Комунальний заклад «Ліцей №11»
Кам’янської міської ради

ФОРМУВАННЯ ІНФОРМАЦІЙНО-КОМУНІКАТИВНОЇ КОМПЕТЕНТНОСТІ В УЧНІВ 5 КЛАСІВ ПІД ЧАС УРОКІВ ІНФОРМАТИКИ

Навчально-методичний посібник

Кам’янське
2026

Укладач, автор: Олійник Н.В., учитель інформатики

Навчально-методичний посібник

«Формування інформаційно-комунікативної компетентності в учнів середньої та старшої школи під час уроків інформатики»

Посібник містить:

- теоретичні положення щодо інформаційно-комунікативної компетентності учнів;
- практичні методи та прийоми її формування під час уроків інформатики;
- кейси інтеграції сучасних медіа- та STEM-технологій;
- питання для контролю навичок учнів та самоконтролю вчителя;
- методичні рекомендації щодо впровадження сучасних технологій навчання.

Призначений для:

- педагогів загальноосвітніх та позашкільних навчальних закладів;
- студентів педагогічних закладів;
- усіх, хто цікавиться сучасними методами навчання інформатики та формування ІК-компетентностей.

Зміст

Анотація

до навчально-методичного посібника Н.В.Олійник

«Формування інформаційно-комунікативної компетентності учнів під час уроків інформатики»

У навчально-методичному посібнику «Формування інформаційно-комунікативної компетентності учнів під час уроків інформатики» здійснено теоретичний аналіз поняття інформаційно-комунікативної компетентності в контексті сучасної інформатичної освіти. Окреслено її структуру, основні складники та визначено етапи формування в учнів базової та середньої школи.

У посібнику розкрито роль інформаційно-комунікаційних технологій як провідного засобу розвитку алгоритмічного мислення, цифрової грамотності, навичок програмування та свідомого використання цифрових ресурсів. Визначено ключові інформаційно-комунікативні вміння, якими має володіти учень у процесі вивчення інформатики, зокрема вміння працювати з інформацією, створювати алгоритми, реалізовувати їх у вигляді програм, аналізувати результати та презентувати власні напрацювання.

Формування інформаційно-комунікативної компетентності пропонується здійснювати шляхом інтеграції практико-орієнтованого навчання, елементів медіаосвіти та STEM-підходу під час уроків інформатики. У посібнику обґрунтовано доцільність використання алгоритмічних задач, практичних вправ з програмування (зокрема мовою Python), проєктної та дослідницької діяльності як ефективних засобів формування ключових компетентностей учнів.

Посібник містить структурований теоретичний матеріал, систему практичних завдань різного рівня складності, приклади навчальних вправ і проєктних робіт, а також методичні рекомендації щодо організації уроків інформатики з компетентнісним і діяльнісним підходами.

Навчально-методичний посібник є актуальним для вчителів інформатики закладів загальної середньої та позашкільної освіти, студентів педагогічних закладів, а також усіх, хто цікавиться сучасними методиками навчання інформатики та формування інформаційно-комунікативної компетентності учнів.

Вступ

Актуальність створення навчально-методичного посібника зумовлена вимогами Державного стандарту базової середньої освіти та компетентнісного підходу Нової української школи, які передбачають формування в учнів інформаційно-комунікативної компетентності як однієї з ключових. У сучасному цифровому середовищі інформатика виступає не лише навчальним предметом, а й інструментом розвитку алгоритмічного мислення, цифрової грамотності, вміння працювати з інформацією та створювати власні цифрові продукти.

Особливої уваги потребує початковий етап вивчення алгоритмізації та програмування, зокрема у 5 класі, коли в учнів закладаються базові уявлення про алгоритм, програму, мову програмування та логіку побудови розв'язків. Саме на цьому етапі доцільно поєднувати теоретичні знання з практичною діяльністю, використовуючи доступні мови програмування, зокрема Python, що відповідає віковим особливостям учнів і дозволяє швидко отримувати наочний результат.

Метою навчально-методичного посібника є надання вчителям інформатики систематизованого теоретичного та практичного матеріалу для формування інформаційно-комунікативної компетентності учнів під час вивчення теми «Алгоритми та програми», а також підвищення ефективності уроків інформатики через використання практико-орієнтованих, інтерактивних і STEM-елементів навчання.

Основними завданнями посібника є:

- розкриття теоретичних засад формування інформаційно-комунікативної компетентності учнів на уроках інформатики;
- визначення структури та складників інформаційно-комунікативної компетентності в контексті вивчення алгоритмів і програмування;
- добір і систематизація практичних завдань з програмування мовою Python для учнів 5 класу;
- надання методичних рекомендацій щодо організації навчальної діяльності з урахуванням компетентнісного та діяльнісного підходів;

- сприяння розвитку в учнів умінь аналізувати, створювати та презентувати власні алгоритмічні й програмні рішення.

Посібник орієнтований на вчителів інформатики закладів загальної середньої та позашкільної освіти, студентів педагогічних закладів освіти та може бути використаний у процесі підготовки до атестації педагогічних працівників, а також для вдосконалення професійної майстерності.

Розділ 1. Формування інформаційно-комунікативної компетентності як інтегративна складова сучасного уроку інформатики та STEM-освіти

У сучасному цифровому середовищі інформаційно-комунікативна компетентність (ІКК) є ключовою складовою розвитку учня. Ця компетентність визначає здатність ефективно використовувати інформаційно-комунікаційні технології (ІКТ) для навчання, дослідження, комунікації та розв'язання прикладних задач [Морзе Н. В., Барна О. В., Вембер В. П.]. Особливо важливо формувати ІКК у школярів середньої та старшої школи в умовах інтеграції STEM-освіти, що передбачає використання міждисциплінарних завдань із природничих наук, технологій, інженерії та математики з акцентом на цифрові технології та алгоритмічне мислення.

1.1. Теоретичні основи інформаційно-комунікативної компетентності

ІКК охоплює такі основні складники:

Інформаційна складова – уміння ефективно шукати, обробляти та аналізувати дані, використовувати цифрові ресурси для навчання та саморозвитку;

Комп'ютерно-технологічна складова – навички роботи з апаратним та програмним забезпеченням, створення простих алгоритмів та програм, використання інтерактивних цифрових інструментів;

Процесуально-діяльнісна складова – здатність застосовувати сучасні ІКТ для вирішення практичних завдань, проєктування моделей, алгоритмів та STEM-проектів [Горобченко Н., Ващенко Т – 2014; Бондаренко Т. – 2016].

Етапи формування інформаційно-комунікативної компетентності (за Т.Бондаренко):

Ознайомлення з базовими цифровими інструментами та їх призначенням.

Навчання пошуку інформації в мережі Інтернет та оцінка її достовірності.

Розробка власних проєктів і алгоритмів із використанням ІКТ.

Рефлексія результатів та обґрунтування вибору оптимальних рішень.

1.2. Інтеграція STEM-елементів у формування ІКК

STEM-освіта є невід'ємною частиною сучасної методики навчання інформатики. Вона дозволяє:

- формувати алгоритмічне мислення та навички програмування;
- інтегрувати знання з математики, фізики, технологій та інженерії у практичні проєкти;
- розвивати критичне мислення та здатність приймати обґрунтовані рішення;
- навчати учнів створювати цифрові продукти, моделювати процеси та використовувати сучасні технології [Калиніна Л., Калінін В., 2018; Лойко Н.П., 2019].

Застосування STEM-підходу підвищує практичну цінність ІКК, оскільки учні вчаться не лише працювати з інформацією, а й інтегрувати її в міждисциплінарні задачі, реалізовувати проєкти та співпрацювати в команді.

1.3. Роль інформаційно-комунікаційних технологій у сучасному уроку інформатики

ІКТ є ключовим інструментом у формуванні ІКК. Їх використання дозволяє:

- індивідуалізувати навчальний процес;
- створювати додаткову мотивацію до навчання;
- забезпечувати доступ до актуальних навчальних ресурсів та цифрових енциклопедій;
- здійснювати комунікацію та спільне вирішення завдань [Мельник Ю., 2017].

Особливу увагу слід приділяти інтерактивним та дистанційним технологіям, що дозволяють учням самостійно опановувати матеріал і розвивати навички цифрової грамотності [Алієв Х., 2018].

1.4. Інформаційно-комунікативні уміння та навички учня

Під час уроків інформатики з інтеграцією STEM-елементів учень повинен навчитися:

- шукати і систематизувати інформацію;
- створювати алгоритми і програмні рішення;
- працювати самостійно або в команді з використанням цифрових ресурсів;
- оцінювати достовірність інформації;
- застосовувати цифрові інструменти для моделювання, презентацій та вирішення практичних задач;
- захищати авторські права та дотримуватись етики роботи з інформацією [Калінін В., Калініна Л., 2018; Горобченко Н., Ващенко Т., 2014].

Розділ 2. Практичні методи і прийоми формування інформаційно-комунікативної компетентності учнів середньої та старшої школи засобами інформатики та STEM-підходу

Хоча сьогодні в педагогічній практиці достатньо прикладів використання STEM-підходів та цифрових технологій на уроках інформатики, більшість із них зосереджена на окремих завданнях або окремих компетенціях учнів. Багато наукових праць описують ІКТ-вправи або методичні рекомендації, проте бракує комплексних, системних підходів, які одночасно поєднували б формування інформаційно-комунікативної компетентності, STEM-проєктну діяльність та розвиток цифрових навичок [Лойко Н.П., 2014].

Саме це зумовило необхідність розробити власну авторську методику, яка відрізняється від існуючих: вона інтегрує практичну роботу з інформацією, медіаосвітні компоненти та STEM-проєкти, а також орієнтована на послідовний розвиток критичного мислення, командної роботи та вмінь представляти власні результати. Такий підхід дозволяє учням не лише здобувати знання з інформатики, а й розвивати навички ефективного комунікування, аналізу та використання інформації у різних форматах.

Моя педагогічна ідея ґрунтується на поетапному формуванні компетентностей: від роботи з інформацією та даними, через створення цифрових продуктів, до презентації та захисту проєктів. Це забезпечує системне поєднання ІКТ, STEM і медіаосвіти, чого бракує в існуючих методичних розробках.

Окреслення реалізації педагогічної ідеї доцільно розпочати з короткого огляду сучасних освітніх тенденцій у STEM та медіаосвіті, а далі представити конкретні методи, прийоми та авторські розробки уроків, спрямовані на формування інформаційно-комунікативних навичок у учнів середньої та старшої школи.

2.1. Практичний кейс із формування інформаційно-комунікативної компетентності на уроках інформатики з інтеграцією STEM та медіаосвіти

STEM-навчання визначається як інтеграція науки, технологій, інженерії та математики, яка спрямована на розвиток практичних навичок, критичного мислення та креативності. У поєднанні з медіаосвітніми компонентами STEM-проєкти дозволяють формувати такі компетентності учнів [Горобченко Н., Ващенко Т., 2014]:

- ефективний пошук та обробка інформації з надійних цифрових джерел;
- аналіз та оцінка достовірності даних;
- створення цифрових продуктів (міні-проєкти, презентації, програми, анімації);
- робота з графіками, таблицями та даними для прийняття обґрунтованих рішень;
- командна робота та презентація результатів;
- дотримання етики роботи з інформацією та авторських прав.

Інтеграція медіаосвітніх компонентів у STEM-проєкти дозволяє учням не лише збирати і обробляти інформацію, а й створювати власні продукти, аналізувати їх та комунікувати результати, що повністю відповідає завданням формування інформаційно-комунікативної компетентності та стандартам НУШ.

Методичною основою моєї практичної діяльності є авторський проєктний метод із поетапним виконанням завдань, який включає: постановку проблеми, збір та обробку даних, створення цифрового продукту, тестування та презентацію результату. Такий підхід дозволяє поєднувати навчальні цілі з інтересами учнів, створюючи інноваційний формат уроку, відмінний від існуючих методик [Василенко О. М.].

До ефективних форм проведення уроків належать:

- STEM-проєкти та міні-проєкти з цифровою реалізацією;
- лабораторні роботи з аналізом даних та алгоритмів;

- цифрові квести та інтерактивні вправи;
- створення презентацій та мультимедійних продуктів;
- командна розробка та захист результатів.

Особливо важливо фіксувати цілі STEM та медіаосвітнього компонентів у поурочному плані, паралельно з навчальними цілями з інформатики. Це дозволяє чіткіше оцінювати результати уроку та формує комплексні інформаційно-комунікативні компетентності учнів.

Таким чином, хоча існують численні приклади STEM-уроків та ІКТ-активностей, бракує авторських системних розробок, які комплексно формують інформаційно-комунікативні навички. Саме це визначає напрям моєї педагогічної діяльності.

Далі представлені розгорнуті плани-конспекти уроків інформатики з інтеграцією STEM та медіаосвіти, спрямовані на розвиток інформаційно-комунікативної компетентності учнів середньої школи, а саме 5 класів.

План-конспект № 1

Клас: **5**

Тема: **«Алгоритми. Команди і виконавці»**

Мета: ***Формування компетентностей***

Предметна компетентність:

- **Навчальна:** удосконалити знання поняття алгоритму; формувати знання про виконавців алгоритму та їхні системи команд; навчити наводити приклади виконавців та команд, які вони виконують.
- **Розвивальна:** розвивати логічне та алгоритмічне мислення, інформаційну культуру.
- **Виховна:** формувати уміння працювати в групі, відповідальність за результат, позитивне ставлення до навчання.

Ключові компетентності:

- **Спілкування державною мовою:** висловлюватись та спілкуватися на тему ІТ з використанням відповідної термінології.
- **Основні компетентності у природничих науках і технологіях:** застосовувати логічне та системне мислення для розв’язування проблем.
- **Уміння вчитися впродовж життя:** організовувати власну діяльність із використанням програмних засобів для структурування роботи.
- **Соціальна та громадянська компетентності:** дотримуватись правил БЖД при роботі з комп’ютером.

Тип уроку: засвоєння нових знань

Обладнання та наочність: дошка, комп’ютери, підручник з інформатики, презентація, роздаткові картки з прикладами алгоритмів.

Хід уроку

І. Організаційний етап
Привітання, перевірка готовності до уроку.

ІІ. Перевірка домашнього завдання
Фронтальна бесіда: що учні пам’ятають про алгоритми, виконавців та команди у Scratch.

ІІІ. Актуалізація опорних знань
Фронтальне опитування:

1. Що таке алгоритм?
2. Хто такий виконавець алгоритму?
3. Наведіть приклади виконавців, яких ви знаєте.
4. Що таке команда виконавця?
5. Наведіть приклад команди, яку може виконувати людина, робот або комп’ютер.
6. Чому важливо правильно складати алгоритм?
7. Які правила БЖД треба пам’ятати, працюючи за комп’ютером?

IV. Мотивація навчальної діяльності

Щодня ми виконуємо правила та інструкції. Наприклад, готуємо страву за рецептом, граємо у спортивну гру, пересуваємось містом за правилами дорожнього руху. Уявіть, що алгоритми – це інструкції, які допомагають комп’ютеру чи роботам правильно виконувати завдання.

V. Повідомлення теми, цілей, завдань уроку

VI. Сприймання й усвідомлення нового матеріалу
Пояснення вчителя з демонстрацією презентації та прикладів:

- *Алгоритм* – послідовність команд для досягнення мети.
- *Виконавці*: люди, роботи, комп’ютерні програми.
- *Система команд виконавця*: базові дії, які він може виконувати.

VI. Фізкультхвилинка

VII. Засвоєння нових знань, формування вмінь

Робота за комп’ютерами (унікальна STEM-практика для 5 класу):

1. **Правила БЖД**: перевірка робочого місця, положення рук, дистанція до монітора.
2. **Інструктаж учителя**: як працювати з алгоритмічними блоками у Scratch/Python (мінімальна складність).

💻 Практична робота 1 – «Алгоритмічний лабіринт» (STEM-компонент):

- Учні отримують завдання: допомогти «роботу-коту» пройти лабіринт, використовуючи команди: вперед, повернути ліворуч, повернути праворуч.
- Додатково: ускладнення для груп – додати «перешкоди» і створити альтернативні алгоритми.

Мета: навчитися складати послідовність команд для досягнення мети, розвивати логічне та алгоритмічне мислення.

Інструкції для учнів:

1. Розділити дітей на групи по 3-4 учні.
2. Кожна група отримує **схему лабіринту** та «робота-кота» (можна позначити фігуркою або на екрані).
3. Використовуйте команди:
    - **Вперед** – котик робить один крок вперед;
    - **Повернути ліворуч** – котик повертається на 90° вліво;
    - **Повернути праворуч** – котик повертається на 90° вправо.
4. Складіть алгоритм, щоб котик дійшов до виходу з лабіринту.
5. Перевірте алгоритм: виконайте команди по черзі та простежте, чи котик дійшов до мети.
6. **Додаткове ускладнення (для груп, які завершили завдання швидко):**
    - Додайте на шляху котика перешкоди (наприклад, стіни, ями).
    - Складіть альтернативний алгоритм, щоб обійти перешкоди.

Обговоріть у групі: який алгоритм був найефективніший, чи можна його покращити, щоб котик дістався мети швидше та безпечно.

💻 Практична робота 2 – «Алгоритм у житті» (інформаційно-комунікативна компетентність):

- Скласти алгоритм для простого щоденного завдання (наприклад, «приготувати сендвіч»).
- Обмін алгоритмами у групах, обговорення, що можна покращити для точності та безпеки.

Мета: навчитися застосовувати поняття алгоритму в повсякденному житті та формувати вміння точно передавати інструкції.

Інструкції для учнів:

1. Розділіться на групи по 3–4 учні.
2. Виберіть просте щоденне завдання, наприклад:
    - приготувати сендвіч;
    - почистити зуби;
    - одягнутися вранці;
    - підготувати портфель до школи.
3. Складіть **покроковий алгоритм** для виконання завдання:

- Кожен крок має бути чітким і зрозумілим для іншого.
- Використовуйте слова-команди: «візьми», «поклади», «помий», «налий», «закрий» тощо.

4. Обміняйтеся алгоритмами з іншою групою.
5. Перевірте: чи можна виконати завдання, слідуючи алгоритму іншої групи?
6. Обговоріть:
   - чи всі кроки зрозумілі;
   - що можна змінити, щоб алгоритм був точнішим;
   - як забезпечити безпеку під час виконання завдання.

📣 Рішення/підказки:

- Лабіринт: послідовність команд залежить від розмітки на екрані; кожна команда повинна бути визначена та скінчена.
- Щоденний алгоритм: виділити чіткі дії, порядок, уникнути небезпечних моментів (наприклад, газ).

VIII. Узагальнення та систематизація знань

- Опитування: що таке алгоритм, хто виконавець, приклади команд.
- Коротка гра «Помилка у алгоритмі» – учні знаходять помилку у запропонованому алгоритмі і виправляють її.

IX. Домашнє завдання

- Підручник § 18.
- Скласти алгоритм для улюбленої гри або заняття, намалювати блок-схему.

X. Підсумки уроку

- Рефлексія: «Для мене сьогодні важливим було…»
- «Сьогодні я дізнався/дізналася про…»
- «Мені хотілося б навчитись…»

План-конспект № 2

Клас: **5**

Тема: **«Способи подання алгоритмів. Алгоритми і програми. Основні поняття мови програмування Python»**

Мета: ***Формування компетентностей***

Предметна компетентність:

- **Навчальна:** ознайомити учнів із способами подання алгоритмів (словесно, графічно, у вигляді блок-схем); формувати поняття програми та основні елементи мови Python; навчити складати прості алгоритми на Python.
- **Розвивальна:** розвивати логічне мислення, увагу, уміння структурувати інформацію.
- **Виховна:** виховувати культуру безпечної роботи за комп'ютером, вміння працювати в групі.

Ключові компетентності:

- **Спілкування державною мовою:** формувати вміння пояснювати алгоритми та описувати дії програмою Python.
- **Основні компетентності у природничих науках і технологіях:** формувати вміння застосовувати алгоритмічне та структурне мислення.
- **Уміння вчитися впродовж життя:** навчити організовувати власну діяльність із використанням програмного забезпечення.
- **Соціальна та громадянська компетентність:** формувати дотримання правил безпеки життєдіяльності під час роботи з ІТ-пристроями.

Тип уроку: засвоєння нових знань з елементами практичної роботи.

Обладнання та наочність: дошка, комп'ютери, підручники «Інформатика, 5 клас» автор Бондаренко, презентація, проектор.

Хід уроку

І. Організаційний етап

- Привітання, перевірка готовності до уроку.

ІІ. Перевірка домашнього завдання

ІІІ. Фронтальне опитування (актуалізація опорних знань)

1. Що таке алгоритм?

2. Назвіть приклади алгоритмів у повсякденному житті.
3. Які команди ми використовували на минулому уроці?
4. Що таке виконавець алгоритму?
5. Які є способи подання алгоритмів (словесно, графічно, у вигляді таблиці)?

IV. Мотивація навчальної діяльності

У житті ми постійно користуємося алгоритмами: готуємо страви, граємо в комп'ютерні ігри, керуємо роботами. Щоб комп'ютер виконав вашу команду, потрібно скласти зрозумілий алгоритм та перевести його у програму. Сьогодні ми навчимося писати прості програми на Python і показувати свої алгоритми у різних формах.

V. Повідомлення теми, цілей та завдань уроку

- Тема: «*Різні способи подання алгоритмів. Алгоритми і програми. Основні поняття мови Python*».
- Мета уроку: навчитися подавати алгоритми словесно та графічно, формувати програми на Python, зрозуміти базові поняття мови програмування.

VI. Сприймання й усвідомлення учнями нового матеріалу

Пояснення вчителя з елементами презентації та демонстрації на екрані:

1. **Способи подання алгоритмів:**
   - словесно;
   - графічно (блок-схеми);
   - таблично (крок-команда-результат).
2. **Алгоритм & програма:**
   - ***Алгоритм*** *– послідовність дій для досягнення мети;*
   - ***Програма*** *– алгоритм, записаний у мові програмування, який може виконувати комп'ютер*.
3. **Основні поняття Python:**
   - Команда/функція;
   - Змінна;
   - Вивід даних (print);
   - Введення даних (input);
   - Коментарі (#).

 # Програма для привітання користувача

```
name = input("Введіть своє ім'я: ")

print("Привіт,", name)
```

VII. Фізкультхвилинка

- Легка розминка для очей та рук (10-15 секунд вправи на відведення погляду від екрану, підняття рук, обертання плечима).

VIII. Засвоєння нових знань, формування вмінь (практична робота за комп'ютерами)

💻 Практична робота 1 – «Алгоритмічний конструктор» (STEM-компонент): Інструкції для учнів:

1. Отримайте завдання: скласти блок-схему алгоритму для простої задачі (наприклад, «полив квітки», «приготувати чай»).
2. Використайте символи блок-схеми:
   - Овал – початок/кінець;
   - прямокутник – дія;
   - ромб – перевірка умови.
3. Після складання блок-схеми, переведіть її у програму на Python:
   - напишіть команди для виконання алгоритму;
   - перевірте, чи працює програма.

Приклад:

- Алгоритм «полив квітки»:
  1. Перевірити, чи горщик сухий;
  2. Якщо сухий, полити водою;
  3. Якщо ні, нічого не робити.
- в Python:

```
is_dry = input("Горщик сухий? (так/ні) ")
if is_dry == "так":
    print("Поливаємо квітку")
else:
    print("Не поливаємо")
```

💻 Практична робота 2 – «Програма для життя» (інформаційно-комунікативна компетентність):

1. Виберіть просту задачу із життя (наприклад, «приготувати чай», «зробити уроки»).
2. Складіть словесний алгоритм.
   - Переведіть алгоритм у Python, використовуючи змінні, input, print, умовні оператори if, як у прикладі «полив квітки».

3. Обміняйтеся програмами з сусідом та протестуйте, чи правильно виконуються кроки.

IX. Узагальнення та систематизація знань

Питання для самоперевірки:

1. Назвіть способи подання алгоритмів.
2. Чим відрізняється алгоритм від програми?
3. Які основні поняття мови Python ви засвоїли сьогодні?
4. Напишіть простий алгоритм та програму для нього.

X. Домашнє завдання

- Підручник §19-21.
- Додатково: скласти алгоритм свого ранкового ритуалу та перевести його у програму на Python (3-5 кроків).

XI. Підсумки уроку

Рефлексія:

- Учням пропонується закінчити речення:
  - «Сьогодні я дізнався…»
  - «Було цікаво…»
  - «Я хотів би спробувати…»

План-конспект № 3

Клас: **5**

Тема: **«Лінійні алгоритми і програми»**

Мета: ***Формування компетентностей***

Предметна компетентність:

- **Навчальна:** ознайомити учнів із лінійними алгоритмами, навчити складати послідовні команди в програмі Python; формувати навички розв'язування простих задач за допомогою лінійних алгоритмів.
- **Розвивальна:** розвивати логічне мислення, вміння аналізувати та планувати послідовність дій.
- **Виховна:** формувати культуру безпечної роботи за комп'ютером, уміння співпрацювати у групі.

Ключові компетентності:

- **Спілкування державною мовою:** описувати алгоритми та пояснювати послідовність команд.
- **Основні компетентності у природничих науках і технологіях:** застосовувати логіку та алгоритмічне мислення для вирішення практичних завдань.
- **Уміння вчитися впродовж життя:** організовувати власну діяльність із використанням Python.
- **Соціальна та громадянська компетентність:** дотримуватися правил безпеки життєдіяльності при роботі з комп'ютерами.

Тип уроку: засвоєння нових знань та формування вмінь.

Обладнання та наочність: дошка, комп'ютери, підручники «Інформатика, 5 клас», презентація, проектор.

Хід уроку

I. Організаційний етап

- Привітання, перевірка готовності, підготовка робочих місць.

II. Перевірка домашнього завдання

- Обговорення алгоритмів ранкового ритуалу, перевірка, як учні перенесли їх у Python.

III. Фронтальне опитування

1. Що таке алгоритм?
2. Які способи подання алгоритмів ви знаєте?
3. Що таке програма?
4. Які основні поняття Python ми вже засвоїли?
5. Які типи алгоритмів ви пам’ятаєте?

IV. Мотивація навчальної діяльності

У багатьох задачах дії виконуються одна за одною, крок за кроком, без перевірок чи умов. Такі алгоритми називають лінійними. Сьогодні ми навчимося створювати лінійні алгоритми та писати їх у Python, а потім перевіримо роботу наших програм на практиці.

V. Повідомлення теми, цілей та завдань уроку

- Тема: «*Лінійні алгоритми і програми*».
- Мета: навчитися складати лінійні алгоритми, перекладати їх у Python та перевіряти виконання.

VI. Сприймання й усвідомлення нового матеріалу

Пояснення вчителя з презентацією та демонстрацією на екрані:

1. **Лінійний алгоритм:** послідовність команд, які виконуються одна за одною без розгалужень.
2. **Приклади лінійних алгоритмів:**
   - приготування чаю;
   - вимірювання температури;
   - виконання вправ фізкультхвилинки.
3. **Запис у Python:**
   - Використання print та input;
   - Послідовне виконання команд;
   - Коментарі для пояснення дій.

Приклад лінійного алгоритму в Python:

```
# Лінійний алгоритм: приготування чаю
print("Кип'ятимо воду")
print("Наливаємо воду у чашку")
print("Додаємо чайний пакетик")
print("Чекаємо 5 хвилин")
print("Чай готовий!")
```

VII. Фізкультхвилинка

- Легка розминка для очей і рук (рухи руками, обертання плечима, кілька глибоких вдихів).

VIII. Засвоєння нових знань, формування вмінь (практична робота за комп'ютерами)

💻 Практична робота 1 – «Лінійна програма для робота» (STEM-компонент):
Інструкції для учнів:

1. Отримайте завдання: скласти лінійний алгоритм для робота, який рухається по прямій доріжці.
2. Запишіть послідовність команд: вперед, поворот, зупинка.
3. Реалізуйте алгоритм у Python, використовуючи print для кожної команди.
4. Перевірте виконання алгоритму, обговоріть можливі помилки та виправлення.

Приклад завдання:

- Робот має пройти 3 кроки вперед, повернутися праворуч, зробити ще 2 кроки.

```
print("Рух вперед 1 крок")
print("Рух вперед 2 крок")
print("Рух вперед 3 крок")
print("Поворот праворуч")
print("Рух вперед 1 крок")
print("Рух вперед 2 крок")
```

Практична робота 2 – «Лінійний алгоритм у житті» (інформаційно-комунікативна компетентність):

1. Оберіть просту дію з життя (наприклад, «зробити бутерброд»).
2. Складіть словесний алгоритм.
3. Переведіть алгоритм у Python, використовуючи print.
4. Поділіться з сусідом і перевірте правильність виконання.

Приклад: «Зробити бутерброд»

Словесний алгоритм:

1. Візьміть хліб.
2. Намажте хліб маслом.
3. Покладіть сир.
4. Покладіть ковбасу.

5. Накрийте другим шматком хліба.
6. Наріжте на половинки.

Python-код з використанням print():

```
print("1. Візьміть хліб")
print("2. Намажте хліб маслом")
print("3. Покладіть сир")
print("4. Покладіть ковбасу")
print("5. Накрийте другим шматком хліба")
print("6. Наріжте на половинки")
```

Додатково (STEM): «Рух робота по зірковій траєкторії»

Мета: розвивати просторове та логічне мислення, вміння складати послідовність команд у Python, практичне застосування лінійних алгоритмів.

Інструкція для учнів:

1. Уявіть, що на екрані ваш робот (або «котик» у Scratch/Python) має пройти **зіркову траєкторію**: рухатися вперед 2 кроки, повертати праворуч на 72°, повторювати 5 разів, щоб «намалювати» зірку.
2. Складіть **словесний алгоритм** руху робота.
3. Переведіть алгоритм у Python, використовуючи послідовність команд *print()* (для 5 класу поки що без циклів).
4. Перевірте, чи правильний результат: робот виконує команди у потрібній послідовності.
5. Для ускладнення: спробуйте скласти альтернативний алгоритм, де робот «малює» зірку, починаючи з іншого кута.

Приклад рішення в Python:

```
print("Рух вперед 2 кроки")
print("Поворот праворуч 72°")
print("Рух вперед 2 кроки")
print("Поворот праворуч 72°")
print("Рух вперед 2 кроки")
print("Поворот праворуч 72°")
print("Рух вперед 2 кроки")
print("Поворот праворуч 72°")
print("Рух вперед 2 кроки")
print("Поворот праворуч 72°")
```

Розвиток STEM-компоненту:

- зв’язок ***математика (кути, повтори) + технології (Python) + інженерне мислення (планування траєкторії)***.

IX. Узагальнення та систематизація знань

Питання для самоперевірки:

1. Що таке лінійний алгоритм?
2. Чим відрізняється лінійний алгоритм від розгалуженого?
3. Як записати лінійний алгоритм у Python?
4. Складіть коротку програму для щоденної дії.

X. Домашнє завдання

- Підручник §22.
- Додатково: скласти лінійний алгоритм для улюбленого настільного або комп’ютерного заняття та перевести його у Python (3-6 команд).

XI. Підсумки уроку

Рефлексія:

- Закінчити речення:
  - «Сьогодні я навчився…»
  - «Мені було цікаво…»
  - «Я хотів би спробувати…»

План-конспект № 4

Клас: **5**

Тема: **«Математичні оператори мови Python»**

Мета: ***Формування компетентностей***

Предметна компетентність:

- **Навчальна:** формувати знання про математичні оператори (+, -, *, /, %, //, **), навчитися складати лінійні алгоритми з використанням обчислень;
- **Розвивальна:** розвивати логічне мислення та навички структурованого оброблення інформації;
- **Виховна:** формувати обережне та уважне ставлення до роботи з комп’ютером.

Ключові компетентності:

- **Спілкування державною мовою:** формування уміння чітко описувати алгоритми та обчислення;
- **Основні компетентності у природничих науках і технологіях:** використання логіки, послідовності і обчислень для розв’язання задач;
- **Уміння вчитися впродовж життя:** організовувати власну роботу з комп’ютером та алгоритмами;
- **Соціальна та громадянська компетентність:** дотримання правил БЖД при роботі з ПК.

Тип уроку: засвоєння нових знань та формування вмінь.

Обладнання та наочність: дошка, комп’ютери, підручники (Бондаренко), презентація, проєктор.

Хід уроку

І. Організаційний етап

Привітання, перевірка готовності до уроку, налаштування комп’ютерів.

ІІ. Перевірка домашнього завдання

Обговорення попередньої практичної роботи: «Лінійний алгоритм у житті» (бутерброд, завдання від учнів).

ІІІ. Фронтальне опитування (актуалізація знань)

1. Що таке алгоритм?

2. Які ви знаєте математичні операції?
3. Що таке змінна в Python?
4. Які ви знаєте способи виводу результату на екран?
5. Як перевірити правильність алгоритму?

IV. Мотивація навчальної діяльності

Щодня ми обчислюємо різні значення: час, гроші, відстані. Щоб навчити комп'ютер робити це за нас, ми використовуємо математичні оператори. Сьогодні ми навчимося записувати обчислення в Python та складати лінійні алгоритми для реальних життєвих ситуацій.

V. Повідомлення теми, цілей та завдань уроку

- Ознайомлення з математичними операторами мови Python.
- Навчитися створювати лінійні алгоритми для обчислень.
- Розвивати навички командної роботи та інформаційно-комунікативну компетентність.

VI. Сприймання та усвідомлення нового матеріалу

- Пояснення вчителя + демонстрація презентації.
- Робота з підручником: § 21, ст. 134–137.

Математичні оператори Python:

- + – додавання
- - – віднімання
- * – множення
- / – ділення
- // – цілочислене ділення
- % – остача від ділення
- ** – піднесення до степеню

Приклади в Python:

```
a = 10
b = 3
print("Сума:", a + b)
print("Різниця:", a - b)
print("Добуток:", a * b)
print("Ділення:", a / b)
print("Цілочислене ділення:", a // b)
print("Остача від ділення:", a % b)
print("Степінь:", a ** b)
```

VII. Фізкультхвилинка

- 2-3 хвилини вправ для очей та рук.

VIII. Засвоєння нових знань, формування вмінь

Робота за комп'ютером (практична):

💻 Завдання 1 – STEM-компонент:

- Створіть програму, яка розраховує площу прямокутника.
- Використайте оператори * для множення.
- Додатково: змініть код так, щоб обчислити площу кількох прямокутників за різними розмірами.

Приклад коду:

```
# Площа прямокутника
length = 5
width = 3
area = length * width
print("Площа прямокутника:", area)

# Додатково: площа іншого прямокутника
length2 = 7
width2 = 4
area2 = length2 * width2
print("Площа іншого прямокутника:", area2)
```

💻 Завдання 2 – інформаційно-комунікативна компетентність:

- Виберіть життєву задачу з числами (наприклад, розрахунок загальної кількості яблук у двох кошиках).
- Складіть словесний алгоритм.
- Переведіть його у Python з використанням математичних операторів.
- Поділіться кодом з сусідом, перевірте правильність.

Приклад коду:

```
# Загальна кількість яблук
apples_basket1 = 12
apples_basket2 = 15
total_apples = apples_basket1 + apples_basket2
print("Загальна кількість яблук:", total_apples)
```

💻 STEM-завдання 3 – «Математичний калькулятор»

Інструкція для учнів:

1. Уявіть, що ви створюєте програму для обчислення сум, різниць, добутків та часток.
2. Виберіть два числа (можна із життя: кількість яблук та апельсинів у кошиках).
3. Обчисліть:
    - суму,
    - різницю,
    - добуток,
    - ділення з точністю до двох знаків після коми,
    - цілочислене ділення,
    - остачу від ділення.
4. Складіть словесний алгоритм для обчислень.
5. Переведіть алгоритм у Python, використовуючи print для виводу результатів.
6. Обміняйтеся кодом із сусідом та перевірте один одного.

Приклад коду:

```
# Обчислення для двох чисел
a = 17
b = 4

print("Сума:", a + b)
print("Різниця:", a - b)
print("Добуток:", a * b)
print("Ділення:", round(a / b, 2))
print("Цілочисельне ділення:", a // b)
print("Остача від ділення:", a % b)
```

IX. Узагальнення та систематизація знань

Питання для самоперевірки:

1. Які математичні оператори ви використали сьогодні?
2. Як змінюється результат, якщо змінити порядок дій?
3. Чим відрізняється ділення / від цілочисленого //?
4. Складіть алгоритм обчислення периметра прямокутника і реалізуйте його у Python.

X. Домашнє завдання

- § 23, ст. 134–137.
- Скласти власний лінійний алгоритм з 5–6 обчислювальними кроками та перевести його у Python.

XI. Підсумки уроку

Рефлексія:

- «Сьогодні я навчився…»
- «Було цікаво…»
- «Хочу дізнатися більше про…»

План-конспект № 5

Клас: **5**

Тема: **«Черепашача графіка»**

Мета: ***Формування компетентностей***

Предметна компетентність:

- навчальна: ознайомити учнів із поняттям «черепашача графіка», командами управління «черепахою»; формувати вміння створювати графічні малюнки за алгоритмами;
- розвивальна: розвивати логічне та алгоритмічне мислення, просторове уявлення;
- виховна: формувати інтерес до творчих завдань з використанням комп'ютера, культуру роботи в групі.

Ключові компетентності:

- спілкування державною мовою: формувати вміння пояснювати алгоритми та графічні дії словами;
- основні компетентності у природничих науках і технологіях: формувати вміння проектувати прості графічні об'єкти за алгоритмом;
- уміння вчитися впродовж життя: формувати навички самостійної роботи з Python;
- соціальна та громадянська компетентності: формувати дотримання правил безпечного використання комп'ютера.

Тип уроку: засвоєння нових знань і практична робота.

Обладнання та наочність: комп'ютери, проектор, підручник «Інформатика 5 клас», презентація.

Хід уроку

I. Організаційний етап

Привітання, перевірка присутності, налаштування робочих місць.

II. Перевірка домашнього завдання

Фронтальне опитування за попередніми темами: математичні оператори, лінійні алгоритми.

III. Актуалізація опорних знань
Фронтальне опитування:

1. Що таке алгоритм і які властивості він має?
2. Що таке лінійний алгоритм?
3. Назвіть основні математичні оператори Python.
4. Як використовувати команду print для відображення результату?

IV. Мотивація навчальної діяльності

Сьогодні ми навчимося малювати на екрані комп'ютера за допомогою «черепахи». Ця графіка допомагає уявити алгоритми візуально, розвиває просторове та логічне мислення, а ще можна створювати цікаві STEM-проєкти!

V. Повідомлення теми, цілей, завдань уроку

VI. Сприймання й усвідомлення учнями нового матеріалу

- Пояснення вчителя з демонстрацією команд черепахи у Python (turtle):
  - forward(), backward(), right(), left(), penup(), pendown(), color(), begin_fill(), end_fill().
- Показати приклад малюнка (трикутник, квадрат, будинок).
- Обговорення, як команди поєднуються в алгоритм для створення зображення.

VII. Фізкультхвилинка

VIII. Засвоєння нових знань, формування вмінь

💻 STEM-завдання 1 – «Малюємо свій прапор»

Інструкції для учнів:

1. Складіть словесний алгоритм малювання прапора України за допомогою прямокутників або смуг.
2. Переведіть алгоритм у Python, використовуючи turtle.
3. Обговоріть з сусідом, як можна оптимізувати код.

Приклад коду:

```
import turtle

t = turtle.Turtle()
t.speed(3)

# Малюємо смуги прапора
```

```
colors = ["blue", "yellow"]
for i in range(2):
    t.color(colors[i])
    t.begin_fill()
    t.forward(200)
    t.right(90)
    t.forward(50)
    t.right(90)
    t.end_fill()
    t.forward(50)  # переміщення на наступну смугу

turtle.done()
```

💻 STEM-завдання 2 – «Черепашка і лабіринт»

Інструкції для учнів:

1. Створіть невеликий лабіринт на екрані.
2. Використовуйте команди forward, left, right для «черепахи», щоб пройти лабіринт.
3. Складіть альтернативні алгоритми для проходження лабіринту.
4. Перевірте роботу алгоритмів із сусідом.

Приклад коду:

```
import turtle

t = turtle.Turtle()
t.speed(2)

# Простий лабіринт
t.forward(100)
t.left(90)
t.forward(50)
t.right(90)
t.forward(100)
t.left(90)
t.forward(50)

turtle.done()
```

Інформаційно-комунікативне завдання:

Обговорення:

1. Як алгоритми черепашки допомагають навчитися планувати дії.
2. Як STEM-підхід (алгоритмічне мислення + програмування + графіка) допомагає розвивати креативність.

IX. Домашнє завдання

- Підручник: §24.
- Придумати та намалювати власний простий малюнок черепашкою у Python.
- Скласти словесний алгоритм малюнка.

X. Підсумки уроку

Рефлексія:

- «Сьогодні я навчився…»
- «Мені було цікаво…»
- «Хочу у майбутньому спробувати…»

XI. Оцінювання роботи учнів

- Активність під час уроку, правильність команд, креативність малюнків.

План-конспект № 6

Клас: **5**

Тема: **«Алгоритми створення зображень»**

Мета: ***Формування компетентностей***

Предметні:

- навчальна: удосконалювати знання про алгоритмічне створення зображень; формувати вміння переводити словесний алгоритм у Python; вчитися працювати з кольорами та формами;
- розвивальна: розвивати логічне мислення, алгоритмічну культуру та креативність;
- виховна: виховувати естетичне сприйняття, терпіння та уважність при роботі з алгоритмами.

Ключові компетентності:

- спілкування державною мовою: описувати створені алгоритми та результати роботи;
- природничо-технічна: застосовувати логіку та системне мислення для побудови зображень;
- уміння вчитися: експериментувати з кодом, перевіряти та коригувати результати;
- соціальна: працювати в парі чи групі для обговорення та порівняння результатів.

Тип уроку: застосування знань і формування навичок.

Обладнання: комп’ютери з Python, середовище Turtle, дошка, презентація, підручник (Бондаренко).

Хід уроку

І. Організаційний етап

- Привітання, перевірка готовності робочих місць.

ІІ. Актуалізація опорних знань

Фронтальне опитування:

1. Що таке алгоритм?
2. Які основні команди Turtle ви пам’ятаєте?

3. Як змінювати колір лінії та заливки?
4. Як створювати повторювані елементи в Python?
5. Чому важливо планувати послідовність команд перед запуском коду?

III. Мотивація навчальної діяльності

Сьогодні ми навчимося створювати власні зображення за допомогою алгоритмів. Це не просто малювання, а програмування мистецтва! Ваші картинки можна оживити, зробити інтерактивними, додати кольори, повтори та навіть «перешкоди».

IV. Сприймання та усвідомлення нового матеріалу

- Пояснення вчителя з демонстрацією: базові команди Turtle (forward, backward, left, right, penup, pendown, color, begin_fill, end_fill).
- Приклади створення простих фігур: коло, квадрат, трикутник.
- Пояснення, як поєднувати фігури в складні зображення.

V. Засвоєння нових знань, формування вмінь (Практика)

Практична частина – STEM-завдання

💻 STEM-завдання 1 – «Кольоровий квадрат»

Інструкції для учнів:

1. Відкрийте Python і Turtle.
2. Намалюйте квадрат, використовуючи команди: forward і left.
3. Залийте квадрат кольором.
4. Спробуйте намалювати ще один квадрат поруч іншого кольору.

Приклад коду:

```
import turtle

t = turtle.Turtle()

                                                    # перший квадрат
t.color("red")
t.begin_fill()
t.forward(50)
t.left(90)
t.forward(50)
t.left(90)
t.forward(50)
t.left(90)
```

```
t.forward(50)
t.left(90)
t.end_fill()

                    # переміститиі «олівець» для другого квадрату
t.penup()
t.goto(60,0)
t.pendown()

                                        # другий квадрат
t.color("blue")
t.begin_fill()
t.forward(50)
t.left(90)
t.forward(50)
t.left(90)
t.forward(50)
t.left(90)
t.forward(50)
t.left(90)
t.end_fill()

turtle.done()
```

💻 STEM-завдання 2 – «Сонечко»

Інструкції для учнів:

1. Намалюйте коло – це буде сонце.
2. Додайте промені вручну, використовуючи команду forward і backward.
3. Кожен промінь намалюйте окремо, повертаючи Turtle вручну (без циклів).

Приклад коду:

```
import turtle

t = turtle.Turtle()

# коло (сонце)
t.color("yellow")
t.begin_fill()
t.circle(50)
t.end_fill()
```

```
                                                    # перший промінь
t.penup()
t.goto(0,50)  # верх кола
t.pendown()
t.forward(40)
t.backward(40)

                                                    # другий промінь
t.penup()
t.goto(50,0)  # право
t.pendown()
t.forward(40)
t.backward(40)

                                                    # третій промінь
t.penup()
t.goto(0,-50)  # низ
t.pendown()
t.forward(40)
t.backward(40)

                                                    # четвертий промінь
t.penup()
t.goto(-50,0)  # ліво
t.pendown()
t.forward(40)
t.backward(40)

turtle.done()
```

💻 STEM-завдання 3 – «Квіточка» (поетапно)

Інструкції для учнів:

1. Намалюйте маленьке коло – серединку квітки.
2. Додайте 4–5 пелюсток, використовуючи circle або forward + left.
3. Кожну пелюстку малюємо окремо.

Приклад коду:

```
import turtle

t = turtle.Turtle()
```

```
                                                    # серединка
t.color("yellow")
t.begin_fill()
t.circle(20)
t.end_fill()

                                                    # пелюстки
t.color("red")
# перша пелюстка
t.penup()
t.goto(0,20)
t.pendown()
t.begin_fill()
t.circle(20)
t.end_fill()

                                                    # друга пелюстка
t.penup()
t.goto(20,0)
t.pendown()
t.begin_fill()
t.circle(20)
t.end_fill()

                                                    # третя пелюстка
t.penup()
t.goto(0,-20)
t.pendown()
t.begin_fill()
t.circle(20)
t.end_fill()

                                                    # четверта пелюстка
t.penup()
t.goto(-20,0)
t.pendown()
t.begin_fill()
t.circle(20)
t.end_fill()

turtle.done()
```

VI. Фізкультхвилинка 🧠👀

Очі – вправо, вліво, вгору, вниз

VII. Узагальнення та систематизація знань

- Поясніть, яку користь дає використання алгоритмів для створення зображень.
- Обговоріть з сусідом, чому деякі алгоритми більш ефективні або швидкі.
- Обміняйтеся кодом та оцініть роботу однокласника.

VIII. Домашнє завдання

- Підручник: §25.
- Створити власний пейзаж (мінімум 3 різні об’єкти) у Turtle.
- Написати короткий словесний алгоритм до свого малюнка.

IX. Підсумки уроку та рефлексія

- «Для мене сьогодні важливим було…»
- «Мені сподобалося…»
- «Я хотів би ще навчитися…»

План-конспект № 7

Клас: **5**

Тема: **«Логічні вирази. Алгоритми і програми з розгалуженнями (оператор if)»**

Мета: ***Формування компетентностей***

Предметна компетентність:

- навчальна: удосконалити знання про логічні висловлювання; навчитися використовувати оператор if для прийняття рішень; формувати вміння перевіряти умови;
- розвивальна: розвивати логічне мислення та навички послідовного планування;
- виховна: виховувати обережність та відповідальність при роботі з ІТ.

Ключові компетентності:

- спілкування державною мовою: описати алгоритм власними словами;
- основні компетентності у природничих науках і технологіях: навчитися приймати рішення на основі умов;
- уміння вчитися впродовж життя: організовувати власну діяльність, застосовувати програмування;
- соціальна компетентність: дотримуватися правил безпеки під час роботи з комп’ютером.

Тип уроку: засвоєння нових знань

Обладнання: дошка, комп’ютери, Python, навчальна презентація.

Хід уроку

I. Організаційний етап

- Перевірка готовності комп’ютерів
- Нагадування правил БЖД

II. Перевірка домашнього завдання

- Обговорення попередніх практичних робіт з лінійними алгоритмами

III. Актуалізація опорних знань

Фронтальне опитування:

1. Що таке алгоритм?
2. Які основні властивості алгоритму ми знаємо?
3. Що таке виконавець алгоритму?
4. Які типи команд ми використовували раніше (forward, left, right)?
5. Що відбувається, якщо команда не виконана у потрібний момент?

IV. Мотивація навчальної діяльності

Щодня ми приймаємо рішення: одягати плащ, якщо йде дощ; брати парасольку, якщо небо похмуре. Так само в програмах ми можемо перевіряти умови та діяти по-різному. Сьогодні ми навчимося робити це в Python!

V. Повідомлення теми, цілей, завдань

VI. Сприймання нового матеріалу

Пояснення вчителя:

- Логічні висловлювання – це запитання, на які можна відповісти «так» або «ні».
- Оператор if дозволяє виконати певну команду **тільки якщо умова є правдивою**.
- Приклад простого логічного виразу:

```
if 3 > 2:
    print("3 більше 2")
```

- Учні бачать, що код виконується, якщо умова істинна.

VII. Практична робота

💻 Практичне завдання 1 – «Чи варто брати парасольку?» (STEM)

Інструкції для учнів:

1. Відкрийте Python.
2. Створіть змінну rain і присвойте їй True або False (дощить чи ні).
3. Використайте оператор if, щоб вивести повідомлення: «Бери парасольку», якщо rain = True.

Приклад коду:

```
rain = True  # якщо йде дощ, True, якщо ні – False

if rain:
```

```
print("Бери парасольку")
```

Варіант для ускладнення:

- Додати змінну temperature і виводити повідомлення: «Одягни куртку», якщо температура менше 15.

```
rain = True
temperature = 12

if rain:
    print("Бери парасольку")

if temperature < 15:
    print("Одягни куртку")
```

💻 Практичне завдання 2 – «Перевірка домашнього завдання» (інформаційно-комунікативна компетентність)

1. Створіть змінну homework_done (True/False).
2. Використайте if, щоб вивести: «Можеш грати на комп’ютері», якщо домашнє завдання виконано.

```
homework_done = True

if homework_done:
    print("Можеш грати на комп’ютері")
```

VIII. Фізкультхвилинка 🧠👀

Очі – вправо, вліво, вгору, вниз

IX. Узагальнення та систематизація знань

Питання для самоперевірки:

1. Що таке логічна умова?
2. Як працює оператор if у Python?
3. Які повідомлення можна виводити залежно від стану змінної?
4. Як би ви написали алгоритм «якщо світло зелене – йти, якщо червоне – стояти»?

X. Домашнє завдання

- Підручник: §26-27.
- Написати 2 прості програми з оператором if:
    1. «Візьми шапку, якщо холодно»

```
# Програма перевіряє температуру і радить, чи брати шапку

temperature = 10  # змініть число на поточну температуру

if temperature < 15:
    print("Візьми шапку!")
```

    2. «Пити воду, якщо спекотно»

XI. Підсумки уроку

- Рефлексія:
  «Сьогодні я дізнався…»
  «Мені цікаво спробувати…»

План-конспект № 8

Клас: **5**

Тема: **«Алгоритми і програми з розгалуженнями. Оператор if ... else»**

Мета: ***Формування компетентностей***

Предметна компетентність:

- навчальна: ознайомити учнів із оператором if ... else, формувати вміння створювати прості програми з розгалуженнями;
- розвивальна: розвивати логічне мислення та алгоритмічну грамотність;
- виховна: виховувати культуру безпечної роботи з комп'ютером.

Ключові компетентності:

- спілкування державною мовою: формування вміння висловлювати алгоритмічні дії словами та кодом;
- основні компетентності у природничих науках і технологіях: формування логічного мислення для аналізу ситуацій;
- уміння вчитися впродовж життя: застосування програмних засобів для планування та перевірки результатів;
- соціальна та громадянська компетентність: дотримання правил безпеки життєдіяльності під час роботи з ІТ.

Тип уроку: засвоєння нового матеріалу

Обладнання та наочність: дошка, комп'ютери, підручники, презентація

Хід уроку

I. Організаційний етап

Привітання, перевірка готовності до уроку, правила безпеки.

II. Перевірка домашнього завдання

Коротке обговорення попередніх практичних робіт, повторення лінійних алгоритмів.

III. Актуалізація опорних знань

Фронтальне опитування:

1. Що таке алгоритм?
2. Назвіть приклад лінійного алгоритму.

3. Що робить оператор if у Python?
4. Як визначити умову для виконання дії?
5. Наведіть приклад алгоритму з розгалуженням у житті.

IV. Мотивація навчальної діяльності

Ми часто приймаємо рішення у житті: брати парасольку, якщо йде дощ, або пити воду, якщо спекотно. Комп'ютер теж може «вирішувати» за нас, використовуючи оператори розгалуження. Сьогодні ми навчимося записувати такі рішення у Python.

V. Повідомлення теми, цілей, завдань

Тема: «*Алгоритми і програми з розгалуженнями. Оператор if ... else*». Мета: навчитися створювати програми з умовами та альтернативними діями.

VI. Сприймання й усвідомлення нового матеріалу

Пояснення вчителя з демонстрацією презентації:

- Оператор if ... else дозволяє вибирати між двома діями.
- Синтаксис:

```
if умова:
  дія_1
else:
  дія_2
```

- Приклад:

```
temperature = 10

if temperature < 15:
    print("Візьми шапку!")
else:
    print("Шапку можна залишити вдома")
```

VII. Засвоєння нових знань, формування вмінь

Робота за комп'ютером: унікальні завдання STEM та інформаційно-комунікативні:

💻 Практична робота 1 – «Що робити, якщо?» (STEM):

1. Створіть змінну weather з можливими значеннями: "сонячно", "дощ".
2. Напишіть програму з if ... else, яка виводить:
    - "Йдемо на прогулянку", якщо сонячно;
    - "Беремо парасольку", якщо дощ.

Приклад коду:

```
weather = "дощ"

if weather == "сонячно":
    print("Йдемо на прогулянку")
else:
    print("Беремо парасольку")
```

💻 Практична робота 2 – «Щоденний вибір» (інформаційно-комунікативна компетентність):

- Оберіть дію з життя: наприклад, «Пити воду чи чай»
- Складіть алгоритм словесно: «Якщо спекотно – пити воду, інакше – чай»
- Переведіть його у Python за допомогою if ... else.

Приклад коду:

```
temperature = 27

if temperature > 25:
    print("Пий воду")
else:
    print("Можна чай")
```

VIII. Фізкультхвилинка 🧠👀

Очі – вправо, вліво, вгору, вниз

IX. Узагальнення та систематизація знань

Питання для самоперевірки:

1. Чим відрізняється if від if ... else?
2. Наведіть приклад алгоритму з альтернативними діями.
3. Складіть простий алгоритм на Python для перевірки числа: якщо число парне – вивести «Парне», інакше – «Непарне».

X. Домашнє завдання

- Підручник: §28.
- Скласти 2 простих алгоритми з if ... else для свого життя (наприклад, «одягнути куртку», «їсти морозиво»).

XI. Підсумки уроку

Рефлексія:
Учні закінчують речення:

- «Сьогодні я дізнався…»
- «Тепер я можу…»
- «Мені цікаво спробувати…»

План-конспект № 9

Клас: **5**

Тема: **«Алгоритми з повтореннями. Цикл із лічильником»**

Мета: ***Формування компетентностей***

Предметна компетентність:

- **навчальна:** сформувати уявлення про алгоритми з повтореннями; ознайомити з циклом із лічильником у Python (for); навчити створювати прості програми з повторенням дій;
- **розвивальна:** розвивати логічне та алгоритмічне мислення;
- **виховна:** виховувати відповідальне ставлення до роботи з комп'ютером.

Ключові компетентності:

- спілкування державною мовою – вміння пояснювати алгоритм словами;
- компетентності у природничих науках і технологіях – розуміння повторюваних процесів;
- уміння вчитися впродовж життя – планування послідовності дій;
- соціальна та громадянська – дотримання правил БЖД.

Тип уроку: Засвоєння нових знань

Обладнання: Комп'ютери, дошка, презентація, середовище Python.

Хід уроку

I. Організаційний етап

Привітання. Нагадування правил безпечної роботи за комп'ютером.

II. Актуалізація опорних знань (фронтальне опитування)

1. Що таке алгоритм?
2. Який алгоритм називається лінійним?
3. Чи є у вашому житті дії, які повторюються щодня?
4. Чи зручно писати одну й ту саму команду багато разів?
5. Як би ви пояснили комп'ютеру команду «повтори 5 разів»?

III. Мотивація навчальної діяльності

Уявіть, що вам потрібно 10 разів написати фразу «Я люблю інформатику». Писати її 10 разів – довго і нудно. А що, якщо комп'ютер може зробити це сам? Сьогодні ми навчимося **наказувати комп'ютеру повторювати дії**.

IV. Повідомлення теми і мети уроку

Тема: **Алгоритми з повтореннями. Цикл із лічильником**
Мета: навчитися створювати прості програми з повторенням дій у Python.

V. Вивчення нового матеріалу

- **Алгоритм з повтореннями** – це алгоритм, у якому деякі дії виконуються кілька разів.

- **Цикл із лічильником** – це команда, яка повторює дію задану кількість разів.

!!!! У Python використовується команда for.

Загальний вигляд:

```
for i in range(кількість):
    команда
```

Пояснення простими словами:

«Виконай команду стільки разів, скільки я скажу»

VI. Засвоєння знань. Робота за комп'ютером

💻 Практична робота 1 – «Повтори привітання»

Завдання для учнів:

1. Напишіть програму, яка 5 разів виведе повідомлення «Привіт, 5 клас!».

Код:

```
for i in range(5):
    print("Привіт, 5 клас!")
```

💻 Практична робота 2 – «Лічба для робота» (STEM-компонент)

Завдання:
Робот вчиться рахувати. Допоможіть йому вивести числа від 1 до 5.

Код:

```
for i in range(1, 6):
    print(i)
```

💬 Обговорення:

- Де в житті нам потрібна лічба?
- Чи може робот рахувати швидше за людину?

💻 Практична робота 3 – «Шкільний дзвінок» (унікальне, життєве)

Завдання:
Шкільний дзвінок дзвонить 3 рази. Напишіть програму.

Код:

```
for i in range(3):
    print("Дзень!")
```

VII. Фізкультхвилинка 🧠👀

Очі – вправо, вліво, вгору, вниз

VIII. Узагальнення і систематизація знань

Питання для самоперевірки:

1. Що таке цикл?
2. Для чого використовується команда for?
3. Як змінити кількість повторень у програмі?
4. Який алгоритм зручніший: з повторенням чи без?

IX. Домашнє завдання

1. Підручник: §29.
2. Створити програму, яка 4 рази виводить фразу **«Я вчу Python»**
3. Придумати приклад повторення з життя (словесно).

X. Підсумок уроку. Рефлексія

Учні завершують речення:

- «Сьогодні я навчився…»
- «Мене здивувало…»
- «Я можу використати цикл, коли…»

Висновок

Формування інформаційно-комунікативної компетентності учнів у сучасній школі є безперервним і динамічним процесом, що безпосередньо пов’язаний із розвитком цифрового суспільства. Особливістю цієї компетентності є її практичний характер: вона проявляється не стільки у володінні окремими знаннями, скільки у здатності учня працювати з інформацією, цифровими пристроями, алгоритмами та програмами у реальних життєвих ситуаціях.

Саме тому курс інформатики у 5 класі виступає фундаментом її формування. Початкове вивчення алгоритмів і програмування створює основу для розвитку логічного, структурного та критичного мислення. Учень не лише засвоює поняття алгоритму, команди чи виконавця, а поступово починає усвідомлювати принципи роботи цифрового світу: передбачуваність, точність, послідовність дій і відповідальність за результат.

У ході роботи було встановлено, що ефективне формування інформаційно-комунікативної компетентності можливе за умов поєднання традиційного пояснення з діяльнісним підходом. Найбільший результат дає організація навчання через практику: створення алгоритмів для життєвих ситуацій, програмування в середовищі Python, робота з «черепашачою» графікою, побудова власних моделей і розв’язання проблемних задач.

Важливою складовою запропонованої методики стала інтеграція STEM-підходу. На відміну від типових вправ на відтворення команд, учні виконували завдання дослідницького характеру: створювали алгоритми руху в лабіринті, моделювали поведінку об’єктів, будували зображення за правилами, аналізували умови та передбачали результат виконання програм. У такій діяльності програмування виступало не метою, а інструментом розв’язування задач.

Практика показала, що саме через STEM-діяльність відбувається природне поєднання предметних і ключових компетентностей. Під час роботи учні:

- планують власні дії (інженерне мислення);
- перевіряють гіпотези (дослідницька діяльність);

- аналізують помилки (критичне мислення);
- працюють у групі (соціальна взаємодія);
- пояснюють алгоритми (комунікація);
- усвідомлюють роль технологій (цифрова грамотність).

Важливим результатом стало зростання навчальної мотивації. Молодші учні значно активніше залучаються до роботи тоді, коли бачать миттєвий результат: рух об’єкта, зміну зображення, реакцію програми на умову. Отже, програмування на початковому рівні повинно будуватись не навколо синтаксису, а навколо ідеї керування процесами.

Запропонована система уроків демонструє, що навіть базові теми – лінійні алгоритми, розгалуження, повторення – можуть стати інструментом формування складних умінь: планування діяльності, прогнозування результату, аналізу інформації та прийняття рішень. У цьому випадку Python використовується як зрозуміла учню мова взаємодії з технологією, а не як складна технічна дисципліна.

Таким чином, поєднання алгоритмічного навчання, практичного програмування та STEM-підходу створює освітнє середовище, у якому учень переходить від споживання цифрового контенту до його усвідомленого створення. Саме цей перехід і є основним показником сформованості інформаційно-комунікативної компетентності.

Отже, розроблена методична система доводить: формування інформаційно-комунікативної компетентності в 5 класі є найбільш ефективним тоді, коли учні не просто вивчають інформатику, а починають мислити як розробники – ставлять задачу, будують алгоритм, перевіряють результат і вдосконалюють рішення. Саме така діяльність формує основу подальшого навчання, професійного самовизначення та успішної взаємодії з цифровим світом майбутнього.

.

Використані джерела

Нормативні документи та стандарти освіти

1. Державний стандарт базової середньої освіти. – К.: Кабінет Міністрів України, 2020.
2. Типова освітня програма для 5–9 класів закладів загальної середньої освіти (НУШ). – МОН України.
3. Концепція «Нова українська школа». – МОН України, 2016.
4. Рамка цифрової компетентності для громадян України (DigCompUA). – Міністерство цифрової трансформації України.
5. Санітарний регламент для закладів загальної середньої освіти (робота з комп'ютером).

Українські підручники та методика навчання інформатики

1. Бондаренко О. О., Ластовецький В. В., Пилипчук О. П., Шестопалов Є. А. **Інформатика. 5 клас.** – Харків: Ранок.
2. Морзе Н. В., Барна О. В., Вембер В. П. **Методика навчання інформатики в школі.** – Київ.
3. Ривкінд Й. Я., Лисенко Т. І., Чернікова Л. А., Шакотько В. В. **Інформатика: підручники та методичні рекомендації.**
4. Калінін В., Калініна Л. **Інноваційні технології навчання інформатики.** – К., 2018.
5. Гуржій А. М., Карташова Л. А. **ІКТ в освіті та формування цифрової компетентності.**
6. Співаковський О. В. **Теорія і практика навчання інформатики в школі.**
7. Мельник Ю. **Дистанційне навчання та цифрові технології в освіті.** – Дніпро, 2017.

STEM-освіта та компетентнісний підхід

1. Лойко Н.П. **STEM-проєкти в школі.** – Львів, 2019.
2. Василенко О. М. **STEM-освіта: теорія і практика впровадження.**
3. Горобченко Н., Ващенко Т. **Інформаційно-комунікативна компетентність учнів, -** К., 2014.
4. Бондаренко Т. **Формування ІКК у школярів.** – К.: Видавництво «Шкільний світ», 2016
5. Алієв Х. **Цифрове навчання в освітньому середовищі.** – К., 2018.

Онлайн-ресурси:

Українські освітні платформи

1. Всеукраїнська школа онлайн – https://lms.e-school.net.ua

2. На Урок – https://naurok.com.ua
3. Всеосвіта – https://vseosvita.ua
4. Освіторія – https://osvitoria.media
5. Інститут модернізації змісту освіти – https://imzo.gov.ua

Матеріали з програмування для дітей

6. Code.org – https://code.org
7. Scratch – https://scratch.mit.edu
8. Trinket (онлайн Python) – https://trinket.io