

Методика вивчення розділу «парадигми та технології програмування»

Кривонос О. М.¹, Мінгальова Ю. І.², Кривонос М. П.³,
Стельмашенко Я. А.⁴, Махенько Я. Д.⁵

Опубліковано	Секція	УДК
18.09.2023	Освіта/Педагогіка	378.147:004.413

DOI: <http://dx.doi.org/10.5281/zenodo.8358944>

Ліцензовано за умовами Creative Commons BY 4.0 International license

Анотація. Стаття присвячена проблемі використання програмних засобів у процесі вивчення розділу «Парадигми та технології програмування» шкільного курсу інформатики з поглибленим вивченням. Проведено порівняння програми стандарту та профільного рівня для 10-11 класів предмету «Інформатика». Визначено передумови для вивчення даного розділу та визначено результати навчання. Розглянуто питання послідовності вивчення об'єктно-орієнтованого програмування та основ програмування для учнів, продемонстровано переваги та недоліки.

Проаналізовано можливість використання мови Python та середовища програмування PyCharm при вивченні даного розділу в профільних класах. Охарактеризовані мови програмування, що доцільно використовувати для початківців (Python, C++, Java). Описано середовища програмування (Python IDLE, PyCharm, PyDev, Komodo IDE, Spyder). Вказані переваги мови Python для викладання даного розділу.

Ключові слова: методика навчання інформатики, програмування, середовище програмування.

Methodology for Studying the Section "Programming Paradigms and Technologies"

Annotation. The article is devoted to the issue of using software tools in the educational process of studying the section "Paradigms and Programming Technologies" of the school computer science course with in-depth study. The article outlines the relevance of this topic. The comparison of the standard and specialised curriculum for grades 10-11 in "Computer Science" is carried out. The main tasks regarding the aim of the profile level study of this subject are indicated. The specifics of the section "Paradigms and Programming Technologies", which includes such topics as "Databases", "Algorithms", "Web Technologies", are considered.

¹ кандидат педагогічних наук, доцент, доцент кафедри комп'ютерних наук та інформаційних технологій, Житомирський державний університет імені Івана Франка, ORCID: <https://orcid.org/0000-0002-4211-6541>

² асистент кафедри комп'ютерних наук та інформаційних технологій, Житомирський державний університет імені Івана Франка, ORCID: <https://orcid.org/0000-0003-1706-2673>

³ асистент кафедри комп'ютерних наук та інформаційних технологій, Житомирський державний університет імені Івана Франка, ORCID: <https://orcid.org/0000-0001-7563-2692>

⁴ лаборант кафедри комп'ютерних наук та інформаційних технологій, Житомирський державний університет імені Івана Франка, ORCID: <https://orcid.org/0009-0008-7938-2617>

⁵ асистент кафедри комп'ютерних наук та інформаційних технологій, Житомирський державний університет імені Івана Франка, ORCID: <https://orcid.org/0009-0008-1380-2754>

The prerequisites for studying this section (the prerequisites for students to study this section for effective knowledge acquisition) and learning outcomes (what knowledge, skills and values should students have after studying the subject) are determined. The issue of the study sequence in object-oriented programming and programming basics for students is considered. The advantages and disadvantages of both approaches (OOP-first, OOP-later) are demonstrated by examples.

The possibility of using the Python language and the PyCharm programming environment when studying this section in specialised classes is analysed. The programming languages that are appropriate for beginner programmers (Python, C++, Java) are described. Programming environments (Python IDLE, PyCharm, PyDev, Komodo IDE, Spyder) are described. These environments have their own characteristics and advantages to help in choosing the right one for specific learning purposes.

Keywords: computer science teaching methods, programming, programming environment, object-oriented programming, programming language

Вступ

Актуальність. Вже багато років базовий курс інформатики в школі готує спеціалістів та досвідчених в інформаційно-комунікаційних технологіях користувачів. Тенденція обрання професії пов'язаної з ІТ-сферою пов'язана із зростанням даного типу вакансій на ринку праці, адже більшість прогресивних компаній зобов'язані мати корпоративний веб-сайт, ряд компаній використовують бази для зберігання та обробки даних та кожне підприємство намагається автоматизувати прості та складні процеси. Ріст попиту на фахівців у сфері ІТ має вплив на створення навчальних програм для профільного рівня з вивчення інформатики.

Варто зауважити, що навчання здобувачів освіти з поглибленим вивченням інформатики, не обмежує їх майбутню професію до фахівця у сфері інформаційних технологій (ІТ). На даний час більшість з професій вміщує в собі елементи використання інформаційно-комунікаційних технологій, тому є необхідність бути компетентним в загальних засадах функціонування цього середовища.

Сучасна школа переживає період реформ, обумовлених переходом до нової освітньої парадигми, пріоритетами якої є інтереси особистості, що зумовлені тенденціями розвитку суспільства. Здійснювані перетворення визначають появу нових цілей в освіті. Вони полягають в досягненні такого рівня освіченості окремої особистості й суспільства в цілому, який забезпечує вирішення важливих завдань.

Однією із загальних тенденцій світового розвитку є перехід до інформаційного суспільства. У зв'язку з цим Рада Європи до числа найбільш значущих для людини (ключових) компетенцій віднесла компетенції пов'язані з її життям в інформаційному суспільстві. У Національній стратегії розвитку освіти в Україні на період до 2021 року, зазначається, що «Пріоритетом розвитку освіти є впровадження сучасних інформаційно-комунікаційних технологій, що забезпечують удосконалення навчально-виховного процесу, доступність та ефективність освіти, підготовку молодого покоління до життя в інформаційному суспільстві» [1].

Сучасний рівень розвитку суспільства вимагає від школи орієнтації на особистість учня, задоволення його інтересів та освітніх потреб шляхом впровадження диференціації та профільності навчання. Національна доктрина розвитку освіти ХХІ ст. [2] наголошує на створенні та впровадженні системи спеціалізованої підготовки (профільного навчання) в старших класах загальноосвітньої школи, орієнтованої на індивідуалізацію навчання та соціалізацію учнів з урахуванням реальних потреб сучасного ринку праці, відпрацюванні гнучкої системи профілів та кооперації старшої школи із закладами молодшої, середньої та вищої освіти. У зв'язку з прийняттям в

Україні рішення про перехід на дванадцятирічну профільну освіту суттєво зростає суспільна актуальність проблем інтелектуального розвитку учнів, що потребує відповідного психолого-педагогічного забезпечення.

Проблема вивчення об'єктно-орієнтованого програмування (ООП) в умовах сьогодення набуває важливого значення з огляду на необхідність підготовки креативної та всебічно розвиненої особистості. Дану проблему можливо вирішити шляхом забезпечення якості формування компетентностей учнів з використання ООП в закладах середньої освіти. Окреслені вище питання зумовлюють актуальність даного дослідження. Протягом багатьох років дослідженням проблеми вивчення ООП в закладах середньої освіти приділялась значна увага.

Окреслена вище проблема набуває особливої значимості в умовах реформ пов'язаних з розбудовою Нової української школи. У нормативних актах наголошується доцільність підвищення якості шкільної освіти, зокрема у «Державному стандарті базової і повної загальної середньої освіти», «Міжнародній програмі з оцінювання освітніх досягнень учнів» тощо.

Аналіз останніх досліджень і публікацій. Дослідженню різних аспектів багатогранної проблеми використання та створення освітнього веб ресурсу приділялась увага у працях Н. Балик, В. Вембер, О. Гончарової, Л. Гризун, В. Івасика, І. Іваськова, О. Кузьмінської, Н. Морзе, В. Олексюка, С. Ракова, І. Роберт, З. Сейдаметової, С. Семерікова, Ю. Триуса, О. Шиман; формування основ інформаційної культури розглядали А. Єршов, М. Жалдак, О. Кузнецов, Е. Кузнецов, Г. Михалін, В. Монахов, В. Мілітарєв, Ю. Рамський, Н. Розенберг, І. Яглом. Водночас певні питання залишаються дискусійними і потребують більш поглибленого вивчення.

Мета дослідження полягає в розкритті окремих аспектів специфіки вивчення теми «Парадигми та технології програмування» учнями 11 класу профільного рівня загальноосвітньої школи.

Результати

Шкільний курс інформатики в старшій школі побудований з метою продовження формування в учнів інформаційної культури й грамотності. Оскільки інформатика в 10-11 класах є продовженням шкільного курсу, який вивчався раніше, учні, у процесі навчання, мають розширити знання з деяких тем, що вивчались в молодшій та середній школах, а також засвоїти нові знання, що допоможуть у формуванні інформаційної культури та базової компетентності у інформаційно-комунікаційних технологіях.

Розділ «Парадигми та технології програмування» є дуже важливим для правильного розуміння створення та роботи над проектами, він вміщує теоретичну інформацію, що стане у нагоді майбутнім членам колективу не лише ІТ-фірми, а й будь якої галузі праці, адже проектна робота має бути правильно реалізована, перевірена за вимогами тощо.

Проте, оскільки шкільний курс може формуватись по різному (учні починають вчити інформатику в 2, або в 5 класі), у старшій школі з'являється розгалуження у рівнях навчання: у профільному рівні діти поглиблено вивчають деякі теми та вивчають більше тем, у рівні стандарту, що спричинює вивчення тем на базовому рівні.

Рівень стандарту. За навчальним планом, він розрахований на 105 годин, при цьому 35 з них складає модуль, що є обов'язковим для вивчення, а решта годин займає вибірковий блок з переліком тем, які здобувачі освіти можуть обрати за власними вподобаннями.

Профільний рівень. Порівняно з програмою рівня стандарту, у профільних класах за мету ставиться не лише розвинути загальне розуміння ІКТ та сформувати комп'ютерну грамотність. Метою визначено навчити більш розширеним процесам:

- розвинути логічне, аналітичне мислення, а також основні види розумової діяльності, вміння використовувати індукцію, дедукцію, аналіз, синтез, робити висновки та узагальнення;
- розвинути теоретичну базу, що відноситься до процесів перетворення, передавання та використання інформації;
- розвинути вміння розв'язувати задачі різної складності за допомогою ІКТ;
- підготувати учнів до олімпіад, конкурсів, науково-дослідницьких робіт тощо;
- вивчення інформатики на творчому рівні;
- створити зв'язки між інформатикою та іншими предметами, що є у учнів[3].

Вчителям, які навчають здобувачів освіти за програмою профільного рівня рекомендується використовувати різноманітні форми організації навчання для більш ефективного навчання, що дозволить мотивувати учнів та зацікавлювати під час навчального процесу.

Основною відмінністю від стандартного рівня є те, що в профільному вчитель самостійно визначає кількість навчальних годин на вивчення тем, а також їх порядок. Це є великою перевагою, адже завдяки цьому можна орієнтуватись на профіль закладу освіти, враховувати здібності класу учнів, проте зміст робочої програми є незмінним.

Розділ «Парадигми та технології програмування» вивчається у 11 класі профільного рівня, таким чином для даного курсу можна взяти за основу підручник автора В. Руденко «Інформатика. Профільний рівень». Даний підручний випущений у 2019 році та рекомендований для вивчення Міністерством освіти і науки України.

Даний розділ є останнім у порядку вивчення та включає дев'ять тем:

1. Уніфікований процес розробки програмного забезпечення.
2. Програмні інструменти для проектної роботи.
3. Візуальне моделювання архітектури програмного забезпечення.
4. Діаграми UML. Діаграми прецедентів.
5. Моделювання даних і архітектури програмного забезпечення
6. Діаграми послідовностей та діяльностей
7. Проектування інтерфейсу користувача.
8. Тестування та оцінювання програмного забезпечення.
9. Системна архітектура. Апаратні та програмні рішення. [4]

Перші спроби викладання об'єктно-орієнтованого програмування в школі підтвердили, що учні, які до цього вивчали програмування хоча б на базову рівні гірше засвоюють ООП, з огляду на інший процес мислення, ніж ті, які спочатку зрозуміли основи ООП, а потім вже почали розвивати знання з програмування. Це явище назвали «зміною парадигми» та зацікавило широке коло науковців у його дослідження, що відобразились в ряді праць.

Звернемо увагу, що при введенні в шкільну програму такої теми як «Парадигми програмування» в шкільний курс інформатики, потрібно завчасно вводити в курс інформатики вивчення об'єктно-орієнтованого програмування до вивчення основ програмування. Цього твердження притримуються певні вчителі та науковці. Так, вчені, що досліджували це поняття, у ході експериментів підтвердили, що «взаємодія з об'єктами з самого початку допомогла учням побудувати конкретне розуміння та забезпечила відповідні концептуальні моделі. Ті, хто вивчає об'єктно-орієнтоване програмування, можуть легко зосередитися на поняттях зв'язків класів та об'єктів замість того, щоб зосереджуватися на фактах структурного програмування» [5].

Проте існує і зворотне твердження, що послідовність вивчення немає ніякого значення. Так у результаті експериментальних досліджень двох вчених було оголошено: «Головний результат дослідження говорить, що немає різниці між тим, вивчати спочатку об'єктно-орієнтоване програмування, а потім основи програмування (ООП-

first), чи навпаки – спочатку основи програмування, а потім ООП (ООP-later) – це не впливає на результат навчання» [6].

Варто також згадати експериментальні дослідження, що показують відмінності в розумінні програмних текстів об'єктно-орієнтованого стилю. В результаті даного експерименту було виявлено різкий контраст між ментальними репрезентаціями імперативних і об'єктно-орієнтованих програм. У той час як розуміння імперативних програм було в цілому кращим, ніж у об'єктно-орієнтованих програм, уявлення імперативних програм були зосереджені на знаннях програмного рівня. З іншого боку, ментальні уявлення об'єктно-орієнтованих програм більше зосереджені на знаннях рівня домену.[7]

У 11 класі до розділу «Парадигми та технології програмування» вивчаються розділи та теми як:

- Бази даних.
- Алгоритми.
- Веб-технології.

Проте, для вивчення обраної теми учень має повністю володіти набутими у процесі навчання за попередні роки навичками. Безпосередньо у старшій школі вивчається розділ «Мова програмування та структури даних», у якому здобувачі освіти вивчають безпосередньо мову програмування Python, використовують різні середовища програмування та основи об'єктно-орієнтованого програмування з огляду обраної мови програмування. Після проходження даного курсу учень має використовувати можливості середовища програмування для створення та налагодження програм, розв'язувати задачі з використанням усіх базових алгоритмічних структур та їх комбінацій, використовувати змінні різних типів та обґрунтовувати вибір типів даних, розробляти як консольні програми, так і програми з графічним інтерфейсом, знати і програмувати всі базові алгоритми обробки лінійних структур даних, такі як алгоритми вставки, видалення, пошуку елементів, сортування тощо, розробляти алгоритми розв'язування практичних завдань з використанням різних структур даних[8].

У навчальній програмі профільного рівня для 10-11 класів у графі «очікувані результати навчання» сформовано перелік вмій та навичок, що мають отримати учні у процесі вивчення розділу «Парадигми та технології програмування» (рис. 1).

Знаннєва складова
<ul style="list-style-type: none"> •знати основні етапи та методології розробки програмного забезпечення, а також програмні інструменти підтримки цієї діяльності •пояснювати основні принципи побудови моделі задачі (проекту)
Діяльнісна складова
<ul style="list-style-type: none"> •Укладає документацію вимог проекту за результатами дослідження потреб предметної галузі. •Будує модель задачі (проекту) за допомогою візуальних засобів моделювання. •Проектує об'єктно-орієнтовану архітектуру програмних рішень на основі моделей даних та процесів. •Проектує інтерфейс користувача програмного продукту. •Створює об'єктно-орієнтовані програмні рішення. •Описує апаратне забезпечення для реалізації програмного проекту. •Реалізує прототип програмного проекту на основі розробленої архітектури. •Користується системами контролю версій у процесі розробки програмного забезпечення.
Ціннісна складова
<ul style="list-style-type: none"> •Оцінює переваги та недоліки різних методологій розробки програмного забезпечення. •Оцінює реалізацію системи відповідно до вимог проекту.

Рис. 1 Результати навчання розділу "Парадигми та технології програмування"

Мова програмування Python найкраще підходить як перша мова для вивчення програмістами-початківцями, оскільки вона має потужні інструменти, які відображають спосіб мислення людей і спосіб реалізації коду [9]. Крім того, мінімізується додаткові ключові слова, необхідні для написання синтаксично правильної програми. Такий підхід видається більш продуктивним, ніж навчання мовам C або Java, де багато термінів і елементів пов'язані з особливостями мови, а не з реалізацією алгоритму. Крім того, викладачі таких університетів, як MIT, UC Berkeley, UC Davis, Sonoma State University, Washington University, University of Waterloo, Luther College і Swarthmore College, використовували її для викладання вступного курсу програмування студентам відділення інформатики [10]. Сьогодні для фахівця з інформатики важливо вивчити принаймні одну мову програмування, оскільки всі інновації та технології базуються на глибокому розумінні комп'ютерів, операційної системи, програмного API або деяких периферійних пристроїв. Усі вони створені програмістами, які використовують особливий спосіб мислення. Для розвитку такого способу мислення, потрібно опанувати хоча б одну з мов програмування та стати кваліфікованим у розробці програмного забезпечення [11]. Для будь-якої людини, яка починає вивчати програмування, важливо зосередитися на концепціях програмування, а не на специфіці мови, оскільки вона може відрізнитися залежно від мови програмування. Python підтримує найвищий рівень програмування. Таким чином, здобувачу освіти не потрібно думати про керування пам'яттю, яке є неминучим у C, або про ієрархію класів, якої неможливо уникнути в Java, або про типи та оголошення змінних, які існують майже в кожній мові програмування.

Найлегше на Python написати програму «Hello World». Більшість мов програмування вимагають написання багатьох специфічних методів або функцій, оголошень класів або програм тощо. Python дає можливість почати програмування не використовуючи дані вимог. Python є інтерпретованою мовою, тому за допомогою інтерпретатора командного рядка здобувач освіти може з легкістю перевірити, як працюють оператори чи функції [12]. Інтерпретатор Python має вбудований модуль довідки, який може значно покращити процес розуміння різних аспектів мови. Розуміння мови програмування, здобувачам освіти (учням, які не мають досвіду програмування) потрібно навчитися мислити як комп'ютер, що вимагає великих зусиль та повної зміни парадигми мислення. Реалізація коду Python досить проста, тому людина, яка пройшла курс елементарної математики, знайде такі інструменти, як «змінні» та «функції», легкими у використанні. Переваги Python значні, тому використання його як основної мови для вивчення програмування може значно вплинути на швидкість вивчення інформатики в цілому [13].

Проте більшість вчителів та викладачів, надають перевагу вивченню C++. Мова програмування C надає модель пам'яті та обчислень, яка майже відповідає моделі більшості комп'ютерів. Крім того, вона забезпечує потужні та гнучкі механізми для абстракції; тобто мовні конструкції, які дозволяють програмісту вводити та використовувати нові типи об'єктів, які відповідають концепціям програми. Таким чином, C підтримує стилі програмування, які покладаються на досить пряме маніпулювання апаратними ресурсами для забезпечення високого рівня ефективності, а також стилі програмування більш високого рівня, які покладаються на типи, визначені користувачем, для забезпечення моделі даних і обчислень, ближчої за розумінням людини завдання, яке виконує комп'ютер. Дані високо рівневі стилі програмування часто називають абстракцією даних, об'єктно-орієнтованим програмуванням і загальним програмуванням.

Аналізуючи джерела виділимо статтю «C++ or Python? Which One to Begin With: A Learners Perspective», де шляхом експерименту працівники університету в The Islamia University of Bahawalpur Pakistan (Пакистан) довели, що вибір мови програмування

вивчення у вступному курсі програмування вимагає ретельності. Висновки їх праці надають розуміння, що вибір мови може вплинути на навчання, володіння вмінням та прогрес здобувачів освіти у сфері комп'ютерної техніки, де головну роль відіграє програмування. Здобувачі освіти насамперед очікують від першої мови програмування, це легкість у розумінні та використанні. Вагомість С, Java та інших мов варто враховувати, Python в свою чергу буде доцільним вибором, для вступного курсу програмування. Здобувачі освіти проявляють більше прогресу під час вивчення Python та висловлюють думку, що більшість функцій мови, включаючи цикли та умови, були легшими порівняно з С.

Виокремимо переваги мови Python серед інших мов програмування:

- Python пропонує чистий та більш інтуїтивно зрозумілий синтаксис порівняно з іншими сучасними конкурентними мовами для цієї мети, такими як С та Java.
- Python має динамічну типізацію, що зменшує час створення коду та полегшує новачкові зосередитися на розв'язанні проблем, а не на специфіці мови.
- Python пропонує кращу виразність завдяки представленню потужних типів даних, таких як списки та словники, які можна легко вивчати одночасно з іншими базовими типами даних. Python також має дуже високий відсоток операторів і подібних до мови програмування С.
- Інтерпретована природа Python дає миттєвий зворотний зв'язок. Програмування не тільки стає швидким, але й допомагає справлятися з помилками краще та простіше.
- Без використання зайвих дужок і крапок з комою Python забезпечує структурний дизайн, дотримуючись відступів і структурованого способу написання програм. Особливості синтаксису надають Python більше схожості з псевдокодом.
- Python пропонує розширюваність мовних функцій завдяки дворівневій мовній конструкції.
- Середовище програмування за замовчуванням (IDLE) є відкритим і безкоштовним. Воно також може працювати на більшості платформ.

Вибір середовища програмування, яке буду використовувати здобувач освіти так само важливий, як і вибір самої мови програмування, адже у доступних середовищах є відмінності у описі функцій, зручності інтерфейс, різні аспекти, що впливають на навчання.

При виборі мови програмування враховують мету, зручність та доцільність використання та залежно від мови програмування потрібно обрати середовище, у якому буде зручно працювати не лише на початкових етапах ознайомлення з мовою, а і повноцінно впродовж проєкту.

У статті Базуріна В.М. «Порівняльний аналіз середовищ програмування мовою Python» йдеться про те, що застосування різних середовищ програмування може бути і перевагою, адже користувач, що працював з декількома середовищами може перейти з одного середовища в інше, при цьому витративши менше часу та зусиль для вивчення нового інтерфейсу, функціоналу тощо. Проте, як зазначає автор, «Якщо ж розглядати навчання програмування з точки зору учня-початківця, то наявність кількох середовищ програмування – швидше недолік, адже в одних підручниках описується одне середовище програмування, а в інших – інше. Виникає проблема оптимального вибору середовища програмування, робота у якому була б якомога комфортною для учня» [14].

Тож, серед існуючих середовищ програмування, можна виокремити найбільш популярні.

Для мови програмування Python найпопулярнішими є:

Python IDLE — це інтегроване середовище розробки та навчання Python. Середовище має такі особливості:

- закодований повністю на Python, використовуючи інструментарій GUI tkinter;
- кросплатформенність: працює майже однаково на Windows, Unix і macOS;
- вікно оболонки Python (інтерактивний інтерпретатор) із визначенням кольору команд введення, виведення коду та повідомлень про помилки;
- багатовіконний текстовий редактор із множинним розпізнаванням, розфарбовуванням Python, розумним відступом, підказками щодо виклику, автозавершенням та іншими функціями;
- пошук у будь-якому вікні, заміна у вікнах редактора та пошук у кількох файлах (grep);
- налагоджувач із постійними точками зупинки, кроками та переглядом глобальних і локальних просторів імен [15].

PyCharm – це середовище програмування від компанії JetBrains. Цінується дане середовище як новачками (за простий та інтуїтивний інтерфейс) так і професійними програмістами зі стажем (за зручність та розширений функціонал)[16].

Функції PyCharm:

- вбудовані інструменти розробника (PyCharm включає інтегрований налагоджувач і засіб виконання тестів; профайлер Python; вбудований термінал; інтеграція з основними VCS і вбудованими засобами баз даних; можливості віддаленої розробки з віддаленими перекладачами; вбудований термінал ssh; і інтеграція з Docker і Vagrant)
- веб-розробка (окрім Python, PyCharm надає першокласну підтримку для різноманітних фреймворків веб-розробки Python, окремих мов шаблонів, JavaScript, CoffeeScript, TypeScript, HTML/CSS, AngularJS, Node.js тощо.)
- науковий інструментарій (PyCharm інтегрується з IPython Notebook, має інтерактивну консоль Python і підтримує Anaconda, а також численні наукові пакети, включаючи Matplotlib і NumPy.)
- кросплатформенність (PyCharm можна використовувати у Windows, macOS і Linux за допомогою одного ліцензійного ключа.)
- інтелектуальна допомога в кодуванні (PyCharm забезпечує інтелектуальне завершення коду, перевірку коду, підсвічування помилок на льоту та швидкі виправлення, а також автоматизовану рефакторинг коду та широкі можливості навігації.)

PyDev — це IDE Python для Eclipse, який можна використовувати в розробці на Python, Jython та IronPython [17].

Воно поставляється з багатьма привабливими функціями, такими як (рис. 2).

<ul style="list-style-type: none"> • Інтеграція Django • Доповнення коду • Доповнення коду з автоматичним імпортом • Підказка типу • Перейти до визначення 	<ul style="list-style-type: none"> • Рефакторинг • Аналіз коду • Налгоджувач • Віддалений налагоджувач • Браузер токенів 	<ul style="list-style-type: none"> • Unittest інтеграція • Покриття коду • Інтеграція PyLint • Інтеграція MyPy • Пошук посилань (Ctrl+Shift+G) • Інтерактивна консоль
---	---	---

Рис.2. Функції середовища програмування PyDev

Рекомендований спосіб використання PyDev міститься в Liclipse, який забезпечує вбудований PyDev, а також підтримку інших мов, таких як Django Templates, Mako, RST, C++, CoffeeScript, Dart, HTML, JavaScript, CSS, серед інших (також за ліцензією Liclipse ви безпосередньо підтримуєте розвиток PyDev).

Komodo IDE – універсальне IDE з кросплатформенністю та відкритим кодом. Дане середовище підтримує віддалену роботу та корпоративне використання [18].

Spyder — це безкоштовне наукове середовище з відкритим кодом, написане мовою Python для Python і розроблене вченими, інженерами та аналітиками даних. Воно характеризується унікальним поєднанням розширених функцій редагування, аналізу, налагодження та профілювання комплексного інструменту розробки з дослідженням даних, інтерактивним виконанням, глибокою перевіркою та чудовими можливостями візуалізації наукового пакету [19].

Висновки

У процесі дослідження методики вивчення розділу «Парадигми та технології програмування» визначено, що маючи знання про етапи розробки та послідовну організацію роботи над розробкою програмного забезпечення, здобувачу освіти повністю забезпечується адекватне та правильне уявлення про роботу над проектами, їх реалізацією тощо.

Використання мови Python для вивчення розділу "Парадигми та технології програмування" в шкільному курсі інформатики має кілька переваг та особливостей.

Легкість вивчення: Python є однією з найбільш доступних мов програмування для початківців. Його зрозумілість та лаконічність допомагають учням швидше засвоювати основні концепції програмування.

Читабельність: синтаксис Python нагадує англійську мову, що робить код більш зрозумілим і читабельним. Це важливо при вивченні концепцій та демонстрації прикладів.

Широкі можливості: Python підтримує різні парадигми програмування, такі як процедурне, об'єктно-орієнтоване та функціональне програмування. Це дозволяє вчителям представити різні підходи до програмування на прикладах.

Велика спільнота та ресурси: Python має велику та активну спільноту розробників, що створює велику кількість навчальних матеріалів, уроків, завдань та проектів, які можна використовувати для навчання.

Обробка даних та наукові обчислення: Python популярний у наукових галузях, адже має багато бібліотек для обробки даних, наукових обчислень та візуалізації результатів. Це може стимулювати інтерес учнів до глибшого вивчення.

Підготовка до майбутньої кар'єри: Python є однією з найпопулярніших мов у сфері програмування та розробки програмного забезпечення, тому вивчення цієї мови може підготувати учнів до подальших можливостей у сфері ІТ.

Проектна діяльність: Python дозволяє легко створювати невеликі проекти та програми. Це дає учням можливість застосовувати свої знання на практиці та реалізовувати власні ідеї.

Інтерактивне навчання: Python часто використовується у віртуальних середовищах, де учні можуть безпосередньо взаємодіяти з кодом та спостерігати результати в реальному часі.

Подальше дослідження з метою максимально ефективного використання Python при вивченні парадигм та технологій програмування, окреслює мову в розробці програми навчання, яка включатиме теоретичні пояснення, практичні завдання, демонстрацію проектів та можливість творчої роботи учнів.

Список використаних джерел

1. Дорошенко Ю. О. Навчання інформатики у структурі 12-річної загальної середньої освіти / Ю. О. Дорошенко, Н. С. Прокопенко // Інформатика та інформаційні технології в навчальних закладах. – 2006. – № 1. – С. 55–72.

2. Морзе Н. В. Методика навчання інформатики : навч. посіб. : [у 3 ч.] / Н. В. Морзе ; за ред. акад. М. І. Жалдака. – К. : Навчальна книга, 2004. – Ч. 1: Загальна методика навчання інформатики. – 2004. – 256 с.
3. Навчальна програма профільного рівня для 10-11 класів з інформатики, веб-сайт. URL: <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-dlya-10-11-klasiv> (дата звернення: 01.11.2022).
4. Руденко В. Д. Інформатика (профільний рівень) : підруч. для 11 кл. закл. загал. серед. освіти /В. Д. Руденко, Н. В. Речич, В. О. Потієнко. — Харків : Вид-во «Ранок», 2019. — 256 с. : іл.
5. The Effects of Objects-First and Objects-Late Methods on Achievements of OOP Learners. URL: <https://www.scirp.org/html/23962.html?pagespeed=noscript> (дата звернення: 01.10.2022).
6. Empirical comparison of objects-first and objects-later. URL: https://dl.acm.org/doi/pdf/10.1145/1584322.1584326?casa_token=xL2ji8IMSTUAAAAA:zivCJixx7uP2Q3K3ECycHzMuGBZpjNicBpHBhtvMSrTB4YsK0H-0B607TzZbMp1_SyriZWDwYp-D (дата звернення: 01.10.2022).
7. An Empirical Study of Novice Program Comprehension in the Imperative and Object-Oriented Styles. URL: <https://dl.acm.org/doi/pdf/10.1145/266399.266411> (дата звернення: 01.10.2022).
8. Навчальна програма профільного рівня для 10-11 класів з інформатики, веб-сайт. URL: <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-dlya-10-11-klasiv> (дата звернення: 01.11.2022).
9. Downey A, Elkner J and Meyers C 2008 "How to think like Computer Scientist` Learning with Python"
10. Python Tutor: Learn Python, JavaScript, C, C++, and Java programming by visualizing code: веб-сайт. URL: <https://pythontutor.com/> (дата звернення 05.05.2023)
11. Sanders ID and Langford S "Students' Perceptions of Python as a First Programming Language at Wits"
12. Downey A, Elkner J and Meyers C 2008 "How to think like Computer Scientist` Learning with Python"
13. Journal of Physics: Conference Series, Volume 423, 2013 International Conference on Science & Engineering in Mathematics, Chemistry and Physics (ScieTech 2013) 24–25 January 2013, Jakarta, Indonesia
14. Базурін, В., Омелечко, Є., & Ковтун, А. (2018). Comparative analysis of development environments on Python language. *New Computer Technology*, 16, 281-292.
15. IDLE – Python 3.11.3 documentation: веб-сайт. URL: <https://docs.python.org/3/library/idle.html> (дата звернення 14.05.2023)
16. Features – PyCharm: веб-сайт. URL: <https://www.jetbrains.com/pycharm/features/> (дата звернення 14.05.2023)
17. PyDev: веб-сайт. URL: <https://www.pydev.org/> (дата звернення 14.05.2023)
18. Komodo IDE By ActiveState - One IDE for All Your Languages: веб-сайт. URL: <https://www.activestate.com/products/komodo-ide/> (дата звернення 05.05.2023)
19. Spyder IDE: веб сайт. URL: <https://www.spyder-ide.org/> (дата звернення 05.05.2023)