

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
Факультет комп'ютерних наук, фізики та математики  
Кафедра інформатики, програмної інженерії  
та економічної кібернетики

**МЕТОДИЧНІ ОСОБЛИВОСТІ ДОБОРУ І ВИКОРИСТАННЯ МОВ  
ПРОГРАМУВАННЯ ПРИ ВИВЧЕННІ РОЗДІЛУ «АЛГОРИТМИ  
ТА ПРОГРАМИ» У 8-9 КЛАСАХ**

**Кваліфікаційна робота (проект)**

на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 4 курсу, групи 15-432  
Спеціальності 014 Середня освіта (Інформатика)  
Освітньо-професійна програма  
Середня освіта (Інформатика)  
Савченко Максим Олександрович

Керівник: кандидат педагогічних наук, доцент  
кафедри інформатики, програмної інженерії та  
економічної кібернетики  
Кушнір Н.О.

Рецензент:  
кандидат педагогічних наук,  
старший викладач Григор'єва В.Б.

**Херсон – 2021**

## ЗМІСТ

<b>РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ДОБОРУ І ВИКОРИСТАННЯ МОВИ ПРОГРАМУВАННЯ В ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ ПРИ ВИВЧЕННІ РОЗДІЛУ «АЛГОРИТМИ ТА ПРОГРАМИ» .....</b>	<b>6</b>
1.1. Загальні питання вивчення розділу «Алгоритми та програми» в основній школі .....	6
1.2. Проблеми вибору мови програмування в шкільному курсі інформатики.....	8
1.3. Аналіз добору і використання мов програмування при вивченні розділу «Алгоритми та програми» у 8-9 класах .....	13
1.4. Аналіз середовищ програмування з позиції можливостей їх використання під час вивчення розділу «Алгоритми і програми»..	14
<b>РОЗДІЛ 2. МЕТОДИКА ЗАСТОСУВАННЯ МОВ ПРОГРАМУВАННЯ ПРИ ВИВЧЕННІ РОЗДІЛУ «АЛГОРИТМИ ТА ПРОГРАМИ» В БАЗОВОМУ ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ.....</b>	<b>300</b>
2.1. Методичні особливості добору і використання мов програмування при вивченні розділу «Алгоритми та програми» у 8-9 класах .....	300
2.2. Порівняльна характеристика мов програмування для учнів 8-9 класів на приклади розв’язування задач на мовах Java, C++ та Python .....	400
2.3. Організація та проведення педагогічного експерименту .....	466
<b>ВИСНОВКИ .....</b>	<b>500</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>522</b>
<b>Додаток А.....</b>	<b>566</b>
<b>Додаток Б .....</b>	<b>577</b>
<b>Додаток В .....</b>	<b>58</b>

## ВСТУП

Інформатика в закладі загальноосвітньої середньої освіти належить до групи технологічних дисциплін з інформаційно-технологічним підходом до визначення мети, змісту і методики навчання, з яскраво вираженими засобами, спрямованими на здобуття учнями практичних компетенцій, які можуть забезпечити вивчення фундаментальних тем інформатики, що сприяють розвитку основних форм мислення. Одними із фундаментальних питань шкільного курсу інформатики є моделювання, алгоритмізація та програмування.

**Актуальність дослідження** — визначення сучасних підходів і шляхів використання мов програмування, як засобу навчання програмування на уроках інформатики загальноосвітніх навчальних закладів.

Таке удосконалення можливе у декількох напрямках, і тут розглянемо лише деякі з них. Передусім варто знайти нові а не старі підходи, які зводять навчання алгоритмізації та програмування до вивчення синтаксису і семантики мови Паскаль у середовищі Turbo Pascal, Free Pascal, Object Pascal і з підходами, коли застарілі дидактичні засоби й методи навчання переносяться до сучасних середовища об'єктно-орієнтованого програмування і його різновиди — середовища візуального програмування.

Методичні аспекти навчання програмування у середній школі досліджують М.І. Жалдак, Н.В. Морзе [4], Ю.С. Рамський, О.І. Кривонос, В.Д. Руденко, С.О. Семеріков, О.М. Спірін [5], П.Г. Шевчук [26] та інші.

Так, у дослідженнях О.М. Спіріна і П.Г. Шевчука розглядаються програмні технології операційної системи Windows з погляду навчання програмування [5], відмінні риси мови програмування C# [6], програмно-технологічні умови застосування мов програмування у

загальноосвітніх навчальних закладах [7]. У працях Ф.С. Ільясової розкриваються особливості навчання мовою програмування Java [9].

Вибору мови програмування для навчання у школі присвячені праці Я.М. Глинського, Л.В. Палюшок [10], С.С. Жуковського, О.В. Коротун [11] та ін.

**Мета дослідження** полягає у теоретичному обґрунтуванні методичних рекомендацій щодо добору і використання мов програмування у процесі вивчення розділу «Алгоритми та програми» у 8-9 класах.

Згідно з поставленою метою в роботі вирішуються наступні **завдання**:

– на основі аналізу науково-методичної літератури з проблеми дослідження, сучасного стану викладання програмування у школах України, трендів ІТ-ринку визначити, які мови програмування є найбільш перспективними та актуальними для навчання дітей у сучасному світі;

– на основі порівняльного аналізу мов та середовищ програмування з точки зору відповідності програмі шкільного курсу інформатики, віковим особливостям учнів, складності опанування тощо, визначити, яка мова найкраще підходить для вивчення розділу «Алгоритми та програми» у 8-9 класах;

– здійснити впровадження розроблених методичних рекомендацій у навчально-виховний процес закладу загальної середньої освіти та узагальнити результати теоретичного та експериментального дослідження.

**Об’єктом дослідження** є процес вивчення інформатики у закладах загальної середньої освіти.

**Предметом дослідження** методичні особливості добору і використання мов програмування у 8-9 класах.

**Методи дослідження:** *теоретичні:* аналіз науково-методичної літератури, порівняння, узагальнення, систематизація, формулювання висновків; *емпіричні:* бесіда, спостереження, анкетування учнів.

**Практичне значення одержаних результатів** полягає в тому, що запропоноване методичне забезпечення щодо добору і використання мов програмування у 8-9 класах у процесі вивчення розділу «Алгоритми та програми».

**Апробація результатів дослідження** проводилася на базі Чаплінської спеціалізованої школи I-III ступенів Чаплінської селищної ради Херсонської області.

**Структура роботи.** Дипломна робота складається зі вступу, двох розділів, висновків, списку використаних джерел, який включає 38 найменувань. Повний обсяг роботи – 54 сторінки.

## РОЗДІЛ 1

### ТЕОРЕТИЧНІ ОСНОВИ ДОБОРУ І ВИКОРИСТАННЯ МОВИ ПРОГРАМУВАННЯ В ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ ПРИ ВИВЧЕННІ РОЗДІЛУ «АЛГОРИТМИ ТА ПРОГРАМИ»

#### 1.1. Загальні питання вивчення розділу «Алгоритми та програми» в основній школі

Ми живемо у цифровому світі, де комп'ютерні програми лежать в основі всього: від бізнесу та маркетингу до науки та медицини. Стрімке зростання ринку праці у сфері ІТ зумовлює збільшення потреби у кваліфікованих фахівцях, які володіють сучасними мовами програмування.

Однією з вагомих проблем вивчення програмування у шкільному курсі інформатики є вибір середовища програмування. Перший етап навчання конкретної мови програмування полягає саме в ознайомленні з інтерфейсом середовища програмування. Від вибору середовища програмування значною мірою залежить ефективність вивчення сучасних мов програмування, більшість з яких передбачають створення екранних форм і використання візуальних компонентів.

Раціонально побудований інтерфейс середовища програмування значно спрощує процес засвоєння учнями початкових навичок роботи з даним середовищем, здійснення основних операцій з проектом і файлами, що містять текст програми та інші складові проекту.

У той же час застосування середовища програмування, яке має зручний і раціонально сконфонований інтерфейс, зводить до мінімуму час пошуку необхідних компонентів, а також час виконання рутинних операцій (збереження файлів, компіляція програми тощо). Важливу роль відіграє також відносна швидкодія середовища програмування, тобто час створення, збереження та компілювання програми.

Знайомство з програмування доцільно здійснювати вже у середній школі. Програмування дає учням уявлення про те, що можна створити за допомогою технологій. Нас оточують пристрої, керовані комп'ютерами. Розуміння того, як вони працюють, і представлення нових пристроїв і

послуг поліпшується за рахунок розуміння кодування [9].

В основі всіх комп'ютерних програм лежать алгоритми, які більш абстрактно визначають спосіб виконання завдання. Алгоритмічне мислення - лежить в основі інформатики, і тому в школах йому приділяють все більше і більше уваги.

Концепція нової української школи, яка була ухвалена рішенням колегії МОН України від 27.10.2018 р., визначає 10 ключових компетентностей, які повинні бути сформовані у випускників середньої школи. Однією з них є інформаційно-цифрова компетентність, яка також передбачає володіння основами програмування і розвиток алгоритмічного мислення [19]. Саме тому формування початкових здатностей до написання програм і розробки алгоритмів є обов'язковою складовою змісту середньої освіти.

Аналізуючи розподіл навчальних годин на вивчення розділів програми «Інформатика», ми бачимо, що саме на вивчення розділу «Основи алгоритмізації та програмування» виділяють найбільше часу у 8-ому та 9-ому класах середньої школи (Додаток А).

Програмування можна віднести до тем, які для учнів є досить складними, при цьому треба визнати, що рішення задач по темі «алгоритмізація і програмування» допомагає більшою мірою розвитку алгоритмічного мислення і формує у школярів навчальні навички. Розвинуте алгоритмічне мислення необхідно при написанні доповідей, рефератів. Крім того, учням треба пояснити, що схожу послідовність дій нам доводиться виконувати, вирішуючи повсякденні побутові або організаційні проблеми.

Вимоги до рівня загальноосвітньої підготовки учнів основної школи з інформатики визначають у «Державному стандарті базової і повної загальної освіти». Зокрема визначено, що випускники середньої школи повинні знати:

- 1) поняття алгоритму;
- 2) етапи і способи описання алгоритмів;
- 3) основні алгоритмічні конструкції: слідування, розгалуження, множинний вибір, цикл, структури алгоритмів;
- 4) призначення мов програмування;
- 5) правила подання даних на одній з мов програмування

високого рівня.

Повинні вміти:

- 1) формулювати і виконувати основні етапи алгоритмічного розв'язування задачі;
- 2) пояснювати операції з даними різних типів, зокрема логічними та рядковими, і використовувати їх в алгоритмах;
- 3) створювати алгоритми з вкладеними структурами та ітеративними обчисленнями, аргументувати їх вибір;
- 4) прогнозувати вплив зміни алгоритму, наборів вхідних даних на результат роботи алгоритму;
- 5) порівнювати альтернативні алгоритми розв'язання однієї задачі за різними ознаками тощо [7].

## **1.2. Проблеми вибору мови програмування в шкільному курсі інформатики**

У кожній конкретній системі програмування можна виділити наступні компоненти:

- середовище;
- режими роботи;
- система команд;
- дані.

Середовище системи програмування (СП) – це обстановка на екрані, у якій працює користувач системи програмування. Учитель показує й описує конкретну оболонку, за допомогою якої будуть працювати учні [12].

Характерні режими роботи систем програмування:

1. режим редагування тексту програми;
2. режим компіляції програми;
3. режим виконання програми;
4. робота з файлами;
5. режим допомоги;
6. налагодження програми [12].



*Режим редагування.* Частіше встановлюється автоматично при ініціалізації роботи системи. При цьому працює вбудований у систему текстовий редактор. Текст програми, за бажанням, можна підготувати в будь-якому текстовому редакторі, але це незручно. Освоїти роботу з текстовим редактором СП учням допоможуть навички, отримані при вивченні теми «Обробка текстової інформації» [12].

*Режим компіляції.* Результатом компіляції є програма, яка записана мовою машинних команд для майбутнього виконання. Одержання програми, що виконується, проходить декілька етапів: компіляція, робота редактора зв'язків, завантаження і т.д., але в базовому курсі ці питання не розглядаються. Їх можна розглянути лише при поглибленому вивченні курсу.

*Режим виконання.* У системах, що компілюють, в цьому режимі виконується програма в машинних командах, яка отримана після трансляції. Інтерпретатор безпосередньо сам виконує програму. У цих двох випадках виконання починається по команді RUN [12].

*Режим роботи з файлами.* У файлах на зовнішніх носіях система зберігає тексти програм вихідною мовою; програми, отримані в результаті трансляції; вихідні і кінцеві результати. У файловому режимі виконуються традиційні операції: зберегти інформацію у файлі, прочитати дані з файлу в оперативну пам'ять, присвоїти ім'я файлу, друкувати текст програми й ін. [12].

*Режим допомоги.* Дозволяє програмісту одержати підказку на екрані, що допомагає як у роботі із системою, так і з мовою програмування [12].

*Режим налагодження.* У цьому режимі можна робити трасування, покрокове виконання програми, слідкувати за зміною визначених величин, зупинити виконання програми у певному місці по визначеній умові. Режим налагодження дає можливість знайти алгоритмічні помилки в програмах [12].

У 8 класі учні починають вивчати алгоритмічні структури з використанням мови програмування високого рівня і відповідного середовища програмування, тому вибір першої мови програмування дуже важливий з точки зору отримання правильних звичок програмування учнями, а також розуміння відповідних алгоритмів. Від вибору мови та системи програмування багато в чому залежить методика вивчення курсу, зміст і послідовність пред'явлення навчального матеріалу, система навчальних завдань і, головне, вся подальша робота по оволодінню програмуванням для вирішення реальних практичних завдань на комп'ютері.

На сьогоднішній день існує великий вибір мов програмування — починаючи від найпростіших (тих, на яких навчають дітей молодших класів), і закінчуючи такими потужними інструментами розробки, як C++, Java, C# і т. д. Зараз вчителі інформатики мають можливість досить вільно обирати мови і середовища програмування для вивчення у 8-9 класах. Наприклад, у нових підручниках для 8 класу пропонують розглянути такі мови програмування і середовища: мова програмування Free Pascal та середовище розробки Lazarus; мова програмування Python та середовище розробки PyCharm; мова програмування Delphi; мова програмування Visual Basic з середовищем розробки Visual Basic.NET; мова програмування Turbo Pascal, середовище розробки Turbo Pascal.

Варто зазначити, що нині з'являються нові погляди на зміст курсу алгоритмізації і програмування, зокрема, одним з питань, що активно обговорюють, є вибір першої мови програмування. Бо саме вона впливатиме на успіх учнів у використанні та створенні алгоритмів. Розглядати необхідно саме мови програмування високого рівня зображені на рисунку 1.1.

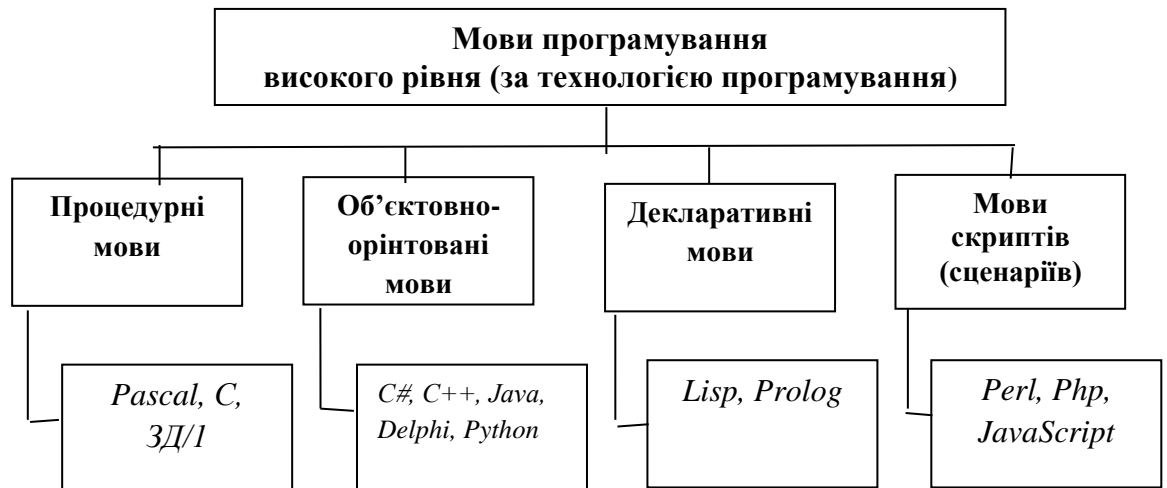


Рис. 1.1 Класифікація мов програмування високого рівня

Можемо виділити наступні вимоги до першої мови програмування:

- 1) прямота або лінійність – легкість кодувати без потреби в термінах і структурах, які користувач ще не знає;
- 2) прозорість – легка читабельність коду навіть для тих, хто вміє програмувати, але не знає мови;
- 3) стійкість – виявляти помилки при виконанні програми та легко виправляти;
- 4) абстракція – вибір рівня абстракції за навичками учнів;
- 5) середовище розвитку – якісне середовище розвитку, що відповідає складності освітньої програми;
- 6) доступність – ліцензії, середовище розробки має бути безкоштовно доступним для всіх учнів на всіх часто використовуваних пристроях та операційних системах.
- 7) наявність методична підтримки (методичного забезпечення) навчального процесу.

Мова, яка може використовуватися в освітніх цілях, повинна дозволяти учням і вчителям сконцентруватися на алгоритмах і придбаних компетенціях, не обтяжуючи їх синтаксисом мови і складним середовищем розробки. Розуміння того, як висловлювати концепції, щоб комп'ютер міг виконувати завдання точно і ефективно, набагато важливіше, ніж деталі мови програмування.

Незважаючи на те, що у 8-9 класах основної школи здебільшого вивчають Pascal і Lazarus, практики вважають більш доцільним впроваджувати мови програмування, які застосовуються на підприємствах IT-індустрії, наприклад: C++ і Java або Python та ін. Наприклад, науковці Маловічко О. та Конюхов С. вважають найкращим варіантом у якості першої мови програмування використовувати C++, оскільки синтаксис багатьох мов програмування ґрунтується на синтаксисі цієї мови [11]. Науковці Круглик В.С. та Львов М.С. пропонують вивчати мову Python першою, оскільки вона є потужною і простою, і навчання можна розпочати без будь-якої підготовки [35].

Mar 2021	Mar 2020	Change	Programming Language	Ratings
1	2	▲	C	15.33%
2	1	▼	Java	10.45%
3	3		Python	10.31%
4	4		C++	6.52%
5	5		C#	4.97%

Рис. 1.2. Рейтинг мов програмування у 2021 році

Якщо поглянути на щорічний рейтинг популярності мов програмування у світі, який складає спілка програмістів TIOBE, ми можемо побачити, що такі мови програмування, як Python, Java та C++, знаходяться у топі цього рейтингу. Це робить їх ще більш привабливими для вивчення.

Як ми бачимо, вибір мови та системи програмування має принципове значення. Від цього вибору безпосередньо залежить доступність сприйняття, вивчення і оволодіння учнями прийомами і методами створення програм та алгоритмів.

Але важливо пам'ятати, що у школі неможливо і не потрібно вивчати всі можливості мови програмування. Метою вивчення розділу «Алгоритмізація та програмування» в школі є опанування учнями

способів, алгоритмів і методів програмування при вирішенні завдань, тобто прищеплення їм навичок алгоритмічного мислення, а не вивчення мови програмування або підготовка висококваліфікованих програмістів. Отже, кінцева мета вивчення будь-якої мови програмування не вивчення мови програмування і не підготовка висококваліфікованих програмістів це заохочення і стимулювання самостійної роботи учнів шляхом написання власних проєктів.

### **1.3. Аналіз добору і використання мов програмування при вивченні розділу «Алгоритми та програми» у 8-9 класах**

На сьогоднішній день доступна величезна кількість мов програмування, тому часто виникає питання, яка є найкращою для навчання основ програмування учнів загальноосвітніх шкіл. Саме тому варто розглянути кожен з найбільш популярних мов програмування для викладання у школі та проаналізувати їх переваги та недоліки.

Мова програмування Java була запропонована Джеймсом Гослінгом і його командою з Sun Microsystems у 1991 році і випущена у 1995 році. Найвидатніша особливість Java полягає в тому, що вона не залежить від платформи, тобто має властивості WORA (писати один раз, запускати будь-де).

Java - це скомпільована мова зі статичною типізацією, тобто її змінні повинні бути оголошені до присвоєння значень. Програми, написані на Java, працюють швидше, ніж на Python, але повільніше в порівнянні з C ++.

Синтаксис мови подібний до синтаксису мов C та C++, але все ж він є більш читабельним. На відміну від C та C++, програми, написані мовою Java, компілюються у байткод, а вже байт код виконується віртуальною машиною Java для конкретної платформи [1].

Для успішної роботи необхідно підключати додаткові бібліотеки або імпортувати в проєкт вже існуючі для їх подальшого використання. Багато функцій, які доступні в інших мовах без особливого зусилля,

необхідно ініціалізувати перед використанням, що не полегшує навчання програмування школярів в базовому курсі інформатики.

На жаль, вивчення Java як першої мови виявляється важчим, ніж здається спочатку. Java синтаксично схожа на C ++, і багато з основних труднощів програмування C ++ (наприклад, синтаксичні проблеми) також виникають в Java. Крім того, деякі важливі теми важче вивчати за допомогою Java, ніж з деякими іншими мовами програмування [12].

Відзначимо *переваги* мови Java:

1) Java - строго типізована мова і має суворі вимоги, які змушують учнів правильно міркувати.

2) Java є простою у використанні, написанні, компіляції, налагодженні та вивченні в порівнянні з іншими мовами, такими як C, C ++ або C#.

3) Дана мова є незалежною від платформи, розподіленою, підтримує багато поточність.

4) Використання ООП в Java дозволяє створювати стандартні програми і повторно використовуваний код [18].

*Недоліки:*

1) Багатослівність і складність коду. Мова з довгими, важкими реченнями допомагає при його вивченні. Але зайва інформація ускладнює читання. Тому Java вважається занадто громіздкою мовою.

2) Може виникати багато проблем, пов'язаних із синтаксисом мови [30].

Отже, можемо зробити висновок, що вивчення Java як першої мови має багато труднощів. Тому вона краще підходить для вивчення програмування у спеціалізованих школах та ліцеях, де діти більш зацікавлені у його вивченні.

C ++ – це мова програмування загального призначення, розроблена Бьярном Страуструпом в Bell Labs як розширення мови програмування C. C ++ – надзвичайно потужна мова, що містить засоби створення ефективних програм практично будь-якого призначення, від низько рівневих утиліт і драйверів до складних програмних комплексів самого різного призначення.

Прийнято вважати, що С ++ – потужна мова, але надзвичайно складна. Її значно важче вивчити, ніж більшість інших мов. У новачків виникає багато проблем з покажчиками, перевантаженням операторів, множинним спадкуванням та шаблонами. Крім того, С ++ повинен слідувати всім синтаксичним обмеженням С та всім додатковим обмеженням, що були введені мовою С ++. Ці синтаксичні обмеження не дуже складні, коли вони засвоєні, проте вони можуть лякати учнів, які тільки починають вчити програмування [12].

Незважаючи на такі переваги, як кросплатформеність і ефективність, С ++ володіє високим порогом входження. Всі ці складності роблять С ++ досить суперечливою мовою. З одного боку, це низькорівнева багатоплатформенна мова програмування, яка ще довгий час буде актуальною через свою універсальність і ефективність. З іншого боку, учням буде складніше опанувати різні алгоритми і принципи програмування на С ++ саме через його статичну типізацію, що обмежує роботу зі змінними [28].

Проаналізувавши мову програмування С++, можемо виділити наступні її *переваги*:

- зручність вивчення об'єктно-орієнтованого програмування;
- наявність значної кількості вільно доступних середовищ розробки, переважна більшість з яких підтримує візуальне програмування;

*Недоліки:*

- складна робота з покажчиками;
- складний синтаксис.

Узагальнюючи вище наведене, можна стверджувати, що мова С++ розрахована більше на зацікавлених користувачів-програмістів, але не дуже підходять для навчання основам програмування в базовому курсі інформатики у школі, так як в класах даної категорії досить велика кількість учнів, які не цікавляться програмуванням. Тому підвищена складність даної мови, викликана її спеціалізованою, може послабити інтерес до програмування.

Мова програмування Python була створена Гідо Ван Россумом в 1989 році. Це мова програмування інтерпретатора, розроблена як проект з відкритим кодом. Python підтримує об'єктно-орієнтоване програмування, процедурне, а також функціональне програмування. Це мова на різних платформах, що означає, що програми, написані на Python, працюють під багатьма операційними системами, включаючи системи Microsoft Windows, Linux та Unix, такі як Mac OS X, з майже повною підтримкою стандартних та сторонніх бібліотек, просто копіюючи вихідний код програми [28].

Python має досить легкий та зрозумілий синтаксис. Наприклад, звичайна програма, яка печатає фразу «Hello, world!», займає лише один рядок коду:

```
#This program prints Hello, world!  
print('Hello, world!')
```

Прості програми записуються у кілька рядків, відсутні інструкції, що не мають безпосереднього відношення до алгоритму (для прикладу `int main()` у мові C). Як правило, програми мовою Python є коротшими, ніж на C, C++ або Pascal.

Варто зазначити, що Python - мова зі строгою динамічною типізацією. Це означає, що у ньому тип змінної трактується в залежності від присвоєного значення та не допускається неявне перетворення типу.

Однією з переваг мови Python є також кількість існуючих бібліотек, доступних безкоштовно, але їх використання не є обов'язковим. Новачкам не потрібно знати бібліотеки, їх функції та особливості для вивчення основ програмування. Тому, на нашу думку, краще інтегрувати бібліотеки в навчальну програму поступово.

Ще однією перевагою для учнів є той факт, що Python не призначений виключно для навчальних цілей, пізніше вони можуть використовувати його під час своєї професійної кар'єри, оскільки він використовується в багатьох областях інформаційних технологій.

До недоліків Python традиційно відносять порівняно невисоку швидкість виконання програм, написаних цією мовою. Дійсно, транслятор Python - інтерпретатор, тому програми працюють



повільніше. Але зі зростанням обчислювальної потужності ПК це стає несуттєвим [28].

Отже, *перевагами* мови Python є:

1. Легка читабельність коду;
2. Чіткий синтаксис мови;
3. Python є інтуїтивно зрозумілою мовою програмування;
4. Підтримує мову сценаріїв, ООП, функціональне програмування.
5. Існує багато навчальної літератури.

*Недоліки:*

1. Єдиним недоліком є те, що типи змінних не оголошуються, і тип значення може змінюватися під час виконання. Це може ускладнювати розуміння учнями різних типів даних та їх використання.

Проаналізувавши мову програмування Python, ми можемо побачити, що вона має свої переваги і недоліки. Але позитивних сторін у цієї мови більше. Python - проста і мінімалістична мова, що дає можливість зосередитися на вирішенні завдання, а не на самій мові.

Зробимо порівняльну характеристику мов програмування, які найкраще можуть підійти для навчання учнів основам програмування та алгоритмізації:

*Таблиця 1.1*

### Оцінка мов програмування

<b>Критерії оцінки</b>	<b>Java</b>	<b>C++</b>	<b>Python</b>
Легкість опанування	-	-	+
Легкий перехід на інші мови	+	+	-
Читабельний синтаксис	-	-	+
Наявність зручних середовищ розробки	+	+	+
Відсутність надмірності	+	-	+
Підтримка структурної парадигми програмування	+	+	+
Підтримка об'єктно-орієнтованої парадигми програмування	+	+	+

Отже, в результаті порівняльного аналізу декількох мов програмування, ми з'ясували, що найбільш методично слушною для навчання учнів 8-9 класів основам програмування та алгоритмізації є мова програмування Python, яка має досить простий і лаконічний синтаксис, завдяки якому учні зможуть зосередитись на вивченні логіки роботи програми замість того, щоб боротися з правилами розставлення у ній розділових знаків. Незважаючи на популярність і затребуваність мов C++ та Java на ринку праці, складність синтаксису, необхідність роботи з об'єктами вже із самого початку роботи можуть ускладнити сприйняття і без того непростого матеріалу учнями. Нестроге приведення типів, яке притаманне для цих мов збільшує вірогідність появи помилок, які початківцю досить складно знайти. З іншого боку, якщо пройти ці труднощі початкового навчання, можна буде створювати досить професійні програми. Але не дуже високий відсоток учнів дійсно подолає цей рубіж. Тому мови C++ та Java для початкового вивчення не рекомендується. Вони можуть бути запропоновані учням для більш поглибленого вивчення програмування на факультативних заняттях або використовуватися на олімпіадах.

#### **1.4. Аналіз середовищ програмування з позиції можливостей їх використання під час вивчення розділу «Алгоритми і програми»**

Якщо ви хочете навчити програмувати своїх учнів на будь-якій мові, то вам не обійтися без двох речей: методики та середовища розробки для цієї мови.

Однією з найбільш важливих проблем вивчення програмування у шкільному курсі інформатики є саме вибір середовища програмування, бо від нього значно залежить успішність виконання учнями основних операцій, пов'язаних зі створенням програм. Важливу роль відіграє також швидкість роботи середовища програмування, тобто час створення, збереження та компілювання програми. Все це значно

впливає на ефективність навчання та швидкість опанування учнями основ програмування та алгоритмізації.

На думку Базуріна В.М., саме правильно та зрозуміло побудований інтерфейс середовища програмування значно спрощує процес засвоєння учнями початкових навичок роботи з ним, здійснення основних операцій з проектом і файлами, що містять текст програми та інші складові проекту [2]. Тому важливо обирати саме ті середовища, які б були найбільш зручними та легкими в опануванні роботи з ними.

Основними вимогами до середовища програмування для освітніх цілей є:

1) Безкоштовне розповсюдження - система програмування повинна бути безкоштовною.

2) Наявність операційних систем і додаткового програмного забезпечення, необхідного для функціонування середовища розробки.

3) Кросплатформеність - бажано не прив'язувати студента до певної платформи, тому що користувач повинен сам вибирати платформу для роботи.

4) Наявність навчально-методичного забезпечення (підручників і збірників задач).

5) Система з відкритим кодом – система програмування повинна бути відкритою, із відкритими вихідними кодами.

Найбільш популярними середовищами для програмування мовою Java є IntelliJ IDEA, Eclipse, JBuilder, NetBeans.

Розглянемо більш детально деякі з них.

IntelliJ IDEA - середовище розробки, яке підтримує такі мови програмування, як Java, Python, Scala та інші [33]. Її розробник – компанія JetBrains – випускає середовище у двох версіях – Community Edition (безкоштовна, має обмежений функціонал) і Ultimate Edition (платна, що має повний функціонал). Дане середовище має багатий функціонал, дозволяє розробляти програми для різних платформ, таких як Windows, Mac, Android, Linux. Версія Ultimate Edition підтримує мови Java, Python, Ruby, PHP та інші. Версія Community Edition допускає лише роботу з кодом програми і лише одну мову програмування – Java.

З огляду на вказані обмеження у навчальному процесі краще використовувати версію IntelliJ IDEA Ultimate Edition, але її ціна є досить високою, тому це може звужувати можливості застосування даного середовища програмування з навчальною метою.

З огляду на все вищесказане, можна виділити наступні переваги та недоліки використання даного середовища для програмування:

*Переваги:*

1. невисокий поріг входження;
2. простий інтерфейс програми;
3. при допущенні помилки, або написання коду, що може потенційно містити помилку, середовище розробки не тільки вкаже на помилку, та її місце, а ще й запропонує варіанти її вирішення. Це значно полегшує процес опанування нової мови програмування;
4. доволі широкий функціонал програмного середовища, що може спонукати учнів до подальшого самостійного, більш глибокого знайомства з ним.

*Недоліки:*

1. Безкоштовна версія програми має дещо обмежений функціонал, а ціна повної версії є доволі високою.

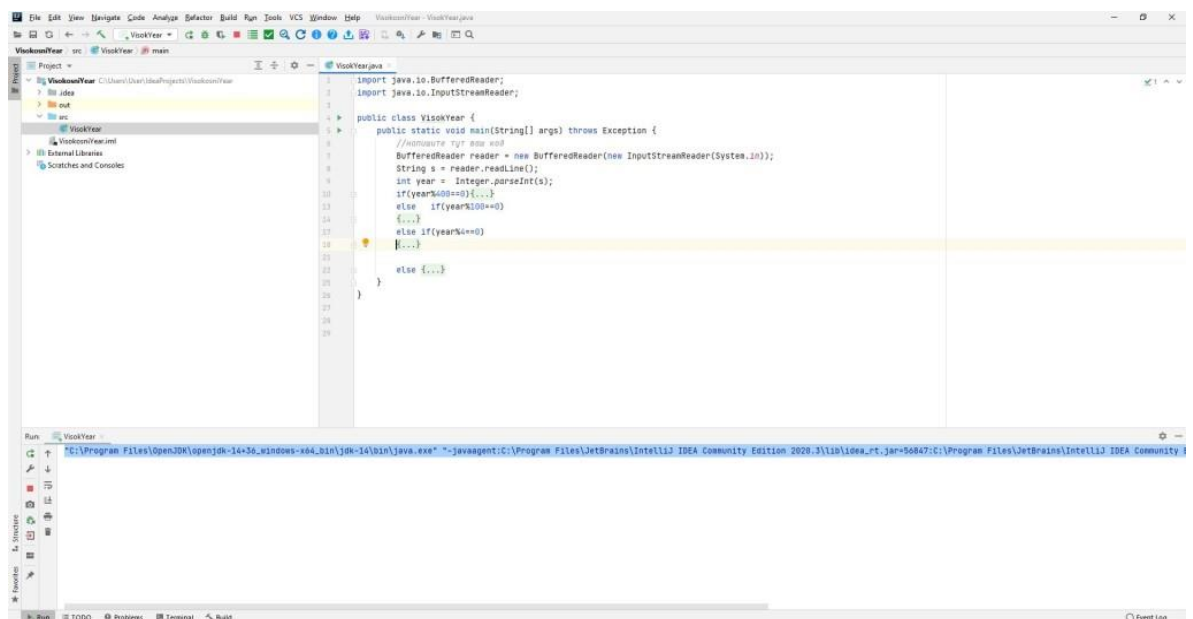


Рис. 1.3. Середовище IntelliJ IDEA

Середовище програмування Eclipse створене для розроблення програм мовою Java, хоча також має і додаткові модулі для створення програм іншими мовами [5]. Поширюється фірмою IBM за безкоштовною ліцензією в якості засобу для розробки програмного забезпечення на мові Java та інших мовах (C, C ++, Ruby, Fortran тощо).

Eclipse є платформи-незалежним середовищем, яке здатне працювати на різних платформах і в різних операційних системах. Для того, щоб дане середовище працювало, потрібно встановлювати додаткові модулі та розширення.

Інтерфейс програми Eclipse є англійським, підказки також здійснюються англійською мовою.

*Переваги* даного середовища програмування:

1) існує багато навчальних посібників щодо розроблення програм мовою Java у середовищі Eclipse;

2) середовище має багатий функціонал;

3) має простий інтерфейс.

Можна відокремити наступні *недоліки*:

1) неможливість створення форм у візуальному режимі. Це може значно ускладнювати процес створення програм учнями;

2) досить високий поріг входження для оволодіння програмуванням у середовищі Eclipse, тому для багатьох учнів буде складно його подолати;

3) для роботи потрібно встановлювати додаткові компоненти, не зручна у використанні.

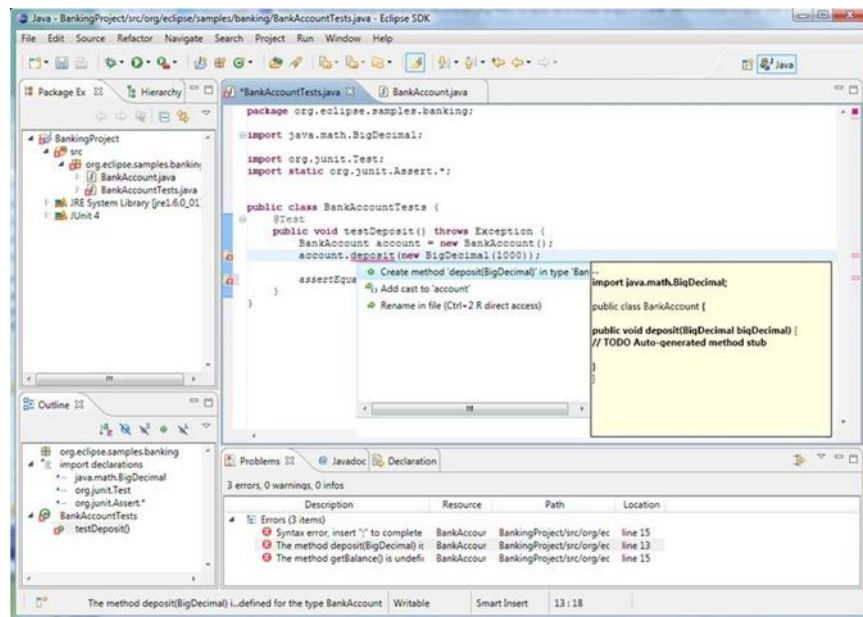


Рис. 1.4. Середовище Eclipse IDE

Порівняємо дані програмні середовища за такими критеріями (таблиця 1.2):

- 1) комерційна складова: платна (-) чи безкоштовна (+);
- 2) простота інтерфейсу;
- 3) зручність у використанні;
- 4) наявність навчально-методичного забезпечення.

Таблиця 1.2

### Порівняння середовищ програмування мовою Java

Критерій	IntelliJ IDEA	Eclipse
Безкоштовне середовище	-	+
Простота інтерфейсу	+	+
Легкість опанування	+	+
Наявність навчально-методичного забезпечення	-	+

Отже, найбільш придатним для використання учнями в процесі вивчення програмування мовою Java є Eclipse. Звичайно, це не виключає можливості використання інших середовищ програмування, але орієнтуватися слід на використання Eclipse.

Найбільш поширені середовища програмування мовами C/C++ є Microsoft Visual Studio, Borland C++, DEV C++ та ін.

Розглянемо Microsoft Visual Studio. На сьогодні розробник середовища - Microsoft - пропонує три версії Visual Studio 2019: Community, Professional та Enterprise. Версія Visual Studio Community представляє найбільший інтерес для навчальних закладів, тому що розповсюджується безкоштовно, потребує менше системних ресурсів і має достатній функціонал для навчання програмування.

*Переваги використання:*

1. Інтуїтивно-зрозумілий інтерфейс середовища;
2. Наявність документації англійською і російською мовами, є велика кількість уроків у мережі Інтернет.

*Недоліки:*

1. Високі системні вимоги.

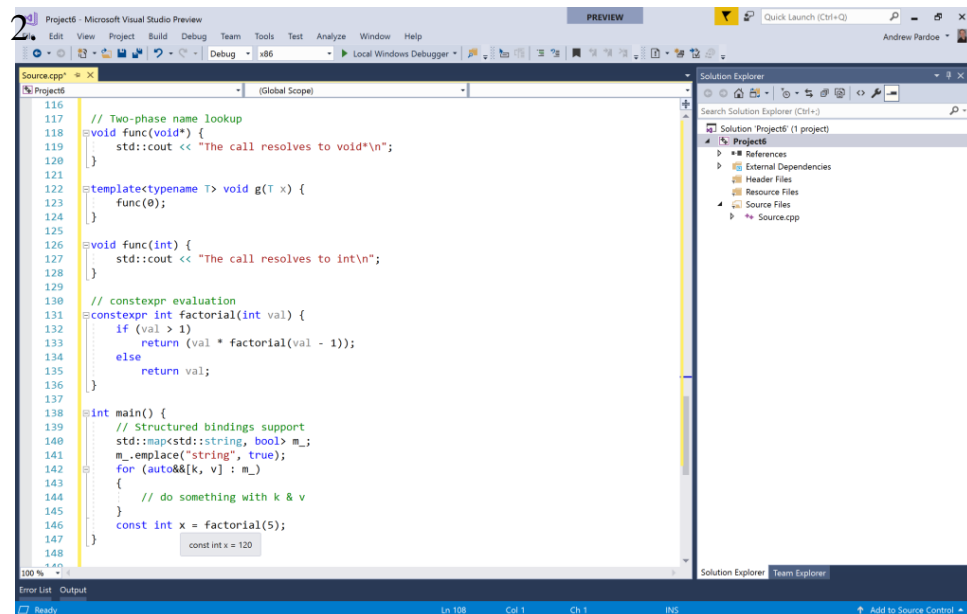


Рис. 1.5. Середовище розробки Microsoft Visual Studio

Середовище програмування Dev-C++ належить до класу програмного забезпечення, що є вільно поширюваним. Має досить низькі системні вимоги, не вимагає дуже потужного процесора або значного обсягу оперативної пам'яті.

Мова інтерфейсу – українська, проте підказки здійснюються англійською мовою. Це може дещо ускладнювати процес налагодження програми учнями, які не дуже добре володіють англійською мовою.

*Плюси середовища розробки:*

- 1) є безкоштовним;
- 2) має низькі системні вимоги;
- 3) має легкий зрозумілий інтерфейс;
- 4) ідеально підходить для вивчення основ програмування на C++.

*Мінуси:*

- 1) тільки для розробки на C++.

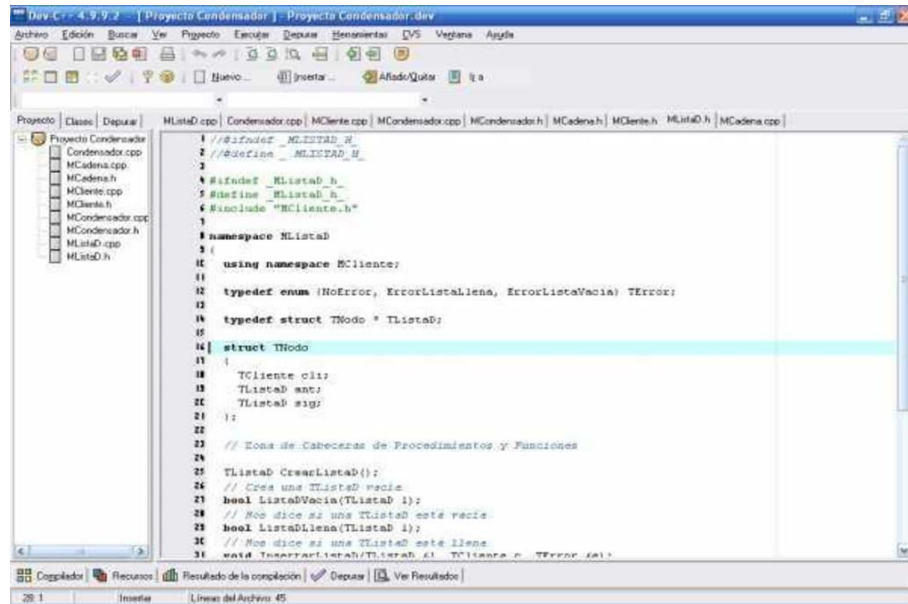


Рис.1.6. Середовище програмування Dev-C++

Таблиця 1.3

### Порівняння середовищ програмування мовою C++

Критерій	Microsoft Visual Studio	DEV C++
Безкоштовне середовище	+	+
Простота інтерфейсу	+	+
Легкість опанування	-	+
Наявність навчально-методичного забезпечення	-	+

Отже можемо зробити висновок, що у випадку, коли вивчення програмування обмежено лише мовами C/C++, то більш доцільним є використання DEV C++. Якщо C або C++ є тільки основою для подальшого вивчення програмування, а далі планується перехід до інших мов, то варто використати Microsoft Visual Studio.



До найбільш популярних середовищ програмування мовою Python належать: PyCharm, Python IDLE, PyDEV та багато інших.

Почати варто з IDLE (інтегроване середовище розробки і навчання), це вже вбудоване середовище розробки в саму мову, тобто встановили мову, а з ним «в коробці» йде середовище розробки. Його можна завантажити з офіційного сайту Python. Це середовище програмування працює в режимі командного рядка. Має простий інтерфейс, мінімальні функції і можливість працювати з ним на Windows, Linux, Mac.

Мова інтерфейсу – англійська. Це може створювати певні труднощі для учня-початківця.

Перевагою даного середовища є досить низькі системні вимоги, це буде дуже доречним під час використання його у школі.

IDLE – добрий редактор для початку програмування і розуміння основ мови. Його можливості великі: автозавершення коду, підсвічування синтаксису, підбір відступу і базовий вбудований відгадчик.

*Плюси:*

- 1) Легкий в опануванні.
- 2) Підходить для початківців.
- 3) Низькі системні вимоги.

*Мінуси:*

- 1) Англійська мова інтерфейсу.
- 2) Бракує просунутих функцій.

Але недоліки даної IDE є незначними та не мають сильного впливу на процес навчання.



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello,world!')
Hello,world!
>>> |
```

Рис. 1.7. Середовище програмування Python IDLE

Наступне середовище програмування – PyCharm [31]. PyCharm розроблений JetBrains і має безліч зручних функцій. Існує як безкоштовний open-source (Community), так і платний (Professional) варіанти IDE. PyCharm доступний на Windows, Mac OS X і Linux.

Дане середовище програмування має віконний інтерфейс, мова інтерфейсу – англійська.

Має можливості інтегрованого модульного тестування, перевірки коду, інтегрованого контролю версій, інструменти рефакторинга коду, набір інструментів для навігації проекту, виділення і автоматичного завершення.

Але всі ці переваги може оцінити вже досвідчений програміст, а в учня-початківця занадто складний та насичений інтерфейс програми може викликати труднощі під час роботи.

*Плюси IDE:*

- 1) має важливі вбудовані функції;
- 2) розроблено професіоналами спеціально для Python;
- 3) має багатий функціонал.

*Мінуси:*

- 1) Високі системні вимоги (якщо ПК або ноутбук має мало оперативної пам'яті, то IDE трохи пригальмовує і працює досить повільно).
- 2) Складний інтерфейс.

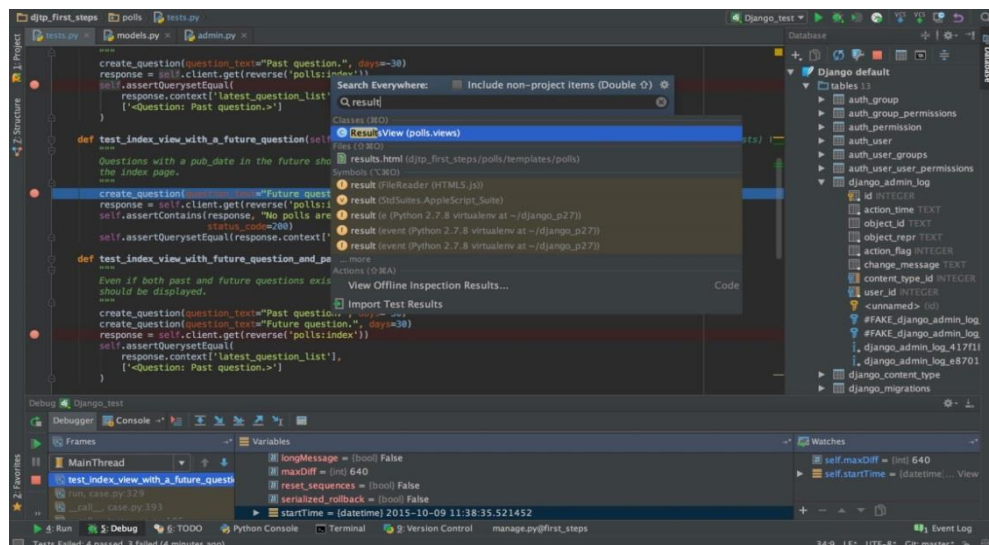


Рис. 1.8. Середовище розробки PyCharm

Ще один гарний редактор для програмування на мові Python для школярів – Mu. Інструмент позиціонується як середовище розробки для початківців і доступний для Windows, MacOS, Linux і Raspbian, офіційної ОС платформи Raspberry Pi. Дозволяє легко створювати, запускати і змінювати власні додатки Python в міру вивчення програмування на Python.

Перевагою є те, що розробники підготували посібник для роботи з новою IDE для початківців для того, щоб забезпечити легкий старт.

Редактор Mu націлений на початківців програмістів зі спробою врахувати потреби вчителів і учнів, ґрунтуючись на їх численних відгуках.

Середовище розробки надає наступні можливості:

- 1) розмітка синтаксису;
- 2) автоматичні відступи;
- 3) вбудована допомога;
- 4) перевірка коду;
- 5) відстеження помилок.

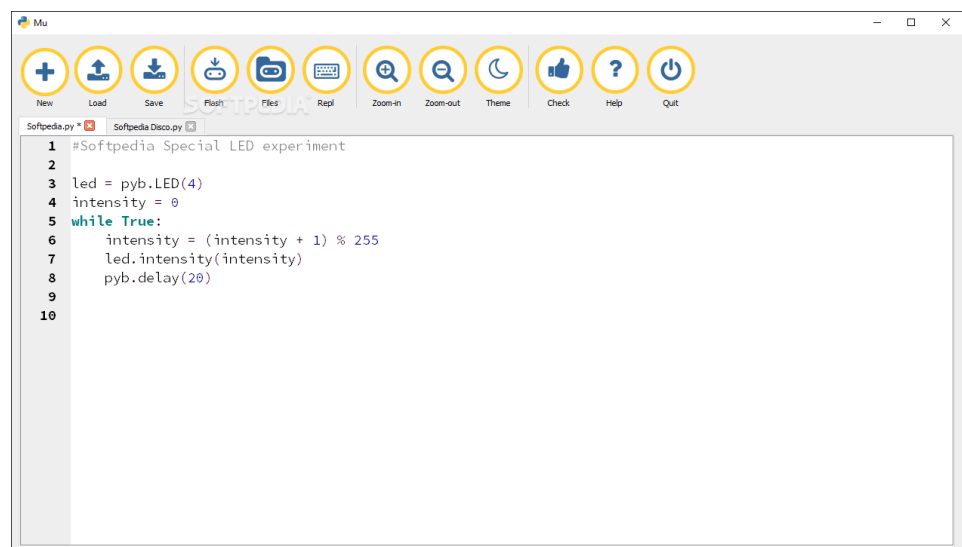


Рис. 1.9. Редактор для програмування Mu

Таблиця 1.4

### Порівняння середовищ програмування мовою Python

Критерій	Python IDLE	PyCharm	Му
Безкоштовне середовище	+	+	+
Простота інтерфейсу	+	-	+
Легкість опанування	+	-	+
Наявність навчально-методичного забезпечення	+	+	+ (лише англійською мовою)

Отже, найбільш придатним для використання учнями в процесі вивчення програмування мовою Python є Python IDLE.

Окрім середовищ програмування, які встановлюються на комп'ютер, існують online-середовища для програмування на мові Python.

Repl.it - популярний онлайн-компілятор для 40 мов програмування (в тому числі для Python) [38]. Це дуже проста, але досить функціональна програма онлайн розробки. У даному онлайн компіляторі доступні сторонні python-пакети.

По суті сервіс являє собою повноцінний IDE в браузері з Python інтерпретатором в комплекті. Тут є підтримка контролю версій, вбудований пошук і установка сторонніх пакетів, дебагер, підсвічування синтаксису і багато іншого.

Ще одне потужне, але просте онлайн середовище для програмування – Tutorialspoint Codingground [14]. Платформа підтримує безліч популярних мов програмування і досить проста в обігу. Тут також є можливість писати на таких мовах, як C, C ++, Java, Perl і т д. Крім того, на сайті є безліч навчальних матеріалів по програмуванню англійською мовою.

Плюси онлайн середовищ:

- 1) Ніяких додаткових установок і налаштувань.
- 2) Простий код запускається без особливого клопоту.

3) Не вимагають встановлення, для роботи з ними достатньо браузера та підключення до мережі Інтернет

Мінуси:

1) Не такі потужні, як IDE.

Підсумовуючи все вищесказане, ми можемо зробити висновок, що середовища розробки є важливим засобом навчання програмування учнів загальноосвітніх шкіл. Від вдалого вибору середовища залежить успіх виконання поставлених задач та подальший інтерес учнів до більш поглибленого вивчення даної технології.

Слід зауважити, що у тому випадку, коли планується вивчати кілька мов програмування, краще використовувати таке середовище розробки, яке надає можливості створення програм більшістю даних мов програмування.

## РОЗДІЛ 2

### МЕТОДИКА ЗАСТОСУВАННЯ МОВ ПРОГРАМУВАННЯ ПРИ ВИВЧЕННІ РОЗДІЛУ «АЛГОРИТМИ ТА ПРОГРАМИ» В БАЗОВОМУ ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ

#### 2.1. Методичні особливості добору і використання мов програмування при вивченні розділу «Алгоритми та програми» у 8-9 класах

Одним із завдань навчання інформатики в основній школі Державний стандарт визначає розвиток алгоритмічного мислення як засобу планування та організації діяльності [12, с. 27].

Тому основи алгоритмізації та програмування становлять одну із змістових ліній шкільного курсу інформатики, яка є наскрізною для всього курсу. У процесі її вивчення учні засвоюють види алгоритмічних конструкцій, прийоми розробки алгоритмів, синтаксис конкретної мови програмування, інтерфейс програмного середовища, а також прийоми створення програми у відповідному програмному середовищі.

Широкий спектр сучасних мов програмування надає вчителю можливість самостійно вибирати мову програмування для вивчення у школі.

Вибір мови програмування є складною психолого-педагогічною проблемою. Дослідники рекомендують використання різних мов програмування залежно від мети вивчення програмування. Одні аргументують вибір мови програмування її простотою, інші – наявністю навчально-методичного забезпечення, інші – актуальністю мови програмування і її поширенням. Дослідженню даного аспекту присвячена значна кількість праць науковців.

У шкільному курсі інформатики доцільно розглядати процедурну та об'єктно-орієнтовну парадигми програмування. У рамках даних парадигм найбільш поширеними мовами є C/C++, Basic, Visual Basic, Pascal, Object Pascal, C#, Java, Python та інші.

Перелічені мови програмування можна умовно поділити на три

групи відповідно до особливостей синтаксису: 1) мови на основі Basic; 2) мови на основі Pascal; 3) мови на основі C [7]. Мови, синтаксис яких базується на синтаксисі C, є найбільш поширеними, тому, як зазначає П.Г. Шевчук, вивчення мов цієї групи дозволяє уникнути значних проблем під час переучування в майбутньому [7, с. 31].

На думку П.Г. Шевчука, на вибір мови програмування впливає низка програмно-технологічних умов, до яких належать:

- технічні характеристики обчислювальної техніки, яка має відповідати системним вимогам програмного середовища;
- наявність операційних систем і додаткового програмного забезпечення, необхідних для повноцінного функціонування середовищ програмування (IDE (англ.) – Integrated Development Environment);
- наявність і доступність середовищ програмування, які доцільно використовувати у навчанні;
- рівень компетентності вчителів інформатики, який необхідний для навчання програмування на відповідній мові [8].

Отже, наявність і доступність програмного середовища відіграють велике значення у виборі мови програмування для вивчення у школі. Причому середовище програмування спочатку є об'єктом вивчення, а в подальшому – засобом навчання.

На думку П.Г. Шевчука, важливими характеристиками середовища програмування, які впливають на його вибір з навчальною метою, є поширеність, доступність, особливості інтерфейсу, спосіб реалізації, системні вимоги, методична підтримка [7, с. 31]. Учений робить висновок про те, що вибір середовища програмування є важливим елементом системи навчання програмування.

Я вважаю що, алгоритмізація, як розділ інформатики, який вивчає процеси створення алгоритмів, традиційно відноситься до теоретичної інформатики внаслідок свого фундаментального характеру. Тому в наступний час є вивчення саме структурованих мов програмування, з наступним вивченням теоретичних основ, які реалізують загальні поняття і принципи об'єктно-орієнтованого програмування, а також формування навичок програмування на об'єктно-орієнтованих мовах [12].

Весь процес навчання програмування в школі розбивається на кілька етапів. Перед початком навчання вчитель стикається з проблемою вибору мови програмування для вивчення. Одна група вчителів навчають темі «Алгоритми та програми» на основі формальних алгоритмів, побудувавши навчання учнів на мові блок-схем. Інша група вчителів інформатики навчають учнів тієї мови програмування, за допомогою якої вміють вирішувати завдання, знають основи відповідної мови і використовують методичний матеріал обраної ними мови програмування. Тим самим втрачено єдиний освітній інформаційний простір не тільки країни, а й окремих регіонів [3].

Таким чином, на початку навчання темі «Алгоритми та програми» вчителю необхідно вибрати мову програмування з урахуванням компетентності вчителя, інтересів учнів, їх спрямованості та структури навчання інформатики в школі.

Сьогодні в багатьох школах проводиться вивчення мови Pascal, яка є гарною з методичної точки зору для вивчення основних принципів програмування. Мова Pascal є навчальною структурною мовою програмування, яка передбачає не тільки вивчення алгоритмічних конструкцій, формування логічного і алгоритмічного мислення в учнів, а й рішення складних технологічних і виробничих завдань. І тільки учні невеликого числа шкіл вивчають мови програмування Visual Basic, C, C ++, Delphi, Java та інші [19].

Наступним важливим етапом при побудові системи вивчення теми «Алгоритми та програми» є організація самого процесу навчання. В якості основних видів організації процесу навчання учнів на уроках інформатики можна виділити традиційні види занять: уроки, та практичні заняття. Для проведення занять необхідно вибрати відповідне навчально-методичне, технічне та програмне забезпечення [6].

Вивчення теми «Алгоритми та програми» на базовому рівні пропонуємо починати з 8-го класу, якщо існує можливість побудови такого процесу навчання інформатики в школі. Спочатку необхідно вивчати алгоритми на формальних мовах і використовувати їх на формальних виконавців [16].



У 9-му класі учням необхідно освоїти поняття мови програмування, розглянути класифікацію мов програмування, їх необхідність і спрямованість. Потім розглядаються основи структурного програмування на одному алгоритмічній мові програмування спільно з основними алгоритмічними конструкціями, для цього найкраще вибрати мову Python. Починається вивчення мови програмування зі знайомства з основними компонентами даної мови (константи, ідентифікатори, змінні, типи даних, принципи запису математичних виразів, складовою оператор, порожній оператор, коментарі), основними стандартними процедурами і функціями, структурою програми на мові Python, описується кожен розділ опису мови. Далі послідовно вивчаються основні алгоритмічні конструкції: лінійна конструкція, розгалужується конструкція (умовний алгоритм), оператор множинного вибору, циклічна конструкція (цикли з параметром, з передумовою та з умовою поста) [17].

Спочатку учням дається конструкція на мові блок-схем, а потім реалізація у вигляді програм на обраною мовою програмування. Кожна конструкція закріплюється самостійним рішенням учнями як загальних, так і індивідуальних завдань на практичних заняттях.

У 10 класі необхідно повторити основні конструкції мови і перейти до вивчення теми «Процедури і функції». Далі рекомендується розглянути поняття локальних і глобальних змінних, їх відмінності, способи реалізації та використання на основі процедур і функцій. При вивченні процедур і функцій на практичних заняттях учні можуть пригадати основні алгоритмічні конструкції. Потім здійснюється перехід на вивчення теми «Масиви», під час практичних занять учні повинні оволодіти вмінням формування та роботою з масивами за допомогою процедур і функцій. Під час вивчення цієї теми учні закріплюють знання алгоритмічних конструкцій і їх використання при вирішенні задач різного рівня складності.

У 11-му класі на базовому рівні учні знайомляться зі строковим і символьним типом даних. Вивчають принципи роботи з символьними і рядковими змінними, також реалізуючи алгоритмічні конструкції.

Додатково розглядаються такі структури даних, як записи і безлічі. Вивчення мови програмування закінчується освоєнням поняття файлової змінної, повторенням і вирішенням завдань складнішого рівня, реалізуючи основні методи роботи з даними. Завдяки такій побудові вивчення мови програмування на базовому рівні дається можливість учням освоїти основи мови Python, сприяє розвитку технологічного, алгоритмічного і творчого мислення учнів, формує основні навички програмування [24].

Перед учителями інформатики постає проблема вибору мови програмування для вивчення теми «Основи алгоритмізації та програмування». Досвідчені вчителі звертають увагу на те, що мова програмування – це всього лише мова. Мови програмування відрізняються лише синтаксисом. Головне не синтаксис, а алгоритмічне мислення. Якщо його не має, то немає навичок програмування. Якщо наявне алгоритмічне мислення, то й не буде проблемою перехід з однієї мови програмування на іншу [28].

На практиці використовуються не мови, а системи програмування, тобто системи для розробки нових програм на конкретній мові програмування. Для популярних мов програмування (C, C++, Паскаль, Бейсик, Фортран та ін.) існує багато таких систем програмування. Почали використовуватись системи програмування на мові Java, які дозволяють створювати програми, що викликаються під час перегляду Web-сторінок у глобальній електронній мережі Internet [33].

З нашої точки зору, потрібно зосередитися на освітній системі на підставі віддалених лекцій і курсів за прикладом: <https://www.coursera.org/>.

На сучасному етапі є досить багато літератури з програмування саме на Паскалі відомі посібники Караванової Т.В.: «Основи алгоритмізації та програмування. 777 задач з рекомендаціями та прикладами»; «Інформатика. Методи побудови алгоритмів та їх аналіз. Необчислювальні алгоритми»; «Інформатика. Методи побудови алгоритмів та їх аналіз. Обчислювальні алгоритми» [23].

У нашому дослідженні ми розглядаємо розділ «Алгоритми та

програми» у 8-9 класах. Ми використовували підручник Н.В.Морзе, О.В. Барна, В.П. Вембер, О.Г. Кузьмінська «Інформатика».

В таблиці 2.1 наведено розподіл годин на розділи навчальної програми для учнів 8-9 класів загальноосвітніх навчальних закладів, зі змінами згідно наказу Міністерства освіти та науки України від 07.06.2017р. № 804.

Таблиця 2.1

**Розподіл годин навчальної програми з Інформатики для учнів  
8-9 класів загальноосвітніх навчальних закладів**

№	Назва розділу	Класи і кількість годин		Всього
		8 кл.	9 кл.	
1	Інформаційні технології у суспільстві	0	3	3
2	Математичні основи обчислювальної	6	0	6
3	Кодування даних	10	0	10
4	Комп'ютер як універсальний пристрій для опрацювання даних	12	0	12
5	Інформаційні технології			
5.1	Створення та опрацювання текстових документів (електронний документообіг,	8	12	20
5.2	Створення та опрацювання графічних	6	0	6
5.3	Створення та опрацювання об'єктів	8	0	8
5.4	Створення та опрацювання	8	0	8
5.5	Технології опрацювання числових даних у середовищі табличного процесора	14	0	14
5.6	Система управління базами даних	0	16	16
7	Мережні технології	0	18	18
8	Основи інформаційної безпеки	0	4	4
9	Створення персонального навчального	0	6	6
10	Моделювання	0	8	8
11	Основи алгоритмізації та програмування	56	56	112
12	Розв'язування компетентнісних задач, виконання навчальних проектів	8	12	20
13	Резерв часу	4	5	9
	Всього	140	140	280

Звернемо увагу на те, що на алгоритмізацію, моделювання, кодування даних, математичні основи обчислювальної техніки відведена значна кількість годин.

Тепер розглянемо найбільш поширені в українській школі мови програмування і середовища розробки програм з огляду на визначені нами критерії.

Здебільшого нині використовуються такі універсальні мови

програмування як Basic, Pascal, C++, C#. Іноді використовуються більш екзотичні для школи мови Java, JavaScript. Basic і Pascal вважаються найбільш простими для початкового вивчення. Програмування на C++, C#, Java вже для найпростіших програм потребує використання класів. Ці мови є об'єктно орієнтованими у своїй суті, мають більше можливостей для роботи над реальними проектами. Попри це, більшість ІТ-компаній виставляють вакансії для спеціалістів саме з цих мов програмування. Мова JavaScript цікава тим, що застосовується для веб-програмування, яке теж є досить популярним нині.

З точки зору реальної ІТ-індустрії, Pascal — застаріла мова програмування. Нині майже не створюється нових проєктів з її використанням. Хоча, при цьому, існує ціла низка діючих програмних продуктів, створених за допомогою Delphi, і необхідні спеціалісти, які можуть їх супроводжувати. З іншого боку, розпочати програмувати на Pascal'і дуже просто. Найпростіша програма складається всього з трьох рядків і не потребує введення складних понять. Мабуть найбільшою перевагою цієї мови є той факт, що це мова програмування із строгим приведенням типів. Це означає, що середовище програмування не дозволить використовувати рядкову величину для присвоєння числовій змінній без явної вказівки програміста. Таку помилку можна буде побачити ще на етапі компіляції.

Як показує практика, навички, отримані під час вивчення Pascal'ю дозволяють легко перейти на будь яку сучасну мову програмування.

Basic – це мова програмування, яка розроблена саме для навчання початківців. Досить легко почати програмувати, як і у випадку з Pascal'ем. Крім того, такі реалізації як VisualBasic.NET дозволяють створювати реальні проєкти. Але існує думка, що для тих, хто звик програмувати цією мовою, перехід до інших мов несе певні труднощі. Здебільшого це стосується старих реалізацій.

Якщо йде мова про підготовку до олімпіади з програмування, то слід зауважити, що в більшості регіонів України на III-му, обласному етапі вже використовується міжнародна система перевірки розв'язків e-judge. Ця система підтримує лише два компілятори FreePascal та C++. При цьому програми учнів повинні працювати у консольному режимі.

Щодо JavaScript, то синтаксис цієї мови нагадує синтаксис C++ та

Java. У найпростішому варіанті не треба окремого середовища програмування. Можна використовувати найпростіший текстовий редактор і браузер для виконання програм. Але в такому випадку з'являється багато проблем з відлагодженням коду [12].

Microsoft анонсувала нове середовище розробки програм TouchDevelop. В основі лежить мова програмування JavaScript, але головне те, що це середовище надає можливість програмувати без клавіатури, лише за допомогою сенсорних пристроїв. Так, можна програмувати за допомогою смартфона або електронної дошки. Процес створення програми відбувається досить швидко. Використання існуючих бібліотек дає можливість як імітувати черепашку з Лого, так і створювати професійні додатки і публікувати їх в магазині Windows Store. Але є і певні проблеми. Так, наприклад, робота з масивами ускладнена тим, що як таких масивів у мові немає. Але є інші структури даних, такі як колекції й об'єкти доступу до даних, які можна використовувати як заміну масивам [7].

Що стосується конкретних середовищ розробки програм, то їх величезна кількість не дозволить розглянути їх у рамках цієї статті. Крім того, існують різні платформи: Windows, Linux, iOS, під кожно з яких розроблені середовища програмування. Тож, вибір буде залежати від комп'ютерної техніки, встановленої в школі, операційної системи, можливостей фінансування та інших причин, що були зазначені вище.

Методика вивчення мови програмування спирається на структуру самого об'єкта вивчення – мови програмування, яку представлено на рисунку 2.1:

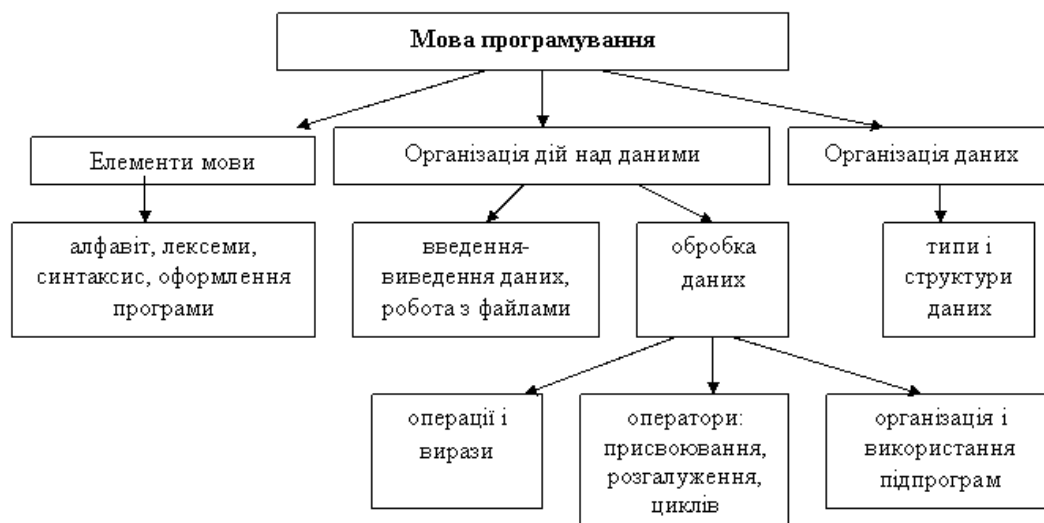


Рис. 2.1. Схема вивчення мови програмування

Основний метод вивчення – демонстрація мови на прикладах простих програм з короткими коментарями.

Для виконання учнями нескладних самостійних завдань на першому етапі досить діяти методом «за зразком» [23].

Проблему як зв'язати вивчення методів побудови алгоритмів роботи з величинами і мови програмування можна вирішити 2 варіантами:

1. Спочатку розглядаються різноманітні алгоритми, для опису яких використовуються блок-схеми й алгоритмічна мова, а потім – правила мови програмування, способи перекладу вже побудованих алгоритмів у програму на цій мові;

2. Алгоритмізація і програмування освоюються паралельно.

Бажано, щоб учні як раніше одержали можливість перевірити правильність своїх алгоритмів, працюючи на комп'ютері. Але навіть при використанні комп'ютера, спочатку, рекомендується не відмовлятися від ручного трасування [23].

Основний методичний принцип вивчення даної теми – принцип виконавця. В якості виконавця виступає комп'ютер. Робота комп'ютера складається з 3 складових: створення програми, налагодження програми, виконання програми.

Учитель, розповідаючи про режими роботи системи програмування, повинен розповідати про систему команд, що використовується в кожному з режимів.

Наприкінці вивчення розділу «Введення до програмування» проходить урок узагальнення знань. На цьому уроці піднімається питання про різні класи мов програмування, про парадигми програмування, про способи трансляції, робляться висновки про особливості та характерні риси структурного підходу в програмуванні та об'єктно-орієнтованого [23].

Більш детально вивчення розділу «Програмування» виноситься для профільного курсу програмування.

У нашому дослідженні ми дамо методичні рекомендації до розв'язування задач з курсу алгоритмізації та програмування.

Навчання складанню алгоритмів роботи з величинами повинне проводитися на прикладах типових задач з поступовим ускладненням структури алгоритмів. Послідовність завдань планується згідно складності та послідовності вивчення алгоритмічних структур:

- лінійні алгоритми (обчислення по формулах, будь-які пересилання значень змінних);
- розгалужені алгоритми (пошук найбільшого чи найменшого значень з декількох даних, сортування 2-3 значень, діалог з розгалуженнями);
- циклічні алгоритми (обчислення сум і добутків числових послідовностей, циклічне введення даних з наступною їх обробкою).

Вивчення мови програмування Паскаль відбувається в контексті розв'язуваних задач, тобто нові засоби мови вводяться по мірі необхідності для розв'язування чергового типу задач. Приклади завдань, які можна запропонувати учням розглянемо у наступному розділі.

Узагальнюючи вище наведене, можна стверджувати, знайомство з програмуванням у ранньому віці дає учням базове розуміння не лише того, як застосовувати технології на практиці, але й того, що можна створити з їхньою допомогою, навіть не будучи фахівцем у сфері ІКТ. Ми й так оточені пристроями, керованими комп'ютерами, а тому просто зобов'язані вникати у принципи їх роботи та впливу на наше повсякденне життя [23].

Зрозуміло, що за нинішнього вибору мов програмування істотну роль відіграють особисті уподобання, тенденції, слідування певним вимогам в освітньому середовищі з точки зору вже вивчених і перевічених на практиці методик. Так, Scratch, наприклад, популярний серед учнів початкової школи, так як простий для розуміння. А Python – серед учнів середньої та старшої школи, так як при бажанні може використовуватися для створення навіть наукових додатків.

Власне, про який би вибір не йшла мова, тут важливо інше: написання й подальше виконання інструкцій у рамках будь-якої з програм дає негайний зворотний зв'язок щодо того, чи правильно учень виконав усі необхідні дії для того, щоб отримати очікуваний результат. У кінцевому рахунку розуміння того, як досягти заданої мети найбільш точно та ефективно, набагато важливіше, ніж деталі самої мови

програмування. Адже у кожній із них лежать алгоритми, а робота з ними формує у дітей унікальне алгоритмічне (обчислювальне) мислення, власники якого, як люблять говорити експерти, і стануть справжнім порятунком для майбутнього прогресуючої цифрової цивілізації.

## **2.2. Порівняльна характеристика мов програмування для учнів 8-9 класів на приклади розв'язування задач на мовах Java, C++ та Python**

Розглянемо приклади розв'язування кількох типових для уроків інформатики 8-9 класу задач на мовах Java, C++ та Python. Однією з найпопулярніших задач, які розв'язують при вивченні основ програмування та алгоритмізації, є знаходження найбільшого спільного дільника (НСД) – найбільше число, на яке без остачі діляться два заданих числа. Для цього використовують Алгоритм Евкліда. Цей алгоритм є гарним прикладом для навчання, бо він допоможе нам проілюструвати перераховані нижче важливі моменти, які повинен знати учень після вивчення теми «Алгоритми»:

1) Кожен крок алгоритму повинен бути чітко і однозначно визначеним. Ця вимога є обов'язковою і не може бути порушена ні за яких обставин.

2) Повинні бути точно вказані діапазони допустимих значень вхідних даних, які обробляються за допомогою алгоритму.

3) Один і той же алгоритм можна представити декількома різними способами.

### ***Розв'язання задачі на мові програмування Java:***

```
import java.util.Scanner;

class Main {
    public static void main(String args[]) {

        Scanner input = new Scanner(System.in);

        int a = input.nextInt();
        int b = input.nextInt();

        System.out.println(egcd(a,b));
```



```

    input.close();
}

public static int egcd(int a, int b) {
    if (a == 0)
        return b;

    while (b != 0) {
        if (a > b)
            a = a - b;
        else
            b = b - a;
    }

    return a;
}
}

```

### ***Реалізація алгоритму на мові C++:***

```

#include <iostream>
using namespace std;

int NOD(int n1, int n2)
{
    int div;
    if (n1 == n2)
        return n1;
    int d = n1 - n2;
    if (d < 0)
    {
        d = -d;
        div = NOD(n1,d);
    }
    else
    {
        div = NOD(n2, d);
    }
    return div;
}

int main()
{
    int n1, n2;
    cout << "n1=";
    cin >> n1;
    cout << "n2=";
    cin >> n2;
    cout << NOD(n1, n2);
    cin.get(); cin.get();
    return 0;
}

```

***Розв'язання задачі на мові Python має наступний вигляд:***

```

a = int(input())
b = int(input())

while a != 0 and b != 0:
    if a > b:
        a %= b
    else:
        b %= a

gcd = a + b
print(gcd)

```

Розглянемо також задачу, яка подана у підручнику «Інформатика» для 8 класу [16] під час вивчення розділу «Алгоритми роботи з об'єктами та величинами».

#### Задача. Порівняння

Члени математичного гуртка ознайомились із поняттям факторіала числа – добутку натуральних чисел від 1 до даного числа:  $n! = 1 * 2 * 3 * \dots * n$ .

Розробіть проект у визначеному середовищі програмування, за допомогою якого можна буде порівняти значення факторіала числа та його квадрата [16, с.180].

#### ***Розв'язання задачі на мові Java:***

```

import java.util.Scanner;

class Main {
    public static void main(String args[]) {

        Scanner input = new Scanner(System.in);
        int number = input.nextInt();

        int factorial=calculateFactorial(number);
        int square=calculatrSquareNumber(number);

        if(factorial>square)
            System.out.println("Факторіал числа n більший, ніж квадрат числа n");
        if(factorial<square)
            System.out.println("Факторіал числа n менший, ніж квадрат числа n");
        if(factorial==square)
            System.out.println("Факторіал числа n і квадрат числа n рівні");

    }

    static int calculateFactorial(int n){

```

```

        int result = 1;
        for (int i = 1; i <=n; i ++){
            result = result*i;
        }
        return result;
    }

static int calculatrSquareNumber(int n){
    return (n*n);
}
}

```

### **Розв'язання задачі на мові C++:**

```

#include <iostream>

int main()
{
    int n;

    std::cin >> n;

    int fact=factorial(n);
    int sqr = square(n);

    if(fact>sqr)
        std::cout << ("Факторіал числа n більший, ніж квадрат числа n");
    if(fact<sqr)
        std::cout << ("Факторіал числа n менший, ніж квадрат числа n");
    if(fact==sqr)
        std::cout << ("Факторіал числа n і квадрат числа n рівні");

    std:: cin.get();
    return 0;

}

int factorial(int i)
{
    if (i==0) return 1;
    else return i*factorial(i-1);
}

int square(int i){
    return i*i;
}

```

### **Розв'язання задачі на мові Python:**

```
n = int(input())
```

```

square = n*n;

factorial = 1
while n > 1:
    factorial *= n
    n -= 1

if factorial>square:
    print("Факторіал числа n більший, ніж квадрат числа n")
elif factorial<square:
    print("Факторіал числа n менший, ніж квадрат числа n")
elif factorial==square:
    print("Факторіал числа n і квадрат числа n рівні")

```

Розглянемо також задачу, яка подана у підручнику «Інформатика» для 9 класу [17] під час вивчення розділу «Табличні величини та алгоритми їх опрацювання».

#### Завдання 2. Кислотність ґрунтів

На 10 земельних ділянках було проведено обстеження рівня рН ґрунту.

Результати обстеження подано в таблиці:

```

1 2 3 4 5 6 7 8 9 10
5 6 7 3 9 8 4 5 7 5

```

У вибраному середовищі програмування створіть проект Ґрунти, у якому можна визначити кількість земельних ділянок з кислими (рН < 7), з нейтральними (рН = 7) та лужними (рН >7) ґрунтами.

#### Розв'язання задачі на мові Java:

```

public class Main {
public static void main(String args[]) {
    int[] a= {5,6,7,3,9,8,4,5,7,5};

    int kk=0;
    int kn=0;
    int kl=0;

    for(int i=0; i<a.length;i++){
        if(a[i] < 7)
            kk+=1;
        else if (a[i]==7)
            kn+=1;
        else

```

```

    kl+=1;
}
System.out.println(kk+" ділянок з кислим ґрунтом");
System.out.println(kn+" ділянок з нейтральним ґрунтом");
System.out.println(kl+" ділянок з лужним ґрунтом");
}
}

```

### Розв'язання задачі на мові C++:

```

#include <iostream>

int main() {

    int array[] {5,6,7,3,9,8,4,5,7,5};
    int size = sizeof(array)/sizeof(array[0]);

    int kk=0;
    int kn=0;
    int kl=0;

    for(int i=0; i<size;i++){
        if(array[i] < 7)
            kk+=1;
        else if (array[i]==7)
            kn+=1;
        else
            kl+=1;
    }

    std::cout << kk << " ділянок з кислим ґрунтом\n";
    std::cout << kn << " ділянок з нейтральним ґрунтом\n";
    std::cout << kl << " ділянок з лужним ґрунтом";

    return 0;
}

```

### Розв'язання задачі на мові Python:

```

a=[5,6,7,3,9,8,4,5,7,5]
kk=0 //змінна для ділянок з кислим ґрунтом
kn=0 //змінна для ділянок з нейтральним ґрунтом
kl=0 //змінна для ділянок з лужним ґрунтом

for i in range (10):
    if a[i]<7:
        kk+=1
    elif a[i] == 7:
        kn+=1
    else:
        kl+=1

print(kk,'ділянок з кислим ґрунтом')

```

```
print(kn, 'ділянок з нейтральним ґрунтом')  
print(kl, 'ділянок з лужним ґрунтом')
```

Узагальнюючи вище наведене, можна стверджувати, що ми провели дослідження взявши задачі для учнів 8-9 класів та розклавши на мови програмування Java, C++ та Python. На уроках інформатики учні вчать мову Python ми додали ще мови Java, C++. Як ми можемо побачити з даних прикладів, саме програми на мові Python мають більш лаконічний вигляд, не містять надмірності. Це ще раз доводить, що саме ця мова є найкращим вибором для вивчення програмування у школі.

### **2.3. Організація та проведення педагогічного експерименту**

Педагогічний експеримент проходив у Чаплінській спеціалізованій школі I-III ступенів Чаплінської селищної ради Херсонської області, під час проходження виробничої практики. Мета педагогічного експерименту полягала у перевірці ефективності теоретично обґрунтованих методичних рекомендацій добору і використання мов програмування при вивченні розділу «Алгоритми та програми» у 8-9 класах.

Основними завданнями педагогічного експерименту були:

- розробити матеріал, для учнів 8-9 класів при вивченні розділу «Алгоритми та програми»;
- впровадження розробленого методичного забезпечення у навчально-виховний процес під час вивчення розділу «Алгоритми та програми» у 8-9 класах;
- здійснення якісного аналізу та статистичного оцінювання результатів проведеного експерименту і формулювання висновків.

Також нами були розроблені завдання для різних мов програмування та запроваджені на уроках інформатики для учнів 8-го при вивченні розділу «Алгоритми роботи з об'єктами та величинами» та

9-го класів при вивченні розділу «Табличні величини та алгоритми їх опрацювання».

У запропонованих завданнях учням потрібно спробувати трьома різними мовами програмування розглянути одну задачу. Також після цього ми провели опитування серед учнів з метою визначення, яка з запропонованих мов була найбільш зрозумілою та доступною для них.

В експерименті брали участь учні 8 класу -17 осіб, та учні 9 класу – 18 осіб.

Перед проведення уроків 8-9 класах, ми провели зріз знань з розділу «Алгоритми та програми»:

- Високий рівень (11-12 балів) – 0 осіб;
- Достатній рівень (8-10 балів ) – 23 особи;
- Середній рівень (5-7 балів) – 10 осіб;
- Початковий рівень (1-4 балів)- 0 осіб.

На підсумкових заняттях педагогічного експерименту проведено зріз знань, на якому учні 8-9 класів показали наступні досягнення:

- Високий рівень (11-12 балів) – 16 осіб;
- Достатній рівень (8-10 балів ) – 17 особи;
- Середній рівень (5-7 балів) – 0 осіб;
- Початковий рівень (1-4 балів)- 0 осіб.

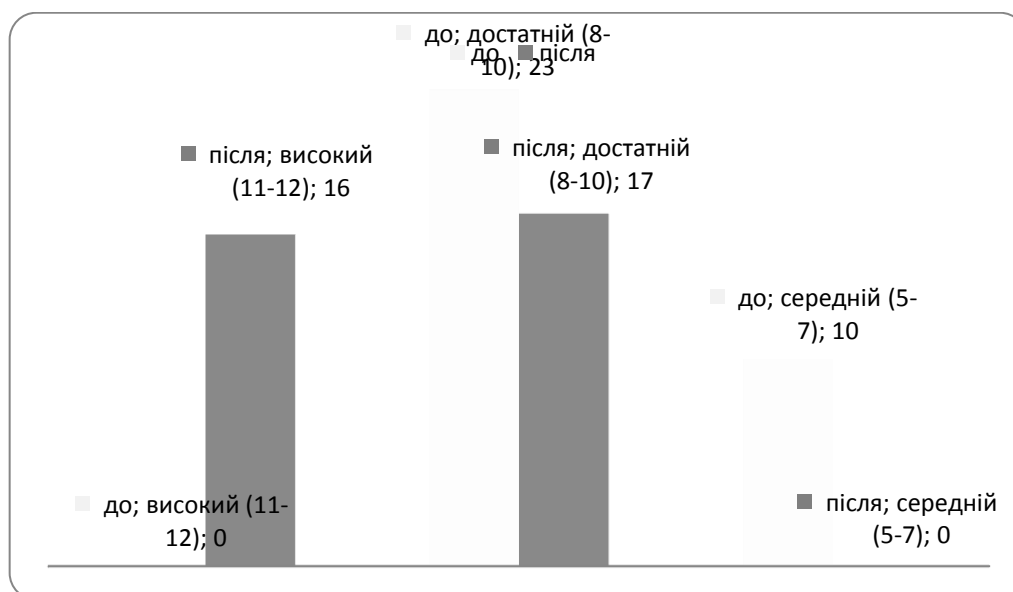


Рис. 2.2. Порівняння успішності учнів

Для порівняння результатів ми побудували наступну діаграму (рис.2.2.).

Аналізуючи дані графіки, ми можемо сказати, що результати учнів після проведення педагогічного експерименту покращилися, в порівнянні з результатами до проведення експерименту.

Також у ході експерименту нами була проведено опитування учнів, ми запитали про загальне ставлення до програмування та зацікавленість у вивченні даного предмету. На рис. 2.3 подано результати опитування учнів 8-9 класів (вибірка 33 особи), наскільки цікавим для них є вивчення програмування на уроках Інформатики?

- 1) дуже цікаво
- 2) 50/50
- 3) не цікаво

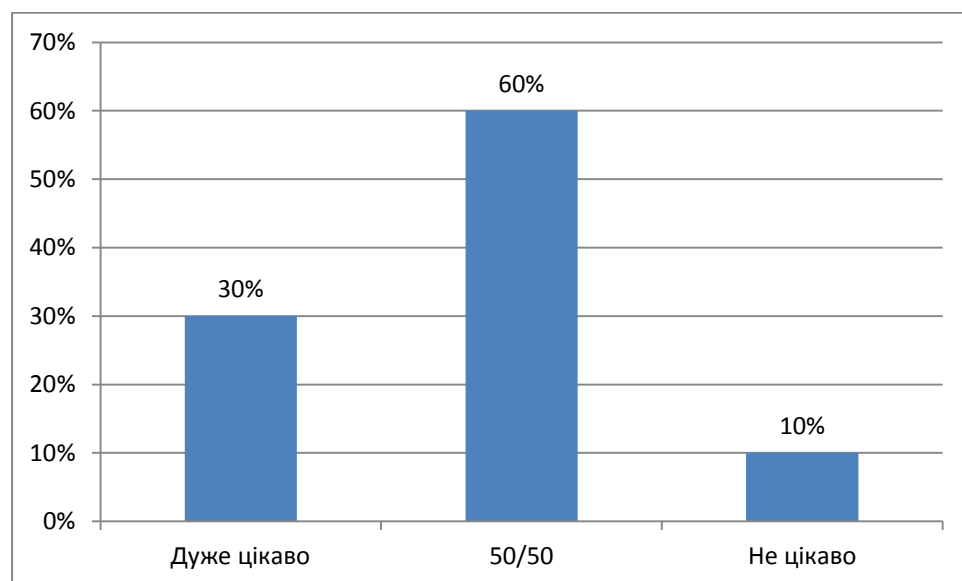


Рис. 2.3. Зацікавленість учнів у вивченні програмування у школі

Загалом можна зробити висновок, що учні в різній мірі готові використовувати різні мови програмування (Java, C++ та Python) в навчанні, проте більша частина учнів лише на 50 відсотків зацікавлена у вивченні програмування лише тому, що не всі учні чітко знають та розуміють для чого воно їм потрібно та які мови з якою метою використовувати.

Нами було досліджено також питання, Які мови програмування ви вивчаєте самі? Python, JavaScript, Arduino, C#, PHP, C++, Logo.



Зокрема ми намагалися з'ясувати наявність в учнів навичок з мов програмування які не вчать на уроках інформатики.

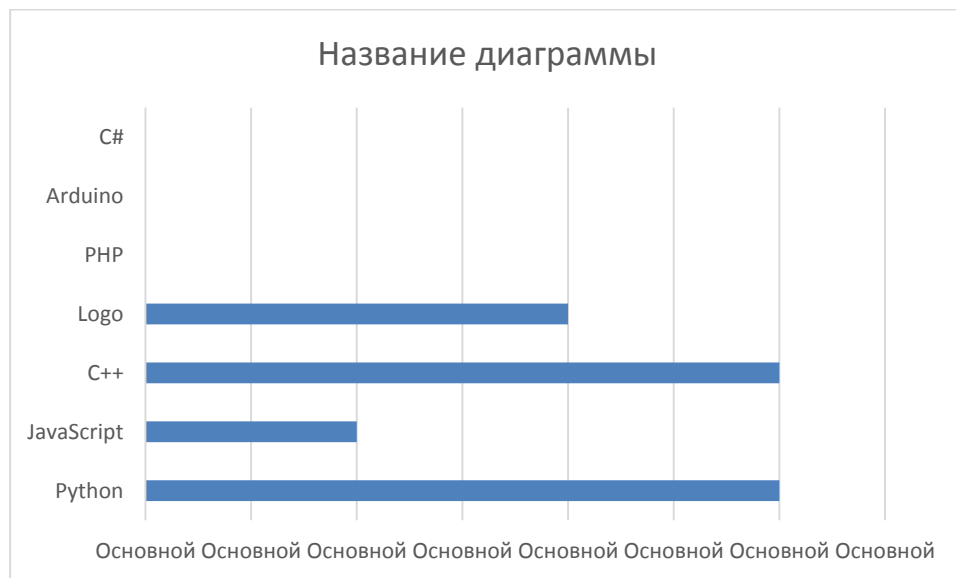


Рис. 2.4 Які мови програмування ви вивчаєте самі

Аналізуючи відповіді респондентів, ми переконалися, що більшість учнів проявляють повну або часткову зацікавленість у вивченні програмування.

Аналіз даних, отриманих у результаті педагогічного експерименту, дає підстави для висновку, що розроблені задачі різними мовами програмування при вивченні розділу «Алгоритми та програми» для учнів 8-9 класів для подальшого вивчення мов програмування та виконання практичних робіт дала позитивний результат.

## ВИСНОВКИ

Середовища програмування – важливий засіб навчання програмування учнів загальноосвітніх шкіл. На початковому етапі вивчення конкретної мови програмування середовище програмування є об'єктом вивчення і лише потім – дидактичним засобом.

На основі аналізу програмно-технологічних і психолого-педагогічних умов вибору середовища програмування визначено такі середовища, застосування яких більш ефективно сприятиме формуванню в учнів навичок створення програм. Причому, для програмістів-початківців доцільно використовувати одні середовища, а для учнів, що вже мають навички програмування – інші.

У тому випадку, коли планується вивчати кілька мов програмування, доцільно використовувати таке середовище програмування, яке надає можливості розроблення програм більшістю даних мов програмування.

1. Аналіз науково-методичної літератури засвідчив, що питанню добору і використання мов програмування при вивченні розділу «Алгоритми та програми» присвячені роботи багатьох вітчизняних та зарубіжних науковців. Аналіз сучасного стану викладання програмування у школі показує нам, що такі мови, як Pascal або Delphi, які зараз є найбільш розповсюдженими для вивчення у школах України, вже втрачають свою актуальність у сучасному світі, тому найкращим варіантом буде обрати одну з сучасних мов програмування, які застосовуються на підприємствах ІТ-індустрії. Саме тому в нашій дипломній роботі було опрацьовано такі мови, як Java, C++ та Python. Ми розглянули переваги та недоліки кожної з них.

2. Проведене дослідження показує, що мова програмування Python є найкращим варіантом для вивчення розділу «Алгоритми та програми» у 8-9 класах. Дана мова має досить простий та зрозумілий синтаксис, не має надмірності, легка в опануванні. Вона має найбільший вибір простих, але зручних середовищ, серед яких є ті, які встановлюються на

комп'ютер, так і online-середовища. Це робить мову Python ще більш привабливою для навчання учнів середньої школи. Мови програмування Java та C++ краще пропонувати учням для вивчення на факультативних заняттях або у школах з поглибленим вивченням інформатики.

3. З метою перевірки ефективності розроблених методичних рекомендацій було проведено педагогічний експеримент, в якому взяли участь учні Чаплінської спеціалізованої школи I-III ступенів Чаплінської селищної ради Херсонської області. Він показав, що саме мова програмування Python є найбільш легкою та зрозумілою для учнів середньої школи.

На нашу думку, використовувати це середовище для навчання програмування мовою Python недоцільно через його високі системні вимоги. У той же час вивчення даного середовища є бажаним, якщо мова Python є лише початковим етапом у вивченні програмування і планується поступовий перехід до потужніших і сучасніших мов.

Узагальнюючі результати дослідження можна стверджувати, що розроблені нами методичні рекомендації щодо можливостей добору та використання мов програмування під час вивчення розділу «Алгоритми та програми» є ефективними та можуть бути використані учителями під час навчального процесу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Антоненко, В.А. Програмне забезпечення для ефективного вивчення школярами основ алгоритмізації та програмування. /В.А.Антоненко Актуальні питання сучасної інформатики, АПН України: Матеріали наукової конференції 2020 - С.179-182.
2. Базурін В. М. Середовища програмування як засіб навчання учнів основ програмування / В. М. Базурін // Інформаційні технології і засоби навчання. - 2017. - Т. 59, вип. 3. - С. 13-27. [Електронний ресурс]. – Режим доступу: [http://nbuv.gov.ua/UJRN/ITZN\\_2017\\_59\\_3\\_4](http://nbuv.gov.ua/UJRN/ITZN_2017_59_3_4)
3. Васильєва, О.К. Методичні основи формування компетентностей учнів з основ алгоритмізації. / О.К. Васильєва Актуальні питання сучасної інформатики, АПН України: Матеріали наукової конференції, 2020, - С. 69-70.
4. Вінник В.Ю. Алгоритмічні мови та основи програмування: мова С / В. Ю. Вінник. – Житомир : ЖДТУ, 2017. – 328 с.
5. Григорович В.Г. Методичні вказівки до виконання лабораторних і практичних завдань з курсу «Алгоритмізація та програмування». Частина 1 : навч. посібник / В. Г. Григорович. – Дрогобич: Редакційно-видавничий відділ Дрогобицького державного педагогічного університету імені Івана Франка, 2016. – 1400 с.
6. Дейтел Х.М. Как программировать на С++ / Х. М. Дейтел, П. Дж. Дейтел. – М. : Бином-Пресс, 2020. – 1248 с.
7. Державний стандарт повної загальної середньої освіти (Постанова КМУ № 898 від 30.09.2020 року): [Електронний ресурс]. – Режим доступу: <https://www.kmu.gov.ua/npas/pro-deyaki-pitannya-derzhavnih-standartiv-povnoyi-zagalnoyi-serednoyi-osviti-i300920-898>
8. Климчук В.В. Методика вивчення об'єктно-орієнтованого програмування. Інформаційно-комунікаційні технології в освіті/ В.В. Климчук – К.: УОВЦ «Оріон», 2018. [Електронний ресурс]. – Режим доступу: <https://e-journals.npu.edu.ua/index.php/ikt/article/view/142> 2018, 5.

9. Ковалюк Т. В. Алгоритмізація та програмування: підручник з грифом МОН України / Т. В. Ковалюк. – Львів : Магнолія-2018. – 400 с.
10. Кравець П. Об'єктно-орієнтоване програмування: навч. посібник / П.О. Кравець. – Львів: Видавництво Львівської політехніки, 2021. – 624 с.
11. Маловічко О.П., Конюхов С.Л. Застосування спеціалізованого педагогічного програмного комплексу у процесі вивчення програмування у восьмому класі./ О.П.Маловічко, С.Л.Конюхов, Ukrainian Journal of Educational Studies and Information Technology, 2017.- С.38-55.
12. Методика вивчення мови програмування. Методика навчання учнів роботі з допоміжним алгоритмом, застосування рекурсії. Особливості вивчення систем візуального програмування. (Лекція) [Електронний ресурс] / [http://vvpk.at.ua/files/el\\_metod/Lektsiya\\_8.htm](http://vvpk.at.ua/files/el_metod/Lektsiya_8.htm)
13. Мартынов Н. Н. Программирование для Windows на C/C++ / Н. Н. Мартынов. – М. : Бинум-Пресс, 2018. – 528 с.
14. Мейн М. Структуры данных и другие объекты в C++ / М. Мейн, У. Савитч. – М. : Вильямс, 2020. – 832 с.
15. Мельник А. Архітектура комп'ютерів: підручник / А. Мельник. – Луцьк : Волинська обласна друкарня, 2008. – 470 с.
16. Морзе Н. В. Методика навчання інформатики: Навч. посібник: У 4 ч. / За ред. акад. М. І. Жалдака. - К.: Навчальна книга, 2004. — 256 с.
17. Морзе Н.В. Інформатика: підруч. для 8 кл. загальноосвіт. навч. Закладів / Н.В. Морзе, О.В.Барна, В.П. Вембер. – К.: УОВЦ «Оріон», 2016. – 240 с. :іл.
18. Морзе Н.В. Інформатика: підруч. для 9 кл. загальноосвіт. навч. Закладів / Н.В. Морзе, О.В.Барна, В.П. Вембер. – К.: УОВЦ «Оріон», 2017. – 240 с. :іл.
19. Пасіхов, Ю. (2016). Концепція шкільної ІТ-освіти в умовах сучасного інформаційного суспільства. Матеріали X Міжнародної науково-практичної конференції «Інтернет –Освіта –Наука», Вінниця, 11-14 жовтня Вінниця: ВНТУ 2020, - С.157-158..
20. Рішення колегії МОН України №10 від 27.10.2019 "Нова українська школа: концептуальні засади реформування середньої школи": [Електронний ресурс] / За заг. ред. Грищенко М. – К.: МОН

України, 2019. – 40 с. – Режим доступу: – <https://base.kristti.com.ua/?p=1129>

21. Співаковський О.В. Стан та перспективи розвитку автоматизованої системи управління у Херсонському педагогічному університеті // Комп'ютер у школі та сім'ї. – 2001. - №2. - С. 26 - 32.

22. Семко Л. Методичні підходи до вивчення інформатики в основній школі / Н. Самойленко, Л. Семко // Наукові записки. Випуск 7— Серія: Проблеми методики фізико-математичної і технологічної освіти. Частина 2. — Кіровоград: РВВ КДПУ ім. В. Винниченка, 2015. — С.76– 825.

23. Семко Л.П. Формування інформаційних компетенцій на уроках інформатики в основній школі / Н. І. Самойленко, Л. П. Семко // Компетентнісні засади змісту освіти в 11-річній школі : матеріали Всеукр. наук-практ. конф. (28- 29 березня 2018 року / Ред. кол. : Федоренко О. А., Єрмаков І. Г. (науковий редактор), Ратушна А. М. — К. : Оберіг, 2018. — 608 с. — С. 435–439.

24. Системне програмування. Основні поняття [Електронний ресурс].– Режим доступу: [Electronic resource]. – Mode of access : WWW/URL: <https://owncloud.kspu.kr.ua/index.php/s/69e0915776ee3225579f734e986ad017>. – Title from the screen.

25. Трофименко О.Г., Прокоп Ю. В. , Швайко І.Г., Буката Л.М. та ін.С++ Теорія та практика: навч. посібник /О.Г. Трофименко, Ю. В. Прокоп, І.Г. Швайко, Л.М. Буката та ін. ; за ред. О.Г. Трофименко. – 587 с.

26. Шаховська Н.Б. Алгоритми та структури даних / Н.Б. Шаховська, Р.О. Голощук. – Львів: Магнолія-2006. – 2009. – 216 с.

27. Шевчук, П.Г. Проблема вибору мови та середовища програмування в якості засобу навчання. / П.Г. Шевчук, Звітна наукова конференція Інституту інформаційних технологій і засобів навчання АПН України: Матеріали наукової конференції, 2010,- С.30-31.

28. Шеховцов В.А. Операційні системи / В.А. Шеховцов. – К.: Видавнича група ВНУ, 2005. – 576 с.

29. Шпак З.Я. Програмування мовою С / З.Я. Шпак. – Львів: “Оріяна-Нова”, 2006. – 431 с.

30. Васильєв О. Програмування мовою Python / О. Васильєв //Тернопіль: Навчальна книга - Богдан, 2019. – С. 504.
31. D. Weller, Sh. Chikkerur. MIT Course Number 6.087 : Practical Programming in C [Electronic resource]. – Mode of access : WWW/URL : <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-087-practical-programming-in-c-january-iap-2010/index.htm>. – Title from the screen.
32. Download PyCharm: Python IDE for Professional Developers by JetBrains [Електронний ресурс] / JetBrains s.r.o. –2021.–Access mode : <https://www.jetbrains.com/pycharm/download/>.
33. Download Python | Python.org [Електронний ресурс]. – Режим доступу:–Access mode : <https://www.python.org/downloads/>.
34. IntelliJ IDEA The Java IDE [Електронний ресурс]. – Режим доступу: <https://www.jetbrains.com/idea/>.
35. Kruglyk V., Lvov M. Choosing the first educational programming language //ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer: Proceedings of the 8th International Conference ICTERI 2012. – Kherson, 2012. – С. 188-189.
36. Kruglyk, V.; Lvov, M. Teaching algorithmization and programming using python language. Інформаційні технології в освіті, 2014, 20: 13-23.
37. Malovichko, Oksana; Koniukhov, Serhii. Using the Specialized Pedagogical Software for Studying Programming in the Eighth Grade. Ukrainian Journal of Educational Studies and Information Technology, 2017, 5.4: 38-55.
38. Oshanova, Nurzhamal, et al. Algorithmization and programming teaching methodology in the course of computer science of secondary school. Australian Educational Computing, 2019, 34.1.
39. Replit: Python – Python online editor, IDE, compiler, interpreter, and REPL [Електронний ресурс]. – Режим доступу:<https://repl.it/languages/python3>.
40. Shevchuk, Petro H. Основні підходи добору мови та середовища програмування як засобів навчання. Інформаційні технології і засоби навчання, 2010, 17.3.

## Додаток А

## Розподіл навчальних годин на вивчення розділів програми

№	Назва розділу	Класи і кількість годин					Всього
		5 кл.	6 кл.	7 кл.	8 кл.	9 кл.	
1	Інформація, інформаційні процеси, системи, технології	4	–	–	3	2	9
2	Комп'ютер як універсальний пристрій для опрацювання даних	10	6	–	5	–	21
3	Інформаційні технології						
3.1	Створення та опрацювання текстових документів	–	8	–	6	5	19
3.2	Створення та опрацювання графічних зображень	9	–	–	–	6	15
3.3	Створення та опрацювання об'єктів мультимедіа	–	4	–	6	–	10
3.4	Створення та опрацювання комп'ютерних презентацій	9	–	–	–	6	15
3.5	Створення та опрацювання числових даних	–	–	8	10	–	18
4	Комп'ютерні мережі	–	8	4	–	16	28
5	Моделювання	–	–	3	–	6	9
6	Основи алгоритмізації та програмування	–	7	9	28	10	54
7	Розв'язування компетентнісних задач, виконання індивідуальних і групових навчальних проєктів	–	–	8	9	15	32
8	Резерв	3	2	3	3	4	15
	Всього	35	35	35	70	70	245



Опитування

1. Наскільки цікавим для Вас є вивчення програмування на уроках Інформатики?
  - дуже цікаво
  - 50/50
  - не цікаво
2. Чи знаєте ви які мови програмування для чого потрібні?
3. Які мови програмування ви вивчаєте в школі?
4. Які мови програмування ви вивчили самі?  
Python,  
JavaScript,  
Arduino  
C#  
PHP  
C++
5. Скільки часу Ви готові витратити на самостійне вивчення матеріалу?
  - 1-2 годину в день
  - 5-6 годин в день весь день.
  - Я дуже зайнятий
6. Як у вас з англійською мовою?
  - дуже добре
  - добре
  - середній рівень
  - зовсім не знаю
7. На уроці ви отримали досвід роботи з мовами програмування?
8. Чи вдалося вам написати мовою програмування хоч одну програму?
9. Що ви можете сказати про проведений урок вирішення завдань різними мовами програмування?
10. Яка мова програмування далася вам найважче?
11. Яка мова програмування далася вам найлегше?
12. Ви б хотіли і далі вивчати програмування?
13. Чи хочете Ви пов'язати своє життя з ІТ після навчання в школі?
  - так
  - ні
  - ще не визначився

## Конспект уроку

### Алгоритми опрацювання табличних величин: знаходження суми значень елементів Інформатика 9 клас

**Алгоритми опрацювання табличних величин: знаходження суми значень елементів; суми або кількості значень елементів, що задовольняють заданим умовам, пошук заданого елемента.**

#### Цілі:

- **навчальна:** сформулювати уявлення про класифікацію задач опрацювання одновимірних масивів; знання стандартних алгоритмів опрацювання одновимірних масивів, вміння використовувати стандартні алгоритми для розв'язування задач;
- **розвивальна:** розвивати здібності до формалізації; виховувати дослідницький підхід у пізнанні нового, формувати вміння застосовувати отримані знання для розв'язування прикладних задач;
- **виховна:** виховувати інформаційну культуру та сприяти розширенню кругозору учнів.

**Тип уроку:** Комбінований.

**Обладнання та наочність:** дошка, комп'ютери з підключенням до мережі Інтернет, підручник, навчальна презентація.

**Програмне забезпечення:** середовище програмування Lazarus, браузер, пакет офісних програм.

#### Хід уроку

##### I. Організаційний етап

- привітання
- перевірка присутніх
- перевірка готовності учнів до уроку

##### II. Актуалізація опорних знань

Фронтальне обговорення понять: опис змінних типу одновимірний масив, задавання значень елементам масиву, введення і виведення значень елементів масиву.

##### III. Формулювання теми, мети й завдань уроку; мотивація навчальної діяльності

*Слово вчителя.* Уявіть себе головним бухгалтером підприємства, який аналізує масив значень заробітної платні співробітників за місяць. Які задачі ви можете поставити при такому аналізі? (Загальна сума виплат; середнє значення; кількість нарахувань із сумою, більшою за певне значення; найбільше значення.) Розв'язування багатьох прикладних задач, що потребують опрацювання сукупності однотипних значень, базується на типових алгоритмах опрацювання масивів. Користуючись типовими алгоритмами опрацювання масивів, ви зможете

створювати ефективні алгоритми розв'язування більш складних прикладних задач.

#### **IV. Сприйняття та усвідомлення нового матеріалу** **Пояснення вчителя**

**Робота з підручником:** § 6.2

#### **VI. Засвоєння нових знань, формування вмінь** **Практичне завдання.**

**Робота за комп'ютером**

1. Повторення правил безпечної поведінки за комп'ютером.
2. Інструктаж учителя.

Підручник ст. 172-173

- *Практична робота за комп'ютерами.*
- *Вправи*

#### **Завдання 2. Кислотність ґрунтів**

На 10 земельних ділянках було проведено обстеження рівня рН ґрунту.

Результати обстеження подано в таблиці:

1	2	3	4	5	6	7	8	9	10
5	6	7	3	9	8	4	5	7	5

У вибраному середовищі програмування створіть проект Ґрунти, у якому можна визначити кількість земельних ділянок з кислими ( $\text{pH} < 7$ ), з нейтральними ( $\text{pH} = 7$ ) та лужними ( $\text{pH} > 7$ ) ґрунтами.

#### **Розв'язання задачі на мові Java:**

```
public class Main {
    public static void main(String args[]) {
        int[] a= {5,6,7,3,9,8,4,5,7,5};

        int kk=0;
        int kn=0;
        int kl=0;

        for(int i=0; i<a.length;i++){
            if(a[i] < 7)
                kk+=1;
            else if (a[i]==7)
                kn+=1;
            else
                kl+=1;
        }
        System.out.println(kk+" ділянок з кислим ґрунтом");
        System.out.println(kn+" ділянок з нейтральним ґрунтом");
        System.out.println(kl+" ділянок з лужним ґрунтом");
    }
}
```

#### **Розв'язання задачі на мові C++:**

```
#include <iostream>
int main() {
    int array[] {5,6,7,3,9,8,4,5,7,5};
    int size = sizeof(array)/sizeof(array[0]);
```

```

int kk=0;
int kn=0;
int kl=0;
for(int i=0; i<size;i++){
    if(array[i] < 7)
        kk+=1;
    else if (array[i]==7)
        kn+=1;
    else
        kl+=1;
}

std::cout << kk << " ділянок з кислим ґрунтом\n";
std::cout << kn << " ділянок з нейтральним ґрунтом\n";
std::cout << kl << " ділянок з лужним ґрунтом";

return 0;
}

```

### **Розв'язання задачі на мові Python:**

```

a=[5,6,7,3,9,8,4,5,7,5]
kk=0 //змінна для ділянок з кислим ґрунтом
kn=0 //змінна для ділянок з нейтральним ґрунтом
kl=0 //змінна для ділянок з лужним ґрунтом

for i in range (10):
    if a[i]<7:
        kk+=1
    elif a[i] == 7:
        kn+=1
    else:
        kl+=1

print(kk,'ділянок з кислим ґрунтом')
print(kn,'ділянок з нейтральним ґрунтом')
print(kl,'ділянок з лужним ґрунтом')

```

### **VII. Підбиття підсумків уроку**

### **VIII. Домашнє завдання**

Підручник § 6.2 ст. 165-167

### **IX Оцінювання роботи учнів**