використання вказівників притаманне в явному вигляді із запропонованого списку мов лише мові C++. Вказівники є ефективним засобом для роботи з пам'яттю та даними. Наприклад, вказівники використовуються для виділення пам'яті для розміщення потрібних даних за допомогою оператора *new*, та звільнення використаної пам'яті за допомогою оператора *delete*; для створення динамічних структур даних, таких як зв'язні списки, дерева, графи тощо; вказівник можна передати до функції як параметр, або отримати з функції його значення, що особливо корисно для функцій, які потребують модифікації значень або повертають більше одного значення; вказівники дають можливість виконувати додаткові операції з пам'яттю, такі як арифметика вказівників, використання вказівників на функції, робота з динамічними масивами, управління ресурсами. Звичайно, для досвідченого програміста ці засоби є потужним інструментом для створення ефективних програмних засобів [11], але для початківців-школярів ця тема є занадто складною для розуміння та вивчення.

Сформуємо таблицю характеристик синтаксичних конструкцій мов програмування Python, C++, C#, Java (див. Таблицю 2).

Таблиця 2.

Характеристики синтаксичних конструкцій мов програмування

Конструкція	Python	C++	C#	Java
Структура програми	Імпорт модулів, пакетів. Визначення функцій та класів. Головна функція має блоки коду.	Використання класів, функцій, заголовних файлів.	Використання класів, об'єктів. Концепція простору імен.	Основна структура – класи та методи, програма починається з методу <i>main</i> .
Оголошення змінних і констант	<code>x = 5</code> <code>y = 10</code> (тип змінних і констант визначається автоматично)	<code>int x;</code> <code>const int y = 10;</code>	<code>int x;</code> <code>const int y = 10;</code>	<code>int x;</code> <code>final int y = 10;</code>
Стандартні типи даних	Динамічна типізація: <code>int</code> , <code>float</code> , <code>str</code> , <code>list</code> , <code>tuple</code> , <code>set</code> , <code>dict</code>	<code>short</code> , <code>int</code> , <code>long</code> , <code>long long</code> , <code>float</code> , <code>double</code> , <code>char</code> , <code>wchar_t</code> , <code>bool</code> , вказівники <code>T*</code>	<code>sbyte</code> , <code>byte</code> , <code>short</code> , <code>ushort</code> , <code>int</code> , <code>uint</code> , <code>long</code> , <code>ulong</code> , <code>float</code> , <code>double</code> , <code>decimal</code> , <code>char</code> , <code>bool</code>	<code>byte</code> , <code>short</code> , <code>int</code> , <code>long</code> , <code>float</code> , <code>double</code> , <code>char</code> , <code>boolean</code>
Оператори присвоєння	<code>x = 5;</code> <code>y = x</code>	<code>x = 5;</code> <code>y = x;</code>	<code>x = 5;</code> <code>y = x;</code>	<code>x = 5;</code> <code>y = x;</code>
Умовний оператор	<code>if</code> умова: оператори <code>elif</code> умова: оператори <code>else</code> : оператори	<code>if</code> (умова) {оператори;} <code>else</code> {оператори;}	<code>if</code> (умова) {оператори;} <code>else</code> {оператори;}	<code>if</code> (умова) {оператори;} <code>else</code> {оператори;}
Оператор вибору	відсутній	<code>switch</code> (вираз){ <code>case</code> значення: оператори; <code>break</code> ; ... <code>default</code> : оператори; }	<code>switch</code> (вираз){ <code>case</code> значення: оператори; <code>break</code> ; ... <code>default</code> : оператори; }	<code>switch</code> (вираз){ <code>case</code> значення: оператори; <code>break</code> ; ... <code>default</code> : оператори; }

Тернарний оператор	значення1 if умова else значення2	умова?значення1: значення2	умова?значення1: значення2	умова?значення1: значення2
Оператори повторення	while for in	while do-while for () for in	while do-while for () for in	while do-while for () for in
Робота з масивами	Вбудований тип list, який може містити значення різних типів.	Масиви однотипних даних можуть бути статичними або динамічними.	Масиви та колекції .NET.	Масиви та колекції Java.
Робота з текстовими рядками	Рядки – це об'єкти з відповідними методами	Використовується бібліотека string	Використовуються об'єкти string та рядкові операції .NET.	Використовується клас String та операції Java для роботи з рядками.
Робота з структурами	Немає окремого службового слова для структур.	Використовується службове слово struct.	Використовується службове слово struct.	Використовується службове слово class.
Робота з файлами	функції open, read, write.	потоків введення / виведення iostream, fstream, istream, ostream	класи FileStream, StreamReader, StreamWriter.	класи File, BufferedReader, BufferedWriter.

Для наочності розглянемо структури програм, написаних різними мовами програмування, на прикладі простої програми для обчислення площі круга за вказаним радіусом (див. Таблицю 3).

Таблиця 3.

Коди програми різними мовами програмування (авторська розробка)

Python

```
# 1. Імпорт бібліотеки математичних функцій
import math
# 2. Опис призначення змінних
# radius - радіус круга
# area - площа круга
# text - для введення текстових даних
# 3. Введення даних
text = input("Ввести радіус круга: ")
radius = float(text)
# 4. Обчислення результату
area = math.pi * radius**2
# 5. Виведення результату
print(f"Площа круга з радіусом {radius} = {area}")
```

C++

```
// 1. під'єднати бібліотеки введення/виведення
// та математичних функцій
#include <iostream>
#include <cmath>
using namespace std;
// 2. Оголошення головної функції - точки входу
int main()
{
    // 3. Оголошення змінних
    double radius; // радіус круга
    double area; // площа круга
    // 4. Введення даних
    cout << "Ввести радіус круга: ";
    cin >> radius;
    // 5. Обчислення результату
    area = M_PI * pow(radius, 2);
    // 6. Виведення результату
    cout << "Площа круга з радіусом " << radius << " = " << area << endl;
    // 7. Вихід з програми
    return 0;
}
```

C# 10.0

```
// 1. Оголошення змінних
double radius; // радіус круга
double area; // площа круга
string input; // для введення текстових даних
// 2. Введення даних
Console.WriteLine("Введіть радіус круга: ");
input = Console.ReadLine();
radius = Convert.ToDouble(input);
// 3. Обчислення результату
area = Math.PI * Math.Pow(radius, 2);
// 4. Виведення результату
Console.WriteLine($"Площа круга з радіусом {radius} = {area}");
```

Java

```
// 1. під'єднати клас Scanner з пакету java.util
import java.util.Scanner;
// 2. Оголошення класу CircleArea
public class CircleArea
{
    // 3. Оголошення головної функції main - точки входу
    public static void main(final String[] args)
    {
        // 4. Оголошення змінних
        Scanner scanner; // об'єкт для введення з клавіатури
        double radius; // радіус круга
        double area; // площа круга
        // 5. Введення даних
        scanner = new Scanner(System.in);
        System.out.print("Введіть радіус круга: ");
        radius = scanner.nextDouble();
        scanner.close();
        // 6. Обчислення результату
        area = Math.PI * Math.pow(radius, 2);
        // 7. Виведення результату
        System.out.println("Площа круга з радіусом " + radius + " одиниць: " + area);
    }
}
```

Проаналізуємо запропоновані коди програм з погляду структурної будови. Програми мовами C++, Java мають описи головної функції `main()`, а у мовах C# і Java – ще й описи класу, що вимагається стандартами проєктів для кожної мови. Крім того, мова Java має ще одну особливість – назва файлу, що містить клас з головною функцією (`main`) має співпадати з назвою цього класу. Прості програми, написані мовою Python, не потребують спеціальних конструкцій ні для головної функції, ні для класу, а тому мають простий і привабливий вигляд. Це є одним з головних аргументів при виборі саме мови Python для вивчення програмування у школі.

У мовах C# і Java для консольного введення/виведення потрібно використовувати методи відповідних класів – це обумовлено тим, що ці мови є об'єктно-орієнтованими. Мова Python також об'єктно-орієнтована, але використовує для цього вбудовані функції `print()` та `input()`. Насправді, ці функції (як і багато інших) належать модулю «builtins», який автоматично імпортується у кожний файл Python, і тому викликати `input()` можна у будь-якому місці програми без додаткового імпорту або нотації належності класу. У програмі мовою C++ для введення/виведення використовуються потокові об'єкти `cin` і `cout` з простору імен `std`, які створюються при роботі з консольним додатком.

У коді програми мовою C# версії 10.0 всі базові описи (підключення потрібних просторів імен, оголошення класу і головної функції для точки входу) можна не використовувати, що значно спрощує вивчення мови для початківців [12]. Така програма за своєю простотою нагадує варіант програми мовою Python.

Звичайно, синтаксичні конструкції структури програм впливають на сприйняття, розуміння та опанування початкових знань з програмування. Так, зайві і, на початковому етапі, практично зовсім необов'язкові для вивчення, такі поняття, як створення функцій і класів користувача, будуть лише відволікати від розгляду та ознайомлення з такими фундаментальними поняттями мови, як змінні, константи, типи даних, вирази, операції, операнди, оператори та інше [13]. Тому природним є вибір саме тих мов для вивчення програмування, які мають прості синтаксичні конструкції структури програм.

Разом з тим, потрібно також звертати увагу на організацію даних у пам'яті. У мовах програмування C++, C# і Java при оголошенні змінних величин вказується їхній тип даних, що при компіляції надає інформацію про потрібний обсяг пам'яті для розміщення відповідних значень, а також про набір операцій, що можна виконувати над такими значеннями, і нарешті про адресу розміщення їх у пам'яті. При виконанні оператора присвоєння, наприклад, `n = 4`, значення у комірці пам'яті за адресою, на яку вказує назва змінної `n`, замінюється на значення 4. При цьому комірка пам'яті з новим значенням залишається за тією ж адресою. У мові Python така організація даних виконується не для всіх типів даних, а лише для, так званих, «змінюваних» типів даних, наприклад для списків і словників. Скалярні типи даних (числові, логічні), а також текстові – належать до «незмінюваних» типів даних. І тому, при виконанні того ж оператора присвоєння `n = 4`, у вільній доступній області оперативної пам'яті розміщується значення 4, і адреса цього нового місця закріплюється за назвою змінної `n`. При цьому, значення, на яке вказувала змінна `n` до виконання оператора присвоєння, стає недоступним. Отже, навіть при ознайомленні з початковими поняттями мови потрібно звертати уваги не лише на зовнішню простоту запису синтаксичних структур, а й на внутрішню будову таких структур [14, 15], так як лише чітке розуміння таких механізмів дає можливість створювати якісні, ефективні програмні засоби і убезпечить від помилкових рішень.

Важливим при виборі мови для навчання програмуванню є аналіз набору управляючих конструкцій – умовних операторів, та операторів повторення. До умовних операторів зазвичай відносять умовний оператор (або оператор альтернативи), оператор вибору та тернарний оператор. У мовах C++, C# і Java всі ці оператори однакові і за складом і за синтаксисом. Однак у мовах C# і Java оператор вибору `switch` також можна використовувати як вираз. В таблиці 4 подано приклад, за яким змінній рядкового типу `mark` надається текстова інтерпретація числової оцінки.

Використання оператора switch. Джерело: авторська розробка.

<i>switch як інструкція</i>	<i>switch як вираз</i>
<pre>int number = 5; string mark = ""; switch (number) { case 3: mark = "задовільно";break; case 4: mark = "добре"; ; break; case 5: mark = "відмінно"; ; break; default: mark = "незадовільно"; ; break; };</pre>	<pre>int number = 5; string mark = number switch { 3 => "задовільно", 4 => "добре", 5 => "відмінно", _ => "незадовільно" };</pre>

У мові Python оператора вибору не має, але його можна замінити різновидом оператора if.

У мовах C++, C# і Java набір операторів повторення є майже однаковим і за складом і за синтаксисом: три звичайні оператори повторення – while(), do while(), for(), а також оператор перебору значень послідовності (колекції, контейнера або діапазону) for in. У мові C++ оператор перебору значень з'явився у версії C++ 11 і має вигляд for (змінна : колекція). У мові Python є лише один оператор повторення while і оператор перебору значень послідовності for in.

У деяких навчальних посібниках і підручниках вивчення інструкцій управління розпочинають не з умовних операторів, а саме з оператора повторення з лічильником for. Це зумовлено тим, що автори цих посібників керувалися такою логікою: при перелічуванні предметів не потрібно аналізувати ніяких умов, просто потрібно ці предмети нумерувати. І коли у процесі нумерації отримується наперед задане значення, то цей процес має зупинитися. Але в такому міркуванні все одно «заховане» порівняння: як визначити, «коли у процесі нумерації отримується наперед задане значення?». Коли номер озвучується, або записується на папері, або загинається відповідний палець, або кладеться на кучку чергова паличка чи камінчик – виконується розумова операція порівняння названого числа з тим, що було запам'ятовано, або десь записано, – тобто виконується обчислення логічного виразу, або простіше можна сказати – «перевіряється виконання умови».

Але оператор повторення for у мовах програмування з C-подібним синтаксисом не є насправді «чистим» оператором з лічильником. За синтаксисом оператора for другим його параметром є логічний вираз. Наприклад, для виведення у рядок перших 10 натуральних чисел, відокремлених одне від одного пропусками, потрібно написати мовою C++ такий фрагмент:

```
for (int i = 1; i < 11; i++){
    cout << i << " ";
}
cout << endl;
```

Отже, другий параметр оператора for має вигляд «i<11». А це є логічний вираз, який потрібно обчислювати при кожній ітерації циклу. Мовою Python цей фрагмент буде виглядати так:

```
for i in range(1,11):
    print(i, end = ' ')
print()
```

Функція range(1,11) створює послідовність чисел від 1 до 10, а змінна i по черзі набуває цих значень. Ітерації закінчуються у випадку, коли змінній i буде надано значення 10 і виконуються оператори тіла циклу. Насправді, при формуванні діапазону виконується неявне порівняння поточного значення з числом 11. Для такого використання оператора for потрібно окремо розібратися з роботою функції range(), що може мати три параметри, які в свою чергу можуть набувати від'ємних значень.

3. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Опис мов програмування, що відображає їхні властивості, особливості, функціональність, можна виразити через загальні характеристики та характеристики синтаксичних конструкцій мови програмування. При обранні мови для вивчення програмування в закладах загальної середньої освіти потрібно враховувати:

- *загальні характеристики мови*: тип системи програмування, механізм керування пам'яттю, підтримка парадигм програмування, сфери використання;
- *характеристики синтаксичних конструкцій*: для опису структури програми; оголошення змінних і констант; опрацювання даних стандартних типів; використання оператора присвоєння; наявність та використання умовних операторів, операторів повторення, засобів для опрацювання масивів даних, текстових даних, структур та файлів даних.

Враховуючи аналіз загальних характеристик і синтаксичних конструкцій структури програм і основних управляючих інструкцій запропонованих мов програмування Python, C++, C# і Java, можна зробити висновок, що всі ці мови можуть бути обрані для вивчення програмування. Але, з огляду на те, що мова Python має суттєві відмінності, і в загальних характеристиках, і особливо в синтаксисі, можна рекомендувати її для вивчення основ програмування в класах другого циклу базової середньої освіти, а для поглибленого вивчення програмування в класах профільної середньої освіти може бути використана об'єктно-орієнтована, зі строгою статичною типізацією, мова програмування C#.

У подальших дослідженнях варто розглянути й дослідити характеристики середовищ програмування, за якими їх слід обирати як засіб для вивчення основ програмування у закладах загальної середньої освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Лист Міністерства освіти і науки України «Методичні рекомендації щодо викладання навчальних предметів у закладах загальної середньої освіти у 2019/2020 навчальному році» № 1/11-5966 (2019). URL: <https://osvita.ua/doc/files/news/650/65024/Informatika.docx>
- [2] Горошко, Ю.В. & Цибко, Г.Ю. (2017). Про перспективи використання мови Swift у навчанні програмування. Науковий часопис НПУ імені М.П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання (19), 59-63. URL: <https://sj.edu.ua/index.php/kosn/article/view/9/10>
- [3] Державний стандарт базової середньої освіти (2020). URL: <https://www.kmu.gov.ua/storage/app/uploads/public/5f7/5e6/b1e/5f75e6b1ee0d8989401323.doc>
- [4] Модельні навчальні програми для 5–9 класів Нової української школи. URL: <https://mon.gov.ua/osvita-2/zagalna-serednya-osviti/osvitni-programi/modelni-navchalni-programi-dlya-5-9-klasiv-novoi-ukrainskoi-shkoli-zaprovadzhuyutsya-poetapno-z-2022-roku>
- [5] Завадський, І.О., Коршунова, О.В., & Твердохліб, І.А. (2023). Інформатика. 7-9 класи: Модельна навчальна програма. URL: <https://lib.iitta.gov.ua/id/eprint/739267>
- [6] Державний стандарт профільної середньої освіти (2024) URL: <https://zakon.rada.gov.ua/laws/show/851-2024-%D0%BF#Text>
- [7] Твердохліб, І.А. (2024) Дослідження шляхів упровадження предмету «Інформатика» в профільну середню освіту. Wydawnictwo: MANS w Łomży, m. Łomża, Polska - m. Kijów, Ukraina. URL: <https://lib.iitta.gov.ua/id/eprint/744847/>
- [8] Ящик, О.Б. (2016). Методика навчання алгоритмізації та програмування старшокласників на рівні поглибленого вивчення інформатики [Дис. канд. пед. наук, НПУ імені М.П. Драгоманова] Репозитарій НПУ імені М.П. Драгоманова URL: <https://enpuir.npu.edu.ua/handle/123456789/40991>
- [9] Морзе, Н.В. (2004). Методика навчання інформатики. (Ч. 4). Навчальна книга.
- [10] Жуковський, В.С. & Коротун, О.В. (2014). Про перспективу введення мови програмування C++ в навчальний процес загальноосвітніх навчальних закладів. Комп'ютер у школі та сім'ї (113), 23-25. URL: http://nbuv.gov.ua/UJRN/komp_2014_1_6
- [11] Шевчук, П.Г. (2013). Методика навчання програмування учнів класів технологічного профілю на основі використання мови C# [Дис. канд. пед. наук, НПУ імені М.П. Драгоманова] Репозитарій НПУ імені М.П. Драгоманова URL: <https://enpuir.npu.edu.ua/handle/123456789/32016>
- [12] W3SchoolsUA. українською. URL: <https://w3schoolsua.github.io/cs/index.html#gsc.tab=0>

- [13] Умрик, М.А., & Біляй, Ю.П. (2014). Використання технологій дистанційного навчання в процесі навчання сучасних мов програмування. Інформаційні технології і засоби навчання, (41, вип. 3), 218-231. URL: http://nbuv.gov.ua/UJRN/ITZN_2014_41_3_23
- [14] Гришко, Л.В. (2009). Методична система навчання основ програмування майбутніх інженерів-програмістів [Автореф. дис. канд. пед. наук, НПУ імені М.П. Драгоманова] Репозитарій НПУ імені М.П. Драгоманова URL: <https://enpuir.npu.edu.ua/handle/123456789/30842?show=full>
- [15] Цибко, Г.Ю., Горошко, Ю.В. & Костюченко, А.О. (2022). Програмування у Python. Практичний курс. ФОП Баликіна С.М.

COMPARATIVE ANALYSIS OF PROGRAMMING LANGUAGES FOR STUDYING AT SCHOOL

Onishchenko Serhii Mykolaiovych

Phd Student of the Department of Information Technologies and Programming

Dragomanov Ukrainian State University,

Kiyv, Ukraine

ORCID ID: 0009-0008-9253-2974

s.m.onishchenko@npu.edu.ua

Abstract. Modern society is characterized by the widespread use of information technology in various fields of human activity. In this regard, there are special requirements for the training of programming specialists in higher education institutions, and, of course, for the study of programming in general secondary education institutions. An important component of students' computer science training is the development of operational and algorithmic thinking, mastering the basic concepts of programming languages, data structures, and the stages of software development. A programming language is the main means of learning programming elements. Therefore, the success of students' mastery of the basics of algorithmisation and programming is largely determined by how well the teacher chooses a programming language from the list of languages for learning programming proposed in the current methodological recommendations of the Ministry of Education and Science of Ukraine. The article analyses the scientific and methodological literature on the research problem, school textbooks and computer science textbooks; systematisation and generalisation of domestic and foreign experience and generalisation of own experience were used. The list of basic general and syntactic characteristics of a programming language, which should be taken into account when choosing a language for studying programming in general secondary education institutions, is considered and studied. The software implementations of the demonstration example in different programming languages are presented with detailed explanations and clarifications of the structural features and language tools used. Recommendations for choosing a programming language for studying the basics of programming in the classes of the second cycle of basic secondary education, as well as for in-depth study of programming in the classes of specialised secondary education, are developed. The analysis of general characteristics, syntactic structures of the programme structure and basic control instructions shows that Python, C++, C# and Java can be chosen for learning programming. But, given that Python has significant differences in syntax, it can be recommended for learning the basics of programming in classes of the second cycle of basic secondary education, and C# can be used for in-depth study of programming in classes of specialised secondary education.

Keywords: language; characteristics; language choice; learning; educational process; secondary education; school.

REFERENCES (TRANSLATED AND TRANSLITERATED)

- [1] Lyst Ministerstva osvity i nauky Ukrainy «Metodychni rekomendatsii shchodo vykladannia navchalnykh predmetiv u zakladykh zahalnoi serednoi osvity u 2019/2020 navchalnomu rotsi» № 1/11-5966 (2019). URL: <https://osvita.ua/doc/files/news/650/65024/Informatika.docx> (in Ukrainian)
- [2] Horoshko, Yu. V. & Tsybko, H. Yu. (2017). Pro perspektyvy vykorystannia movy Swift u navchanni prohrumuvannia. Naukovyi chasopys NPU imeni M.P. Drahomanova. Seriiia 2: Kompiuterno-oriiientovani systemy navchannia (19), 59-63. URL: <https://sj.udu.edu.ua/index.php/kosn/article/view/9/10> (in Ukrainian)
- [3] Derzhavnyi standart bazovoi serednoi osvity (2020). URL: <https://www.kmu.gov.ua/storage/app/uploads/public/5f7/5e6/b1e/5f75e6b1ee0d8989401323.doc> (in Ukrainian)
- [4] Modelni navchalni prohramy dlia 5–9 klasiv Novoi ukrainskoi shkoly. URL: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programi/modelni-navchalni-programi-dlya-5-9-klasiv-novoi-ukrainskoi-shkoli-zaprovadzhuyutsya-poetapno-z-2022-roku> (in Ukrainian)
- [5] Zavadskiy, I. O., Korshunova, O. V., & Tverdokhlib, I. A. (2023). Informatyka. 7-9 klasy: Modelna navchalna prohrama. URL: <https://mon.gov.ua/storage/app/media/zagalna%20ser...> (in Ukrainian)

- [6] Derzhavnyi standart profilnoi serednoi osvity (2024) URL: <https://zakon.rada.gov.ua/laws/show/851-2024-%D0%BF#Text> (in Ukrainian)
- [7] Tverdokhlib, I. A (2024) Doslidzhennia shliakhiv uprovozhennia predmetu «Informatyka» v profilnu seredniu osvitu. Wydawnictwo: MANS w Łomży, m. Łomża, Polska - m. Kijów, Ukraina. URL: https://mans.edu.pl/fcp/iOEUFzs9BjEkLTg1Y1BSe0N_Y... (in Ukrainian)
- [8] Iashchuk, O. B. (2016). Metodyka navchannia alhorytmizatsii ta prohramuvannia starshoklasnykiv na rivni pohlyblenoho vyvchennia informatyky [Dys. kand. ped. nauk, NPU imeni M.P. Drahomanova] Repozytarii NPU imeni M.P. Drahomanova URL: <https://enpuir.npu.edu.ua/handle/123456789/40991> (in Ukrainian)
- [9] Morze, N. V. (2004). Metodyka navchannia informatyky. (Ch. 4). Navchalna knyha. (in Ukrainian)
- [10] Zhukovskiy, V. S. & Korotun, O. V. (2014). Pro perspektyvu vvedennia movy prohramuvannia S++ v navchalnyi protses zahalnoosvitnikh navchalnykh zakladiv. Kompiuter u shkoli ta simi (113), 23-25. URL: http://nbuv.gov.ua/UJRN/komp_2014_1_6 (in Ukrainian)
- [11] Shevchuk, P. H. (2013). Metodyka navchannia prohramuvannia uchniv klasiv tekhnolohichnoho profilu na osnovi vykorystannia movy C# [Dys. kand. ped. nauk, NPU imeni M.P. Drahomanova] Repozytarii NPU imeni M.P. Drahomanova URL: <https://enpuir.npu.edu.ua/handle/123456789/32016> (in Ukrainian)
- [12] W3SchoolsUA.ukrainskoiu. URL: <https://w3schoolsua.github.io/cs/index.html#gsc.tab=0> (in Ukrainian)
- [13] Umryk, M. A., & Bilai, Yu. P. (2014). Vykorystannia tekhnolohii dystantsiinoho navchannia v protsesi navchannia suchasnykh mov prohramuvannia. Informatsiini tekhnolohii i zasoby navchannia, (41, vyp. 3), 218-231. URL: http://nbuv.gov.ua/UJRN/ITZN_2014_41_3_23 (in Ukrainian)
- [14] Hryshko, L. V. (2009). Metodychna systema navchannia osnov prohramuvannia maibutnikh inzheneriv-prohramistiv [Avtoref. dys. kand. ped. nauk, NPU imeni M.P. Drahomanova] Repozytarii NPU imeni M.P. Drahomanova URL: <https://enpuir.npu.edu.ua/handle/123456789/30842?show=full> (in Ukrainian)
- [15] Tsybko, H. Yu., Horoshko, Yu. V. & Kostiuhenko, A. O. (2022). Prohramuvannia u Python. Praktychnyi kurs. FOP Balykina S.M. (in Ukrainian)