

КИЇВСЬКИЙ ОБЛАСНИЙ ІНСТИТУТ
ПІСЛЯДИПЛОМНОЇ ОСВІТИ ПЕДАГОГІЧНИХ КАДРІВ
НАУКОВА ЛАБОРАТОРІЯ
ІНТЕРАКТИВНИХ ТЕХНОЛОГІЙ НАВЧАННЯ

В.С. ОСТАПЕЦЬ

**РОЗВ'ЯЗАННЯ АКТУАЛЬНИХ ПРОБЛЕМ
ШКІЛЬНОЇ ІНФОРМАТИКИ
ІЗ ЗАСТОСУВАННЯМ ІННОВАЦІЙНИХ
ТЕХНОЛОГІЙ НАВЧАННЯ
ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ**

*(з досвіду роботи ОШППД
“Шляхи розв'язання актуальних проблем шкільної
інформатики”
при КОІПОПК у 2007-2010 р.р.)*

2009 р.

Рекомендовано до друку

Вченою радою

Київського обласного інституту післядипломної освіти педагогічних кадрів

(Протокол № ____ від “ ____ “ _____ 2010 року)

Рецензенти: **Федорчук В.А.,**

Завідуючий

центром інформаційних технологій навчання

Київського обласного інституту

післядипломної освіти педагогічних кадрів

Скляр І.В.,

викладач УФМЛ КПІ ім. Т. Шевченка,

науковий редактор газети "Інформатика",

Заслужений учитель України

Остапець В.С.

РОЗВ'ЯЗАННЯ АКТУАЛЬНИХ ПРОБЛЕМ ШКІЛЬНОЇ ІНФОРМАТИКИ

ІЗ ЗАСТОСУВАННЯМ ІННОВАЦІЙНИХ ТЕХНОЛОГІЙ НАВЧАННЯ

ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

(з досвіду роботи ОШППД "Шляхи розв'язання актуальних проблем

шкільної інформатики" при КОІПОПК у 2007-2010 р.р.)

ЗМІСТ:

Розв'язання актуальних проблем шкільної інформатики із застосуванням інноваційних технологій навчання та інформаційно-комунікаційних технологій	4
I. Актуальні проблеми курсу основ інформатики та їх дослідження.	4
II. Місце інформаційно-комп'ютерних технологій у викладанні інформатики.	9
II.1 Електронні конспекти уроків.	11
II.2 Приклад ЕКУ.	13
II.3 Електронні сценарії уроків.	18
III. Застосування методу проектів у процесі викладання основ інформатики.	23
IV. Вивчення алгоритмізації та програмування у формі експрес-курсу.	26
IV.1 Проблеми вивчення теми "Комп'ютерне моделювання. Основи алгоритмізації та програмування" у школі.	26
IV.2 Методичне обґрунтування експрес-курсу алгоритмізації та програмування	27
IV.3 Висновки.	34
V. Міжпредметні зв'язки та математична складова курсу основ інформатики.	37
V.1 Значення міжпредметних зв'язків для інформатики.	37
V.2 Бінарні уроки.	40
V.3 Зразки бінарних уроків.	42
V.4 Зв'язки інформатики з виховною роботою.	50
VI. Методична робота учителя інформатики.	52
VI.1 Робота з дидактичними матеріалами як складова методичної роботи.	56
VII. Особливості викладання основ інформатики в 9-у класі.	59
VIII. Організація позакласної роботи з основ інформатики.	69
VIII.1 Про сучасні олімпіади з інформатики.	71
VIII.2 Якою повинна бути олімпіадна задача?	75
IX. Замість післямови.	78
Література.	80

До цього збірника можна було б додати методичні матеріали понад півтора десятків із 32 вчителів-слухачів з Київської області цієї ОШПД (обласної школи передового педагогічного досвіду) що також були надруковані в збірнику майже на 100 сторінках, проте значна частина цих матеріалів не належить мені, тим більше, значною мірою протягом 2008-10 років вони були опубліковані авторами окремо.

**РОЗВ'ЯЗАННЯ АКТУАЛЬНИХ ПРОБЛЕМ ШКІЛЬНОЇ ІНФОРМАТИКИ
ІЗ ЗАСТОСУВАННЯМ ІННОВАЦІЙНИХ ТЕХНОЛОГІЙ НАВЧАННЯ
ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

*(з досвіду роботи ОШППД "Шляхи розв'язання актуальних проблем шкільної інформатики"
при КОППОК у 2007-2010 р.р.)*

I. Актуальні проблеми курсу основ інформатики та їх дослідження.

Протягом 2007-2010 навчальних років на базі Щасливського НВК Бориспільського району відповідно до Наказу **Управління освіти і науки Київської обласної державної адміністрації від 02. 04.2004 р. № 221** працює обласна школа передового педагогічного досвіду¹ "Шляхи розв'язання актуальних проблем шкільної інформатики". Її робота спрямована на пошуки ефективних форм і методів подолання об'єктивних проблем, які накопилися протягом двох десятиріч існування шкільного предмету "Основи інформатики". Вони виникали через недосконалість навчальних програм і підручників, недостатню кількість передбачених програмами навчальних годин, бідність та несвоєчасність методичного супроводу, а також завдяки небаченому розвитку комп'ютерної техніки і програмного забезпечення, що вимагає докорінних змін в інформаційній культурі та інформативній компетентності всього суспільства.

Сучасні пошуки оптимальних шляхів організації навчально-виховного процесу сконцентровані навколо пріоритетних нині інноваційних технологій навчання (ІТН) та інформаційно-комунікаційних технологій навчання (ІКТ). З одного боку – це правильно, бо продиктовано доцільністю, адже світовий розвиток, чим далі - тим сильніше, зазнає впливу процесів глобалізації, і освітня галузь, не зважаючи на її традиційно консервативний характер, не може стояти осторонь від них. Але, з іншого боку, ці пріоритети вже по своїй природі *суб'єктивні*, тобто такі, що не можуть бути абсолютними й довготривалими. Досить згадати, для прикладу, проблеми глобального потепління в природі. Адже вони скоріше гіпотетичні, ніж наукові, бо успішно пояснюються та підтверджуються циклічністю природних процесів, які спостерігались протягом всієї історії людства.

Хоча для всіх понять намагаються дати чіткі тлумачення, абсолютних серед них не існує. Наведемо, так би мовити, канонічний приклад, про основні поняття геометрії: точку, пряму чи площину. Через описовий характер різні люди, хоча б у дрібних деталях, уявляють ці поняття по-різному, причому така обставина не заважає їм розуміти, що мається на увазі одне і те ж. Ще в більшій мірі це стосується похідних понять. Отже, можна стверджувати, що й поняття "інноваційні технології навчання" та "інтерактивні методи" мають в деякій мірі приблизний зміст, а інколи їх свідомо чи не свідомо плутають. І все ж, беручи до уваги думку Р. Декарта: "Дайте точні визначення поняттям і ви позбавите людство від половини його помилок", в пошуках істини потрібно постійно намагатись уточнювати терміни. Слово *інновація* в перекладі з латинської мови означає оновлення, зміну, введення нового.

¹ Далі вживається скорочена назва ОШППД.

У педагогічній інтерпретації інновація означає нововведення, що поліпшує хід і результати навчально-виховного процесу.

На наш погляд, треба бути обережними в оцінці ролі впровадження ІТН та ІКТ. Відігравши роль засобу *перегрупування освітніх ресурсів на базі сучасних технічних досягнень*, вони з часом стануть звичними складовими навчально-виховного процесу. Отже, інноваційні та комп'ютерні технології навчання, незважаючи на їх всеохоплюючий характер, не можуть претендувати на роль провідних напрямків розвитку освітньої галузі.

На таких міркуваннях ґрунтувався вибір проблем та шляхів їх вирішення для трирічного періоду діяльності ОШПД. По-перше, ІКТ на уроках інформатики через специфіку предмета вже давно стали звичними й ординарними засобами навчання. По-друге, використання ІКТ може бути ефективним лише при умові доцільності. По-третє, застосування інформаційно-комп'ютерних технологій є окремою проблемою навчально-виховного процесу в цілому, тому дослідження в цій галузі повинні бути винесеними за межі методики викладання основ інформатики.

Сьогодні дії надто часто підміняються *деклараціями*. Це стосується й використання інноваційних технологій навчання, адже практично всі так звані інноваційні методи навчання були давно відомими, правда, на них не часто акцентувалась увага. У висновках авторитетних комісій, які років 40 тому вивчали унікальний досвід знаменитого учителя математики В.Ф. Шаталова, констатувалось, що всі понад 200 дидактичних прийомів, які він широко застосовував у своїй практиці, були широко відомими. У зв'язку з цим його досвід, не зважаючи на високу результативність, не визнавався новаторським. Але у наш час більшість методів та прийомів із арсеналу В.Ф. Шаталова належать до інноваційних технологій навчання. Ради справедливості слід зауважити, що в ті роки носіями передового педагогічного досвіду були цілі плеяди учителів-новаторів. Варто також сказати, що тотальне, здебільшого декларативне, вживання терміну "інновації" замість впровадження досвіду учителів-новаторів у наш час приносить більше шкоди, ніж практичної користі.

Впровадження вище згаданих ІКТ, нерідко також підноситься до рангу інноваційних технологій. При цьому, через недостатню підготовку широкого кола педагогічних працівників, використання комп'ютерів здебільшого зводиться до ролі звичайних технічних засобів навчання, які використовуються для аудіовізуального супроводу навчальних занять чи виховних заходів. Але використання лише демонстраційних функцій сучасних комп'ютерів (*демонстраційна складова*) не може виправдати навіть матеріальних затрат на них.

Ще в 60-80-і роки минулого сторіччя, коли в навчально-виховному процесі масово використовувались такі традиційні ТЗН, як кінопроектори, діаскопи, кодоскопи, програвачі, телевізори, було централізовано створено великі навчально-пізнавальні фільмотеки, фонотеки, діатеки тощо. Але дуже швидко стало зрозуміло, що все це малоефективне. Дійсно, навчальні кінофільми чи діафільми могли використовуватись лише епізодично,

фрагментарно, на вступних чи підсумкових етапах вивчення теми, вимагали значного часу для монтажу і експлуатації навчальної техніки, мало відповідали конкретним умовам на уроці. Тому на практиці технічні засоби навчання застосовувались рідко. Природно, що при розгляді комп'ютера в якості демонструючого ТЗН, сучасні, навіть найкращі, ліцензовані ПНЗ успадкували три дуже суттєві хиби тих фільмо-, фоно- та діатек. Перша – *консервативність*, адже їх важко адаптувати до потреб учителя з конкретними педагогічними ситуаціями, навпаки, використання ПНЗ змушує учителя самому пристосовуватись до них. Це *нівелює творчий характер навчального процесу*, якщо не ставити за мету провести показовий, "відкритий" урок. Друга – *переважно інформаційний характер ПНЗ*. Учителю ж необхідні динамічні, із експериментальними, а не тільки демонстраційними та тестуючими можливостями програмні засоби. Нарешті, третя - *всеоб'ємність*, тобто поверхове охоплення всього курсу.

Справжньою метою застосування ІКТ повинна бути *інтенсифікація розумової діяльності під час навчально-виховного процесу*. Тут маються на увазі можливості комп'ютерної техніки в моделюванні та проведенні чисельних експериментів, статистичних та аналітико-синтетичних досліджень, автоматизації пошукової роботи, класифікації й експертних оцінок, тестування і т.д., тобто *аналітичної складової* можливостей сучасної комп'ютерної техніки. Один із варіантів схеми досягнення цієї мети такий:



схема I.1

Зі схеми I.1 видно, що застосування ІКТ в навчально-виховному процесі доцільно розділяти на три етапи, які можна умовно прив'язати до вікових категорій учнів, зокрема: I етап – школа I ступеня, II етап – школа II ступеня, III етап – школа III ступеня. Але, залежно від категорії навчального закладу,

термінів початку та рівня вивчення основ інформатики, етапи можна зміщувати.

З точки зору учителя-практика основними об'єктивними й актуальними проблемами сучасного курсу шкільної інформатики є:

1. *Не обґрунтованість встановлених діючими програмами пропорцій і акцентів між двома складовими – розділом “Курс інформаційно-комунікаційних технологій” (ІКТ - курс користувача комп'ютера) і розділом “Алгоритмізація та програмування” (АП);*
2. *Відсутність достатнього зв'язку між розділами ІКТ і АП;*
3. *Відсутність необхідного рівня зв'язків курсу “Основи інформатики” з іншими навчальними дисциплінами, у першу чергу з точними та природничими;*
4. *Відсутність правильного і чіткого усвідомлення місця курсу основ інформатики в системі загальної середньої освіти та у формуванні гармонійно розвиненої особистості;*
5. *Відсутність спільної бази між курсом основ інформатики та позакласною роботою з цього предмету.*

У 2009-2010 н.р. в зв'язку із введенням предмета “Основи інформатики” в навчальні плани для 9-х класів виникла ще одна проблема:

6. *Відсутність методичного забезпечення викладання основ інформатики в умовах 12-річної школи.*

Всі щойно перераховані проблеми взаємопов'язані (див. схему І.2). Дійсно, необхідна зміна пропорції між розділами "Практичний курс інформатики (курс користувача комп'ютера)" і "Алгоритмізація та програмування" в сторону збільшення долі останнього (проблема 1), але це приведе до збільшення присутності математичної компоненти в шкільному курсі основ інформатики, адже алгоритмізація та програмування – це, по суті, прикладна математика (проблема 3). Вкрай необхідне встановлення тіснішого зв'язку між вищеназваними складовими шкільного курсу інформатики (проблема 3), у свою чергу це веде до фактичного проникнення програмування, у сучасному варіанті візуального проектування та об'єктно-орієнтованого програмування, у більшість тем курсу основ інформатики (проблема 1). Особливо актуальна сьогодні потреба формування сучасної інформаційної культури та інформативної компетентності учнів (проблема 4) залежить від усвідомлення місця курсу основ інформатики в навчально-виховному процесі (проблема 3). Нарешті, без ефективної позакласної роботи з предмету (проблема 5), яка дуже залежить від рівня культури програмування (проблема 1) та взаємопов'язаності вміння програмувати з навичками користувача (проблема 2) не можливо якісно організувати вивчення основ інформатики в цілому. Очевидно, що проблеми 3 та 4 мають прямий зв'язок з проблемою 6. Виходячи з цього, сукупність проблем 1 - 6 доцільно сприймати, як єдину і масштабну проблему.

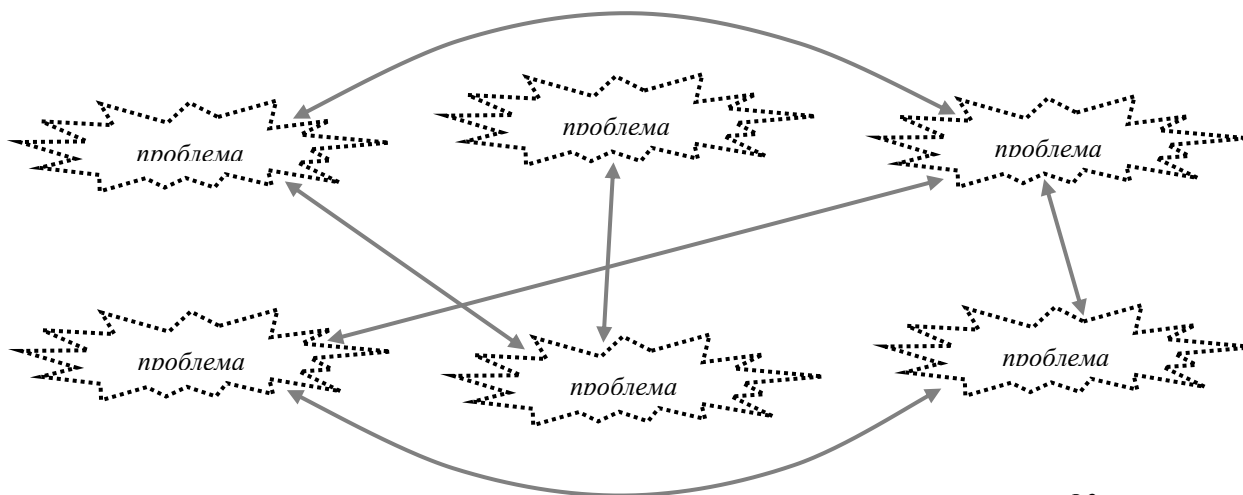


схема 1.2

Тому завдання, яке ставить перед собою ОШППД, повинне полягати в тому, щоб, спираючись на досвід керівника школи та її слухачів, забезпечити ефективні пошуки шляхів розв'язання цієї масштабної проблеми.

Необхідність пошуків ефективних форм і методів подолання названих проблем визначає основні напрямки роботи ОШППД:

1. Удосконалення методики проведення урочної та позакласної роботи з основ інформатики в умовах інтенсивного використання інформаційно-комп'ютерних та інтерактивних технологій навчання;
2. Розробка сучасних електронних дидактичних засобів навчання з основ інформатики у середовищі *Microsoft Office*;
3. Вивчення шляхів удосконалення організації позакласної роботи з основ інформатики;
4. Розробка методики поєднання індивідуальних та інтерактивних методів навчання у процесі викладання інформатики.

Отже, ІТН та ІКТ належить до основних напрямків роботи ОШППД (виділено курсивом у напрямку 1), тому вони є не предметом, а засобами навчальної та дослідницької діяльності.

Діяльність ОШППД ґрунтується на багаторічному досвіді роботи, провідною ідеєю якого є оптимізація викладання курсу основ інформатики з метою подолання актуальних проблем в умовах впровадження інформаційно-комп'ютерних технологій навчання, формування сучасної інформаційної культури та інформативної компетентності підростаючого покоління. Актуальність досвіду впливає з потреби піднесення ролі курсу основ інформатики для формування сучасної інформаційної культури та інформативної компетентності учнів.

ОШППД ставить за мету досягнення якісних результатів:

- а) У підвищенні рівня професійної майстерності вчителів інформатики;
- б) У формуванні сучасної інформаційної культури та інформативної компетентності учнів;
- в) У підвищенні рівня культури програмування вчителів та учнів.

Протягом трьох навчальних років у процесі роботи ОШППД було досліджено теми:

- I. Сучасні проблеми у викладанні основ інформатики;
- II. Місце ІКТ у викладанні основ інформатики;

- III. Застосування методу проектів у процесі викладання основ інформатики;
- IV. Вивчення теми “Алгоритмізація та програмування“ у формі експрес-курсу;
- V. Міжпредметні зв’язки та математична складова курсу основ інформатики;
- VI. Методична робота учителя інформатики;
- VII. Особливості викладання основ інформатики в 9-у класі за новими програмою та підручниками;
- VIII. Методичні особливості вивчення комп’ютерних мереж в 9 класі;
- IX. Методичні особливості вивчення векторної графіки в 9 класі;
- X. Організація позакласної роботи з основ інформатики.

Розгляд першої теми передбачав ініціювання основних актуальних проблем 1-5 курсу інформатики та шляхів їх розв’язування, тобто постановку загальних завдань перед ОШППД ([1], [2], [3])².

Зміст другої теми полягав у встановленні ролі ІКТ та ІТН, як засобів навчально-виховної роботи з основ інформатики. Акцентувалось на навчально-дидактичних можливостях сучасних комп’ютерів, співвідношенні між їх демонстраційною та аналітичною складовими. Розглянуто питання ефективності ліцензованих програмних засобів навчального призначення (ППЗ НП) з огляду на низькі ступені їх універсальності та своєчасності, а також складності використання, пов’язані з отриманням ліцензійних копій. Зроблено висновок 1: найбільш ефективними програмними засобами при вивченні основ інформатики на сучасному рівні ІКТ можуть стати так звані *електронні дидактичні засоби навчання* (ЕДЗ)³ та їх окремий різновид – *комплексні електронні дидактичні засоби* (КЕДЗ) ([4]). До складу ЕДЗ можна включити: електронні сценарії уроків (ЕСУ), електронні конспекти уроків (ЕКУ), електронні роздавальні матеріали (ЕРМ).

II. Місце інформаційно-комп’ютерних технологій у викладанні інформатики.

Для початку слід зауважити, що тут розглядаються лише саморобні ЕДЗ, про які мовилось вище, адже методичні поради по застосуванню ліцензованих ППЗ НП повинні включатись до їх складу, як частина, і захищатись авторським правом.

На уроках основ інформатики діючими програмами рекомендується використовувати практичну форму занять, тобто, на *кожному* уроці *кожен* учень повинен якомога більше працювати *безпосередньо* з комп’ютером. Щоб не відходити від цього принципу, очевидно, слід відмовитись від застосування проекторів, інтерактивних дошок та демонстраційних екранів. Тим більше, що є чудова можливість користуватись локальною мережею, з якою учні досить детально знайомляться ще на початкових етапах вивчення основ інформатики. Серед системного програмного забезпечення є ряд спеціальних програм, що дають можливість організувати дистанційне керування робочими місцями учнів (РМУ) з центрального комп’ютера локальної мережі. Як оптимальний варіант можна запропонувати *Net OpSchool* (версія 3-5). Перш за все з його допомогою учитель зможе легко організувати демонстрацію потрібної порції навчального

² У квадратних дужках вказується номер використаного джерела згідно списку літератури.

³ Поняття *електронних дидактичних засобів навчання* та їх окремих видів, а також умовні позначення введено автором для зручності опису.

матеріалу, що у поєднанні власного голосу учителя та маніпуляцій курсором миші практично виходить ще ефективніше, ніж використання демонстраційного екрану. Адже у цьому випадку кожен учень відчуває себе більш залежним від дій учителя, ніж коли він у складі групи спостерігає за спільним екраном. Програма *Net OpSchool* дає ще багато дидактичних можливостей, у тому числі спостерігати за подіями на екранах РМУ, блокувати лише вибрані з них, застосовувати режим індивідуального управління кожним комп'ютером, робити аудіо та візуальні акценти, проводити тестування та спілкування за зразком чату.

Під час уроку з використанням *Net OpSchool* такі види роботи легко поєднувати з індивідуальною практикою учнів. Суттєвим недоліком ліцензованих ППЗ НП є неможливість ефективного застосування *Net OpSchool*, адже вони підпорядковують собі ресурси локальної мережі. Задум описаних нижче видів ЕДЗ, навпаки, передбачає його постійне застосування. Враховуючи можливості самостійного виготовлення і редагування а також роботу у звичному для пересічних користувачів (якими є учителі й учні) середовищі *Microsoft Office*, ЕДЗ стають не тільки конкурентоздатною альтернативою застосуванню ліцензованих ППЗ НП, а інноваційною формою використання ІКТ в навчально-виховному процесі.

ЕДЗ, під якими слід розуміти файли чи пакети файлів, що містять будь-яку інформацію навчального характеру, дидактичний або наочний засіб, що розсилається на робочі місця учнів (РМУ) через локальну мережу для використання учнями під час роботи з комп'ютером. ЕДЗ, як уже зазначалось, не слід плутати із ліцензованими програмними продуктами навчального призначення, бо він виготовляється учителем, гнучкий за характером будови та застосування, може оперативно змінюватись, включати гіперпосилання, анімаційні чи звукові ефекти, адаптований до конкретної ситуації, має, як правило, локальний зміст, володіє багатофункціональністю.

ЕДЗ можуть містити теоретичні відомості з необхідними питаннями, завданнями, тестами, бути оснащеними таблицями, схемами, малюнками. Завдяки системі гіперпосилань ці документи мають динамічний характер, розсилка на РМУ через локальну мережу дає змогу використовувати їх оперативно і комплексно з системами програмування та навчальними пакетами. Бажано, щоб кожен ЕДЗ охоплював вузьку тему, тоді створювати його легко, при потребі завжди можна доопрацювати і удосконалити. Крім того, робота з ним урізноманітнить урок, зробить його більш динамічним, створить атмосферу позитивної мотивації вивчення навчального матеріалу, дасть можливість відчути курс інформатики, як єдине ціле, учні ють можливість майже безперервно працювати з комп'ютером, не відчуваючи нав'язливої ролі учителя. Такі форми роботи на уроці спонукають і вчителя підходити до більшості тем не формально, строго відповідно програмі, а із спрямуванням на наступне практичне застосування, все щойно вивчене при першій нагоді використовувати практично.

Постійно використовуючи ІКТ, від електронних дидактичних засобів, що носять локальний за змістом характер, можна безболісно переходити до великих за обсягом і змістом комплексних електронних дидактичних засобів (КЕДЗ). При творчому підході вони будуть ефективнішими за ліцензовані ПНЗ, маючи надзвичайно важливі переваги перед останніми: можливості конструювання, доповнення та удосконалення на будь-якому етапі.

Варто звернути увагу ще на одну важливу деталь. Справа в тому, що описані форми роботи на уроках інформатики реалізують безпаперову технологію навчання, коли учні не потребують ні підручника, ні зошита, а привчаються працювати лише з консоллю (екраном, клавіатурою та ручним маніпулятором). Така форма економічно доцільна і сприяє ефективнішому формуванню сучасної інформаційної культури та інформативної компетентності. Детальніше повернемося до неї нижче (у п.).

II.1 Електронні конспекти уроків.

Будемо розглядати *електронні конспекти уроків* (ЕКУ), як основну форму ЕДЗ на уроці. Тому детальніше зупинимось на *понятті* ЕКУ. Зміст значною мірою впливає з назви, тобто, перш за все це конспект уроку. Але якщо його не ототожнювати з терміном “поурочний план”, а мати на увазі достатньо деталізований опис відповідного темі навчального матеріалу з прикладами та методичними порадами (ремарками) щодо вивчення цього матеріалу, то такі конспекти уроків можна успішно використовувати і як фрагменти підручника для учнів, особливо в умовах недостатнього забезпечення учнів якісними підручниками, і як керівництво до дії для учителя. Необхідно лише “сховати” методичні ремарки. Кожен виготовлений засобами *Microsoft Office* друкований конспект уроку, подібно ЕСУ, повинен містити основну частину - ядро⁴. ЕКУ з допомогою локальної мережі слід розсилати на РМУ. Використання таких конспектів буде вирішувати одночасно кілька дидактичних завдань. По-перше, учні матимуть можливість читати, так би мовити, в оригіналі думки учителя, причому навіть під час підготовки домашнього завдання, якщо скопіювати конспект на ГМД. По-друге, працюючи над темою, кожен учень буде постійно комплексно застосовувати знання, уміння та практичні навички, одержані на уроках інформатики раніше, що буде для них додатковим чинником у формуванні сучасної інформаційної культури.

В електронних конспектах уроків широко застосовуються як графічні, так і анімаційні засоби та гіперпосилання. Це дає можливість роботу над навчальним текстом із статичного стану перевести в динамічний, що має дуже великі, раніше недоступні, дидактичні можливості.

Як уже згадувалось вище, на прикладі електронного конспекту уроку вводиться якісно нове поняття *комплексного електронного дидактичного засобу*.

⁴ На відміну від ЕСУ, ядром яких є навчальні презентації, ядро ЕКУ доцільно створювати у формі документа *Microsoft Word*, який найзручніше піддається прокрутці.

Для оцінювання якості ЕСУ та ЕКУ можна користуватись умовними критеріями. Спочатку наведемо зразок для ЕКУ⁵: таблиця II.1

№	критерій	оцінка в балах
1	Ступінь охоплення навчального матеріалу	__ балів
2	Логічність побудови та якість оформлення	__ балів
3	Відповідність загальним вимогам до конспекту уроку, лекції	__ балів
4	Відповідність загальним вимогам до електронних навчальних засобів	__ балів
5	Відокремлення навчальних матеріалів від методичних рекомендацій	__ балів
6	Ступінь структурованості	__ балів
6.1	Якість ядра	__ балів
6.2	Наявність додатків	__ балів
7	Навігаційні можливості, ознаки гіпертекстового документа	__ балів
7.1	Гіперпосилання на закладки	__ балів
7.2	Гіперпосилання на файли	__ балів
7.3	Гіперпосилання на інтернетресурси	__ балів
8	Можливості редагування	__ балів
9	Інтеграційні можливості з іншими ЕДЗ	__ балів
10	Можливості закріплення та тестування	__ балів
10.1	Наявність та наочність прикладів	__ балів
10.2	Наявність і форма запитань для закріплення	__ балів
10.3	Наявність завдань для закріплення (робочий бланк відповідей)	__ балів
	Загальна кількість балів	__ балів

Коротко прокоментуємо запропонований зразок критеріїв. Перш за все зауважимо, що вони необхідні тому, хто створює та використовує ЕКУ, адже і в процесі створення і для використання необхідно мати інструменти оцінювання. Більшість пунктів очевидні, тому прокоментуємо лише деякі нюанси оцінювання, які стосуються лише ЕКУ.

Перший пункт дає оцінку об'єму навчального матеріалу, представленою в ЕКУ. Він, як правило, повинен відповідати одному уроку, одній темі чи одному розділу. Але залежно від обсягу матеріалу, ЕКУ повинен мати чітке призначення: повторення, вивчення нового матеріалу, закріплення, узагальнення чи перевірки знань, наприклад, коли ЕКУ охоплює навчальний матеріал цілого розділу, то його тип може бути: урок повторення, урок узагальнення або урок перевірки знань. Як видно із наведеного зразка критеріїв, п.1 прямо пов'язаний з п. 3; 5; 6, тому виставлення балів у цих пунктах залежить від ступеня охоплення навчального матеріалу і оцінки п.1. Отже, необхідно, щоб ЕКУ охоплював максимально ефективний за обсягом навчальний матеріал.

Виходячи з того, що тут ідеться про план навчального заняття, пункти 2 та 3 необхідні лише для контролю відповідності ЕКУ загальним критеріям оцінювання уроків. Пункт 4 призначений для контролю відповідності загальноприйнятим порадам щодо розміщення матеріалу в електронних документах (презентації, публікації, текстові документи, *web*-сторінки тощо),

⁵ Кожен пункт та підпункти оцінюються по десятибальній шкалі. Загальна оцінка виставляється по десятибальній шкалі (10% балів від загального числа – 1 бал).

тобто зовнішнього вигляду сторінки чи кадру, палітри кольорів, розміру та гарнітури шрифту, кількості тексту і т.д.

З допомогою п. 5 контролюється чи відокремлені навчальні матеріали, призначені для учнів від специфічних, але не потрібних учням методичних порад та коментарів, які адресовані учителям та лекторам.

У пункті 6 мається на увазі кількість видів структурних компонентів ядра та додаткових електронних джерел. Може бути однорівнева (проста) структура, вона передбачає наявність пунктів плану уроку та підпунктів до них. Дворівневе або кількарівневе структурування передбачає наявність посилань на додатки в кінці документа або в інших файлах.

Деякі пункти, наприклад п. 7, мають свою структуру, бо є необхідність не тільки загальної оцінки, а й контролю за наявністю елементів. Навігаційні можливості – це послуги для користувачів, що дають змогу вільно пересуватись по ЕКУ, а основними засобами навігації в документах *Microsoft Word* є гіперпосилання. Слід зауважити, що ЕКУ, як і більшість інших типів ЕДЗ, немислимі без гіперпосилань та засобів навігації, причому для ЕКУ вони особливо важливі, адже тільки ядро може містити достатньо багато сторінок.

Залежно від мети конкретного ЕКУ можуть бути необхідними можливості санкціонованого (захищеного) доступу до редагування та узгодження для спільного використання з іншими ЕДЗ. Це передбачено п. 8-9.

Однією з найсуттєвіших ознак кожного ЕКУ є засоби закріплення та тестування, тому в критерії включено п.10 з окремими підпунктами. Приклади та запитання для закріплення в електронних документах можуть бути більш наочними та ефективними за рахунок включення елементів комп'ютерної графіки (п. 10.1-10.2). Під робочими бланками відповідей (п.10.3) маються на увазі документи, що містять індивідуальні завдання з розподілом на варіанти до конкретно взятого ЕКУ, які можуть мати вигляд окремої сторінки робочого електронного зошита. При створенні серії ЕКУ до окремої теми можна у вигляді додатку підготувати електронний робочий зошит, який можна буде фрагментарно використовувати протягом вивчення цілої теми, якій відповідає серія ЕКУ.

Об'єднавши серію ЕКУ та відповідний їй електронний робочий зошит, одержимо цілий електронний курс з окремо вибраної теми, який можна класифікувати як комплексний електронний дидактичний засіб. У якості зразка КЕДЗ під час другого заняття ОШППД розглядався прикладі дидактичної довідкової системи "Турбо Паскаль у таблицях".

II.2 Приклад ЕКУ.

(окремий файл № 1 - ядро) **ЛІНІЙНІ ПРОГРАМИ**

Тема, мета, обладнання, література 1

ПЛАН УРОКУ

I. Повторення та перевірка домашнього завдання

Дати відповіді на питання:

- 1) Що розуміють під поняттям структури?
- 2) Користуючись додатком 2 пояснити, яку структуру має програма на мовою Паскаль.
- 3) Що розуміють під оголошенням та описом величин?
- 4) Користуючись додатком 3 пояснити різні способи оголошення та опису величин.

II. Теоретичні відомості

Лінійними називають алгоритми та програми, якщо їх команди виконуються у порядку запису, тобто:

команда 1;	виконується першою
команда 2;	виконується другою
команда 3;	виконується третьою
...	
команда n-1;	виконується передостанньою
команда n;	виконується останньою

Звідси зрозуміло, що лінійні алгоритми та програми можуть містити лише команди вводу даних, присвоєння, виклику підпрограм 2 та виводу результатів 6 (див. зноску), тому для розгляду і написання лінійних алгоритмів і програм, крім знання структури 3 алгоритму та програми, правил оголошення та опису величин, 4 необхідно знати формат та правила виконання команд вводу 5, виводу 6 та присвоєння 7.

III. Приклад

```
program LineProgr; {заголовок програми}
uses Crt;          {блок опису модулів}
var a,b,p,s:real;  {блок оголошення і опису величин}
begin             {початок тіла програми}
  WriteLn('a?');ReadLn(a);    {блок вводу даних}
  WriteLn('b?');ReadLn(b);
  p:=2*(a+b);                {блок опрацювання інформації}
  s:=a*b;
  WriteLn( 'p=',p:6:2,'; s=',s:6:2); {блок виводу результатів}
```

⁶ Команди вводу та виводу розглядаються, як прості команди, але кожна з них по суті є допоміжною програмою (підпрограмою). Підпрограми поділяються на стандартні та створені користувачем. Запис команд вводу/виводу у програмі насправді є командами виклику відповідних підпрограм. Тому команду виклику 2 будь-якої підпрограми слід розглядати як просту команду.

end.

{кінець тіла програми}

IV. Практична робота

1) Користуючись гіперпосиланням

{запуск програми з допомогою транслятора PasABC⁰⁸}

2) Створити документ D:\klass\11_a\Group_1\Практична робота.doc згідно додатку 8, виконавши запропоновані завдання. ⁰⁹

завантажити та виконати зразок лінійної програми в середовищі PasABC покроково (F8) та прослідкувати послідовність виконання команд. Переконайтесь, що ця програма – лінійна.

V. Домашнє завдання

Повторити: Структуру програми мовою Паскаль;

Оголошення та опис величин у програмах мовою Паскаль.

Вивчити: Поняття лінійної програми;

Формат і особливості виконання команд вводу/виводу;

Формат і правило виконання команди присвоєння;

Практичне завдання : Написати програму визначення довжини кола і площі круга

за введеним з клавіатури значенням радіуса кола.

VI. Додатки:

(окремий файл № 2)

додаток 1 

Тема

- Лінійна програма та її структура;
- Способи вводу даних та виводу результатів, вікно *Crt*;
- Коментарі у програмі;
- Покрокове виконання програми;

Мета:

- Познайти учнів із прикладами лінійних програм, сформувані навички складання і запису лінійних програм, вставки коментарів у програмах, навички виконання та покрокового виконання програм;
- На прикладі модуля *Crt* познайти учнів із поняттям модуля.

Обладнання:

- Інтерпретатор мови програмування Паскаль *PascalABC⁷*;
- Програма для управління локальною мережею *NetOp School*, версія 5.0⁸;
- Таблиці для демонстрації структури програми, оголошення та опису величин і прикладів лінійної програми;

Література: І.Т. Зарецька, А.М. Гуржій, О.Ю. Соколов, Інформатика, частина 2, Київ, “Форум“, 2004р.

⁷ При вивченні теми “Алгоритмізація і програмування“ у базовому курсі основ інформатики використання одного із сучасних трансляторів мови програмування, створених під операційну систему *Windows*, наприклад *PascalABC*, значно ефективніше за використання сьогодні застарілих *Turbo Pascal* чи *Borland Pascal*.

⁸ На наше переконання сучасний урок основ інформатики в умовах практичного проведення кожного заняття.

(окремий файл 3)

додаток 2

СТРУКТУРА ПРОГРАМИ НА PasABC 3

```

[program ім'я;]
[uses імена модулів{через кому};]
[const ім'я1 = значення; ім'я2 = значення; ..]
[type опис нестандартних типів]
[var ім'я1 : тип; ім'я1 : тип;] {однотипні – через кому}
[блок опису підпрограми]
begin
  [тіло програми]
end.

```

блок оголошень та описів

текст, взятий в квадратні дужки, може бути відсутній

(окремий файл 4)

додаток 3

ОГОЛОШЕННЯ І ОПИС ВЕЛИЧИН 4

```

const n=5;
      M : array[1..n] of 0..9 = (8,2,1,0,7);

```

ТИП ЕЛЕМЕНТІВ МАСИВА - ДІАПАЗОН

```

type Mas = array[1..n,1..n] of real;

```

ТАК ОГОЛОШУЮТЬСЯ КІЛЬКА ОДНОТИПНИХ ВЕЛИЧИН

```

var a, b, c, max : real;
    st : string; d : byte;
    Mas1 : array[1..n,1..n] of integer;
    Mas2 : Mas;

```

ВЕЛИЧИНА Mas2
МАС ОГОЛОШЕНИЙ ТИП
array[1..n,1..n] of real;

(окремий файл № 5)

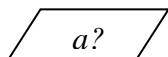
додаток 4

команда вводу 5:

Команда вводу в мові програмування Паскаль називається процедурою вводу.

Словесно блок схемою на алгоритмічній мові на мові паскаль

Ввести a



вести(a)

Read(a);

Read(a,b,...);

ReadLn(a);

ReadLn(a,b,...);

додаток 5

команда присвоєння 7:

Формат

правило виконання

Ім'я величини := вираз

Вираз записується за правилами лінійного запису, може містити змінні величини.

Команда виконується так:

- величинам у виразі надаються їх значення;

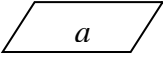
- визначається значення виразу;

- одержане значення надається величині, ім'я якої записано зліва від знака " := "

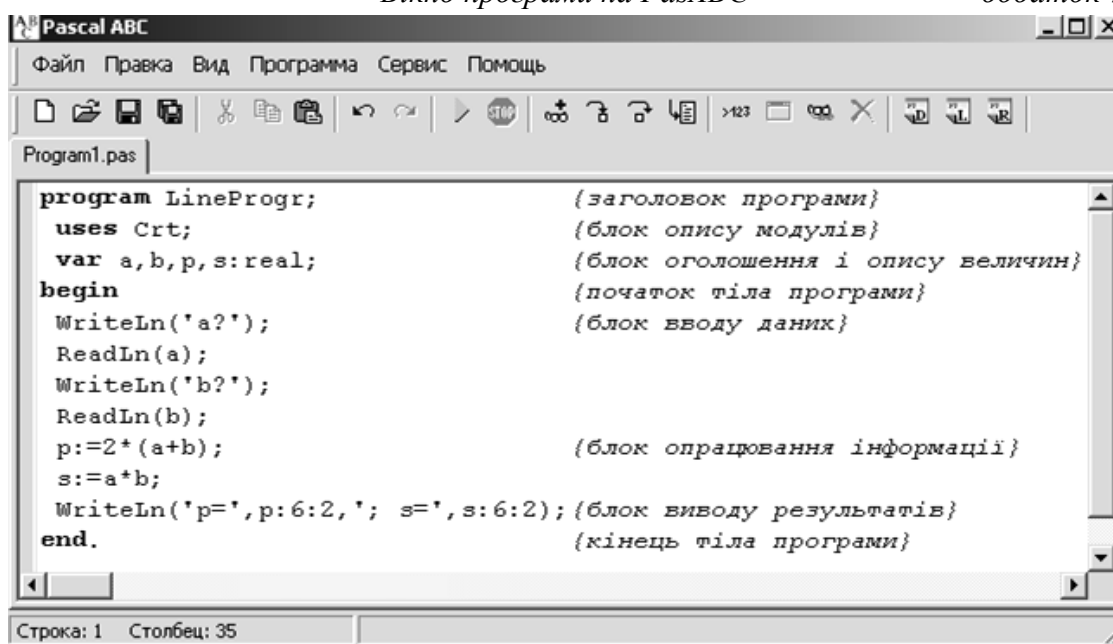
команда виводу 6:

Команда виводу в мові програмування Паскаль називається процедурою виводу.

Словесно блок схемою на алгоритмічній мові на мові паскаль

вивести *a*  вивести(*a*) `Write(a);`
`Write(a,b,...);`
`WriteLn(a);`
`WriteLn(a,b,...);`

Вікно програми на PasABC








(окремий файл № 6)

Завдання для практичної роботи:

№	I варіант	бали	II варіант	бали
1	Чи може програма мати вигляд: <code>begin end.</code> так/ні (залишіть правильну відповідь)	0/1	Чи може програма мати вигляд: <code>begin end;</code> так/ні (залишіть правильну відповідь)	0/1
2	обґрунтуйте відповідь:	0/1	обґрунтуйте відповідь:	0/1
3	Чи може бути у програмі команда: <code>begin end;</code> так/ні (залишіть правильну відповідь)	0/1	Чи може бути у програмі команда: <code>begin end.</code> так/ні (залишіть правильну відповідь)	0/1
4	обґрунтуйте відповідь:	0/1	обґрунтуйте відповідь:	0/1
5	Порівняйте записи: 1) <code>var a,b:integer;p,s:real;</code> 2) <code>var a,b,p,s:real;</code> Який з них правильний? 1/2 (залишіть правильну відповідь)	0/1	Порівняйте записи: 1) <code>var a;b;p;s:real;</code> 2) <code>var a,b:integer;p,s:real;</code> Який з них правильний? 1/2 (залишіть правильну відповідь)	0/1
6	обґрунтуйте відповідь:	0/1	обґрунтуйте відповідь:	0/1
7	Порівняйте записи: <code>(1+n)*n/2:=Sn;</code> та <code>Sn:=(1+n)*n/2;</code> Який з них не є командою? 1/2 (залишіть правильну відповідь)	0/1	Порівняйте записи: <code>Sn:=(1+n)*n/2;</code> та <code>Sn:=(1+n)*n/2</code> Який з них не є командою? 1/2 (залишіть правильну відповідь)	0/1
8	обґрунтуйте відповідь:	0/1	обґрунтуйте відповідь:	0/1
9	Яка з команд правильна: <code>Read(); Write;</code>	0/1	Яка з команд правильна: <code>Read; Write();</code>	0/1




	1/2 (залишіть правильну відповідь)		1/2 (залишіть правильну відповідь)	
10	обґрунтуйте відповідь:	0/1	обґрунтуйте відповідь:	0/1
11	Пояснити команду: <code>WriteLn(Sn);</code>	0/1	Пояснити команду: <code>WriteLn('Sn=',Sn);</code>	0/1
12	Пояснити команду: <code>Write('Sn=',Sn);</code>	0/1	Пояснити команду: <code>Write('Sn=',Sn:12:1);</code>	0/1
	Загальна кількість балів:	0/12	Загальна кількість балів:	0/12

Форма запису електронних конспектів уроку дещо незвична у порівнянні з традиційним виглядом поурочного плану, адже тут обов'язкові гіперпосилання, ілюстрації, додатки тощо. ЕКУ – це найчастіше пакет електронних документів різного формату, призначених для використання на уроці з допомогою локальної комп'ютерної мережі, причому, ці документи мають, так би мовити, різні області видимості. У наведеному вище прикладі, як уже зазначалось, додаток 1 призначений для викладача, тому його слід зробити невидимим для учнів. Подібно до цього, ЕКУ може містити цілий ряд адресованих тільки учителеві методичних ремарок.

Виходячи з потреби описати електронний дидактичний засіб, у прикладі з допомогою значків зразка ,  позначено гіперпосилання, які візуально такі ж, як і в справжньому ЕКУ⁹. Додатки, розміщені в окремих файлах, тут помічені таким чином: (окремий файл № 1). Особливо звертаємо увагу на гіперпосилання запуск програми з допомогою транслятора PasABC та відповідний йому додаток 7. Таким чином у прикладі зображено вікно системи програмування PasABC, що використовується для самостійного закріплення вивченого матеріалу. За реальних обставин гіперпосилання  дійсно запускає на виконання інтерпретатор PasABC, але, нажаль, учневі самостійно необхідно відкрити програму, яку ми бачимо в додатку 7. Так само, нажаль перехід до інтерпретатора PasABC не має зворотного зв'язку. Уникнути таких незручностей у принципі можна, застосувавши замість системи PasABC, яка є інтерпретатором, компілятор, зокрема Delphi. Тоді гіперпосилання  зверталось би до файлу формату *.exe. Але нагальної дидактичної потреби в цьому немає, тому не будемо концентруватись на тому, що зв'язок з додатком, який не входить до пакету Microsoft Office, однонапрямлений. Нижче, при описі ЕДЗ, будемо дотримуватись подібних позначень. Для практичного засвоєння радимо згідно наведеного опису створити ЕКУ і випробувати його в режимі виконання. Пропонуємо також оцінити ЕКУ "Лінійні програми" згідно наведених у таблиці 1 критеріїв.

II.3 Електронні сценарії уроків.

Описані нижче ЕСУ суттєво відрізняються від ЕКУ як за формою, так і за призначенням. Під **електронними сценаріями уроків** тут розуміються комп'ютерні презентації, але на відміну від традиційних презентацій - аудіовізуальних супроводів до окремих навчальних занять чи виховних заходів, вони є *сценаріями* для проведення. Використання традиційної презентації – це пасивне навчання, адже учні не мають можливостей для втручання в процес демонстрації, їх роль зводиться лише до спостереження. ЕСУ – це *алгоритм дій учнів* по вивченню оголошеної теми із

⁹ На відміну від природного для позначення гіперпосилання символа  позначення ,  вибрано через їх зрозумілий додатковий навігаційний зміст.

завданнями, запитаннями, ілюстраціями та прикладами. ЕСУ – це сценарій, тому всю заплановану на уроці роботу слід повністю відображена в ньому.

ЕСУ близький до електронного конспекту уроку, але, на відміну від нього, не містить блоків, призначених для учителя: запису мети, опису обладнання, теоретичних відомостей з методичними порадами, хронології дій учителя та учнів, які притаманні традиційним поурочним планам і їх електронним аналогам. Він може використовуватись паралельно з ЕКУ, але не обов'язково, адже включає в себе всі елементи ЕКУ, необхідні для роботи з учнями. Роль учителя на уроці з використанням ЕСУ зводиться до ролі арбітра, що займається хронометражем, оцінюванням, коментарями тощо.

Електронні сценарії уроків містять загальний план всього заняття, зміст його окремих структурних елементів, запитання та вправи для колективного і самостійного закріплення, домашні завдання, гіперпосилання на додаткові електронні джерела до даної теми. Щоб охопити тему, в ядро ЕСУ, створене з допомогою *Microsoft PowerPoint*, необхідно включити достатньо багато кадрів, часто із складною ієрархією. Цього можна досягти в одній презентації, де кожен структурний елемент становить один слайд, або в групі взаємопов'язаних презентацій. Але у будь-якому випадку елементи ЕСУ слід зв'язувати з допомогою кнопок переходу та гіперпосилань. Дуже корисними і ефективними можуть бути й перемикачі (тригери). Зрозуміло, що ядро ЕСУ створюється власними зусиллями учителя, можливо в кооперації з іншими учителями чи з учнями. У ядрі ЕСУ майже обов'язкові гіперпосилання на допоміжні ЕДЗ, створені з допомогою компонентів пакету *Microsoft Office* чи інших додатків *Windows* та автономні частини ліцензованих ППЗ НП, які можна імпортувати також як мультимедійні компоненти в окремі слайди. Проте, доцільніше їх підключати до ЕСУ з допомогою гіперпосилань, це допоможе уникнути переобтяження ядра, що часто приводить до аварійних зупинок, тобто "зависань". Пропонуємо скористатись порадою. Якщо презентацію зберегти у форматі *.pps (демонстрація), то це дасть суттєві переваги у її використанні. ЕСУ можна буде управляти без послуг *Microsoft PowerPoint*, як звичайним додатком *Windows*. ЕСУ слід поєднувати з використанням КЕДЗ і прикладних пакетів.

В зв'язку з тим, що учні переважну частину уроку з використанням ЕСУ індивідуально працюють з комп'ютером, він завжди має елементи застосування ІКТ, але здебільшого цього буває недостатньо, тому ЕСУ корисно застосовувати на бінарних уроках¹⁰.

Для оцінювання якості ЕСУ також необхідні певні критерії, тому наведемо та коротко прокоментуємо зразок: таблиця II.2

№	критерій	оцінка в балах
1	Відповідність темі уроку	___ балів
2	Відповідність загальним вимогам до електронних навчальних засобів	___ балів
3	Структура ядра (головної презентації)	___ балів

¹⁰ Різницю уроків з використанням ІКТ та бінарних уроків з інформатикою розглянемо нижче.

3.1	Якість головного меню (плану)	__ балів
3.2	Ступінь відображення плану	__ балів
4	Якість інтерфейсу (навігаційні можливості)	__ балів
4.1	Кнопки переходів	__ балів
4.2	Підказки	__ балів
5	Змістовність слайдів	__ балів
6	Якість унаочнення	__ балів
6.1	Якість графічних елементів	__ балів
6.2	Якість анімаційних елементів	__ балів
7	Інтеграційні можливості	__ балів
8	Коефіцієнт змістовності	__ балів
9	Можливості редагування	__ балів
10	Можливості закріплення	__ балів
	Загальна кількість балів	__ балів

Порівняння орієнтовних критеріїв оцінювання ЕКУ та ЕСУ приводить до висновків: по-перше, спостерігається зовнішня та деяка змістовна подібність, по-друге, існують суттєві й принципові відмінності в підході до оцінювання. Зовнішня, а особливо змістовна подібність пояснюється тим, що обидві форми ЕДЗ – це, як уже зазначалось вище, засоби ІКТ, тобто стосуються форми представлення та вивчення навчального матеріалу. Відмінності випливають із принципової відмінності ЕКУ та ЕСУ (нагадаємо, ЕКУ – це по'єднання методичних поміток для викладача та стислого конспекту для учнів, а ЕСУ – це спільна для учителя та учнів алгоритмічна схема заняття).

При оцінці ЕСУ перш за все слід виявити відповідність темі уроку (п.1) та загальним вимогам до електронних навчальних засобів (п.2). Проте, на відміну від ЕКУ може охоплювати лише один урок, або одну вузьку тему, що й вимагається в п.1.

Оцінка якості структури ядра, тобто головної презентації (п.3), дуже важлива, вона функціональність та якість головного меню, а також якість та ступінь наповнення ядра відповідно плану.

ЕКУ - переважно цілісні текстові документи, вони не мають інтерфейсу, навігаційні можливості здійснюються практично виключно з допомогою гіперпосилань. Для ЕСУ якість інтерфейсу має суттєве значення, адже відповідно до специфіки він потребує більш чітко виражених і надійних навігаційних можливостей. Тому п.4 включає аналіз системи кнопок переходів та впливаючих підказок.

Якість будь-якої презентації залежить від якості окремих слайдів. Для ЕСУ ця обставина особливо важлива, адже ми маємо справу з презентацією для індивідуальної, а не колективної роботи. Звідси впливає необхідність послайдового та покадрового¹¹ контролю за змістовністю, наочністю та естетичністю ілюстративного матеріалу (п.5-6).

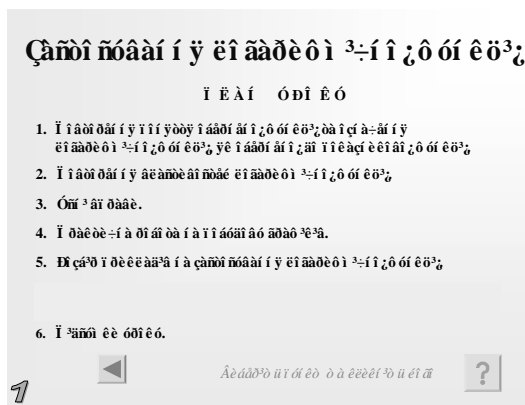
Пункти п.7, 9-10, які присутні й серед критеріїв оцінювання ЕКУ, але неважко зрозуміти, що при оцінюванні ЕСУ вони мають трохи змінені акценти. Наприклад, п.10 не потребує такої деталізації, як це видно для випадку ЕКУ.

¹¹ Під кадрами будемо розуміти модифікації слайдів, досягнену з допомогою анімаційних ефектів.

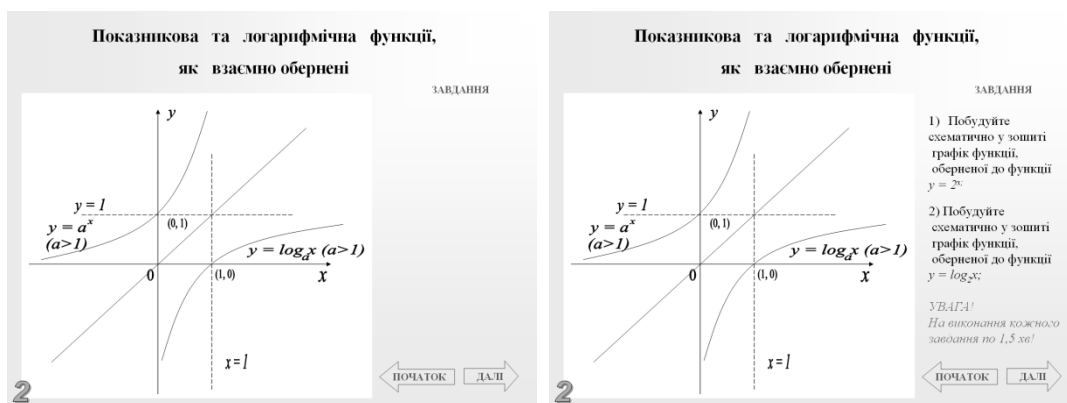
Особливо слід вказати на п.8. Він введений для контролю за тим, щоб досягалось оптимальне співвідношення між кількістю слайдів

II.4 Приклад електронного сценарію уроку.

Особливістю уроків з ЕСУ є універсальний характер по відношенню до навчальних предметів. У зв'язку з цим за приклад візьмемо урок алгебри на тему "Застосування логарифмічної функції"¹². Спочатку опишемо ядро. Кожен слайд пронумеровано в нижньому лівому кутку. Отже ядро складається з 8 слайдів, перший з них містить головне меню. Навігація по ядру реалізована через гіперпосилання на слайди, про що свідчить статична підказка "Виберіть пункт та клацніть його" та стандартні кнопки НАЗАД і ДОВІДКА.



Як видно з ілюстрації до другого слайду він містить два кадри, причому, перший кадр (зліва) треба вважати базовим, а другий похідним, бо він (справа) одержується з першого додаванням завдань та підказки для їх виконання.



Третій слайд, крім базового, містить цілу ієрархію похідних, з них вісім лише є похідними першого рівня (бо одержуються з базового одним доповненням). Створення таких кадрів досягається з допомогою вставки анімаційних ефектів та перемикачів. Доцільність таких дій очевидна, адже без них довелось би створювати надто багато слайдів, що привело б до надмірності розміру файлу, який містить презентацію, а їх лінійна демонстрація мала б дуже малу дидактичну цінність.

¹² Без ілюстрації більшості слайдів важко стисло описати ЕСУ в цілому, але умови друку дозволяють їх представляти у чорно-білому варіанті.

**В Л А С Т И В О С Т І
ЛОГАРИФМІЧНОЇ ФУНКЦІЇ**

$y = \log_a x \quad (a > 1)$
 $y > 0 \quad y < 0$
 $y = \log_a x \quad (0 < a < 1)$
 $y > 0 \quad y < 0$
 $D(x) = (0; +\infty)$
 $E(x) = R$
 нулі ← ПОЧАТОК → ДАЛІ

**В Л А С Т И В О С Т І
ЛОГАРИФМІЧНОЇ ФУНКЦІЇ**

$y = \log_a x \quad (a > 1)$
 $y > 0 \quad y < 0$
 $y = \log_a x \quad (0 < a < 1)$
 $y > 0 \quad y < 0$
 $D(x) = (0; +\infty)$
 $E(x) = R$
 нулі ← ПОЧАТОК → ДАЛІ

Багаторівневу ієрархію слайду легше проілюструвати на "біднішому" анімаційними ефектами та навігаційними можливостями четвертому слайді, бо він всього три похідні кадри.

УСНІ ВПРАВИ

Що більше? Знайти область визначення:

1) $\log_2 2$ чи 0?
 2) $\log_3 \frac{1}{3}$ чи 0?
 3) $\log_4 4$ чи 1?
 4) $\log_2 3$ чи $\log_2 5$?
 5) $\log_2 7$ чи $\log_2 7$?
 6) $\log_2 7$ чи $\log_2 \frac{1}{9}$?
 7) $\log_2 3$ чи $\log_2 \frac{1}{3}$?

← ПОЧАТОК → ДАЛІ

УСНІ ВПРАВИ

Що більше? Знайти область визначення:

1) $\log_2 2$ чи 0?
 2) $\log_3 \frac{1}{3}$ чи 0?
 3) $\log_4 4$ чи 1?
 4) $\log_2 3$ чи $\log_2 5$?
 5) $\log_2 7$ чи $\log_2 7$?
 6) $\log_2 7$ чи $\log_2 \frac{1}{9}$?
 7) $\log_2 3$ чи $\log_2 \frac{1}{3}$?

← ПОЧАТОК → ДАЛІ

УСНІ ВПРАВИ

Що більше? Знайти область визначення:

1) $y = \log_2 x$;
 2) $y = \log_2(x+1)$;
 3) $y = \log_2(3-x) + \log_2(3+x)$;
 4) $y = \log_2|x|$;
 5) $y = \log_2(-x)$;
 6) $y = \log_2(x^2 - x + 6)$.

← ПОЧАТОК → ДАЛІ

УСНІ ВПРАВИ

Що більше? Знайти область визначення:

1) $y = \log_2 x$;
 2) $y = \log_2(x+1)$;
 3) $y = \log_2(3-x) + \log_2(3+x)$;
 4) $y = \log_2|x|$;
 5) $y = \log_2(-x)$;
 6) $y = \log_2(x^2 - x + 6)$.

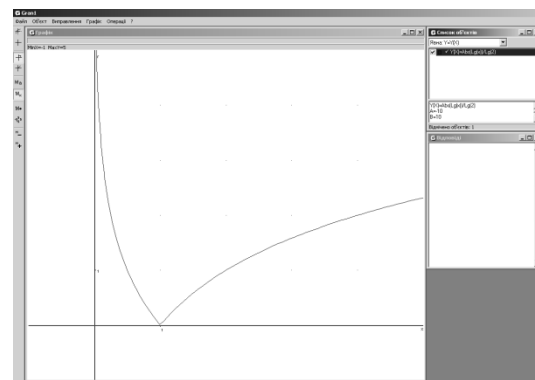
← ПОЧАТОК → ДАЛІ

Кожен, хоч трохи обізнаний з технологією роботи з *Microsoft Power Point*, знає, що можна досягти практично будь-якої послідовності демонстрації слайдів та кадрів. Такі широкі навігаційні можливості пакету створюють надзвичайно потужні й цінні дидактичні можливості.

Побудувати ескізи графіків функцій у зошиті та зробити перевірку з допомогою GRAN1:

а) $y = \log_2|x|$;
 б) $y = \log_{\frac{1}{2}}|x-1|$;
 в) $y = \log_3(3-x)$.

Gran1
← ПОЧАТОК → ДАЛІ



Окрему роль має п'ятий слайд, що містить єдиний кадр. Він призначений для переходу в автономний тренувальний режим (схематична побудова графіків функцій у робочих зошитах та перехресна перевірка з допомогою пакета *Gran1*).

Для скорочення опису всього ядра далі наведемо лише базовий та вихідний (остаточний) кадри наступних слайдів (№ 6-8), сподіваючись, що уважний читач зможе уявити досить точно все ядро ЕСУ.

5.1 Вказати множину точок на площині, координати яких пов'язані співвідношенням $\log_y x < 1$

$x > 0, y > 0, y \neq 1$
 $0 < y < 1$
 $y < x$
 $y > 1$
 $y > x$

ПОЧАТОК ДАЛІ

5.1 Вказати множину точок на площині, координати яких пов'язані співвідношенням $\log_y x < 1$

$x > 0, y > 0, y \neq 1$
 $0 < y < 1$
 $y < x$
 $y > 1$
 $y > x$

ПОЧАТОК ДАЛІ

5.2 Зобразити на площині всі точки (x, y) , для яких виконується рівність: $\log_{x-y}(x^2+y^2) \geq \log_{x-y} 4$

$x - y > 1$
 $x^2 + y^2 \geq 4$
 $0 < x - y < 1$
 $0 < x^2 + y^2 \leq 4$

ПОЧАТОК ДАЛІ

5.2 Зобразити на площині всі точки (x, y) , для яких виконується рівність: $\log_{x-y}(x^2+y^2) \geq \log_{x-y} 4$

$x - y > 1$
 $x^2 + y^2 \geq 4$
 $0 < x - y < 1$
 $0 < x^2 + y^2 \leq 4$

ПОЧАТОК ДАЛІ

5.3 Вказати множину точок на площині, координати яких пов'язані співвідношенням $\log_{|y|} |x| > 0$

1) якщо $0 < |y| < 1$,
 то $0 < |x| < |y|$
 2) якщо $|y| > 1$,
 то $|x| > |y|$

ПОЧАТОК

5.3 Вказати множину точок на площині, координати яких пов'язані співвідношенням $\log_{|y|} |x| > 0$

1) якщо $0 < |y| < 1$,
 то $0 < |x| < |y|$
 2) якщо $|y| > 1$,
 то $|x| > |y|$

ПОЧАТОК

Для слайдів 1 та 6-8 пропонуємо самостійно провести аналіз навігаційних та дидактичних можливостей, як це було зроблено вище для слайду 4.

Залишається нагадати, що при потребі легко провести необхідне редагування або змінити інтеграційні зв'язки з іншими додатками з допомогою гіперпосилань чи кнопок переходів, як це було видно у слайді 5 відносно додатка *Gran1*. Для цього необхідно відкрити файл, що містить ядро, у форматі *.ppt.

Пропонуємо оцінити описаний приклад ЕСУ з допомогою наведених критеріїв оцінювання.

III. Застосування методу проектів у процесі викладання основ інформатики.

Вище було сформульовано роль і місце інноваційних технологій навчання в сучасній освіті. Проте ще зовсім мало згадувались інтерактивні методи навчання, які мають пряме відношення до ІТН.

Термін “інтерактивний” має англійське походження і означає “взаємодіючий”. Існують різні підходи до визначення інтерактивного навчання, зокрема, його часто трактують, як діалогове навчання: “Інтерактивний – означає здатність взаємодіяти чи знаходитись в режимі бесіди, діалогу з чим-небудь (наприклад, комп’ютером) або ким-небудь (людиною). Отже, інтерактивне навчання – це перш за все діалогове навчання, в ході якого здійснюється взаємодія вчителя та учня” [13]. За О.Пометун та Л.Пироженко “Сутність інтерактивного навчання полягає в тому, що навчальний процес відбувається за умов постійної, активної взаємодії всіх учнів. Це *співнавчання, взаємонавчання* (колективне, групове навчання в співпраці)...”[14].

Розшифровуючи поняття “інтерактивні методи” буквально, можна погодитись, що їх зміст походить від двох складових: “інтер” та “активний”, тобто означає методи групової активізації іззовні. Звідси випливає справжня, а формальна сутність інтерактивних методів навчання. І вона означає колективне добування знань через зовнішню керованість, тобто через маніпуляцію свідомістю об’єкта навчання.

Тому інтерактивні технології не можуть бути самоціллю [16, 17]. Потрібно постійно контролювати процес досягнення поставлених цілей (вони повинні бути чітко сформульовані і легко контрольовані), у випадку невдачі переглядати стратегію і тактику роботи, шукати і виправляти недоліки. *Урок не повинен бути перевантаженим інтерактивною роботою.* Оптимально рекомендується застосовувати не більше 1-2 методів за урок. Слід поєднувати взаємонавчання з іншими методами роботи, зокрема з традиційними методами та *самостійним пошуком*. Неможливо побудувати весь процес навчання виключно на інтерактивних методах. Це один з багатьох прийомів, які допомагають досягнути мети і приносять результат тільки в поєднанні з іншими.

Одним із найвідоміших та найефективніших інтерактивних методів вважається метод проектів. Не слід плутати *метод проектів* - як форму організації навчально-виховного процесу та *проект* - як етап розв’язання проблеми. Сьогодні більше акцентують увагу саме на методі проектів, маючи на увазі одну із форм інтерактивних технологій. Але – це класичний приклад того, як за лісом не бачать дерев, або, захопившись вихованням, гублять його об’єкт – дитину. Отже, без правильного розуміння поняття “проект” не можна організувати ні проектну діяльність, ні застосувати метод проектів.

Перед тим, як сконцентруватись на поняттях “проект” та “проектна діяльність” корисно згадати, що існує класична, але по суті абсурдна класифікація видів навчання. Говорять, що існує *пасивне, активне та інтерактивне* навчання. З цим ніяк не можна погодитись. Хіба не безглуздо стверджувати, що існує пасивне навчання? *Навчання – це процес, в якому*

одночасно і активно приймають участь або дві особи – учитель і учень, або одна особа - учень. Якщо учень пасивний (читай - відсутній), то це зовсім не навчання, причому, можна сказати “за визначенням“. Отже, **будь-яке навчання активне**. Щодо інтерактивного навчання можна сказати наступне. Інтерактивне – значить стимульоване із зовні, тобто навчання під впливом зовнішніх стимуляторів. Тому інтерактивні елементи в навчально-виховному процесі корисно застосовувати *не постійно, а епізодично*, наприклад, для актуалізації, концентрації, закріплення чи в процесі повторення та узагальнення. Слід завжди мати на увазі, що *інтерактивні методи – це окремий вид маніпуляцій свідомістю*, їх надмірне “вживання“ може привести до формування в учнів ефекту (а то й дефекту) залежності, тобто нездатності самостійно мислити, виконувати роботу, приймати рішення.

Важко погодитись, що широко відома програма “Інтел[®] Навчання для майбутнього“ правильно орієнтує вчителів щодо підвищення якості навчально-виховного процесу. Основна вада цієї програми в тому, що під керівництвом учителя учні повинні вибирати так звані *ключове та тематичні питання*, що стосуються проблеми, яку *тільки збираються досліджувати*. Не знаючи предмету дослідження, учні не можуть свідомо, без допомоги учителя, вибирати жодних конкретних питань. Проект по своїй суті – це алгоритм розв’язування поставленої задачі, а не сукупність прийомів досягнення результатів. Ми часто спостерігаємо, як при формулюванні алгоритма чи опису програми учні концентрують увагу не на *визначенні суми, пошуці, сортуванні, переборі* тощо, а говорять про *цикли, масиви* і т.д. Цикли – це лише інструменти, що можуть допомогти знайти суму, виконати пошук, сортування, перебір і т.д., а масиви – це форма представлення даних. Самі ж питання, як *тематичні, так і ключові*, можуть виникати в процесі розвитку (реалізації) проекту. Ніякого відношення до проекту не мають також різноманітні інтерактивні прийоми, адже, як зазначалось вище, їх суть в маніпулюванні свідомістю учнів, тоді, як проектна діяльність повинна бути свідомим вибором кожного учня.

Спробуємо дати відповідь на питання: *чи можливо навчити?* Відповідь: неможливо, *можливо лише навчитись*. Тоді виникне питання: Як *учитись і навчитись?* Відповідь: *учитись – це самостійно шукати запитання і відповіді на них*. З огляду на це можна говорити про навчально-пізнавальний проект, який має на меті освіту. Він полягає в організації навчальної діяльності, за принципом: *від керованого і контрольованого до самостійного навчання*.

Проект - в буквальному значенні - це „кинутий вперед“, тобто прототип якогось виду діяльності. Виконавець будь-якого завдання, якщо він достатньо організований, весь час процює з проектом - спочатку планує (*проектуює*) майбутню роботу, потім поетапно *виконує проект*. Отже проект може бути в двох стадіях: стадії *складання* і стадії *виконання*. Зрозуміло, що спочатку слід навчитись виконувати проекти, запропоновані кимось іншим. У цьому випадку складання проекту і контроль за його виконанням покладається на *керівника проекту*. Як правило, до цього зводяться традиційні форми навчальної роботи.

У педагогічній літературі наводиться кілька типів проектів, що використовуються у шкільному навчанні.

1. *Дослідницькі проекти*, які цілком підпорядковані логіці дослідження і мають структуру, наближену або повністю відповідну справжньому науковому дослідженню (аргументація, актуальність теми, визначення проблеми, предмета і об'єкта дослідження, окреслення завдань, методів дослідження, джерел інформації, висування гіпотез, означення шляхів її розв'язання, обговорення отриманих результатів, їх оформлення).

2. *Творчі проекти*, які не мають детально опрацьованої структури діяльності учасників. Вона лише накреслюється і розвивається відповідно до жанру і форми кінцевого результату (спільна газета, твір, відеофільм, драматизація тощо.)

3. *Пригодницькі та ігрові проекти*, які також не мають чіткої структури. Вона визначається під час опрацювання учасниками проекту ролей, зумовлених характером і змістом проекту (історичні особи, вигадані герої, які імітують офіційні та ділові відносини).

4. *Інформаційні проекти*, які зорієнтовані на збирання інформації, мають чітку структуру (мета, актуальність, методи отримання і обробки інформації, оформлення результатів та їх презентація).

5. *Практично-організаційні проекти*, спрямовані на вироблення конкретної програми дій, методичних рекомендацій.

Методика виконання проектів передбачає:

- Вибір теми або проблеми. Теми може запропонувати вчитель або ж їх вибирає клас безпосередньо із тих пропозицій, які були запропоновані під час „мозкової атаки”.
- Планування проекту. Вчитель і учні узгоджують всі складові процесу виконання проекту (на який час він розрахований, які ресурси будуть використані, яким чином учні працюватимуть (групою чи поодиночці)).
- Процес дослідження / дії.
- Результати і / чи висновки.
- Представлення результату роботи.
- Оцінка результатів і процесу.

Проект може бути монопредметним, міжпредметним і надпредметним. Виконання проектів має різні етапи, однак на всіх етапах необхідно наголошувати, що відповідальність учнів за їхнє навчання лежить на них самих.

Це дуже ефективний спосіб співпраці у рамках узгодженої теми чи питання. Він вчить окреслювати цілі, планувати (структурувати) роботу, оцінювати результати. Передбачає високу активність і незалежність виконавців проекту.

Процес навчання має різні рівні, найвищим з яких є *самостійне навчання*. Для цього рівня необхідні достатні навички як виконання проектів, так і їх складання, тобто *проекткування*. При самостійному навчанні метод проектів має зовсім інший зміст, ніж при застосуванні інтерактивних технологій.

Третє заняття ОШППД (2008 р.) було присвячене застосуванню методу проектів. Розглядалися два варіанти застосування цього методу: для підготовки та проведення уроків-семінарів та при організації державної підсумкової атестації у формі захисту навчальних проектів.

Одна із ефективних структурних схем семінарської роботи така:

1. План підготовки до семінару;
2. План проведення семінару.

На першому етапі слід встановити тему та ключове питання семінару, терміни підготовки, розподілити теми та тематичні питання для учасників, питання для всіх учасників, пов'язані з теоретичними положеннями вибраної теми, форми підготовки питань (реферати, виступи, повідомлення тощо), форми роботи над питаннями (індивідуальні, групові, колективні).

На другому етапі слід визначити послідовність проведення семінару, вибрати спікерів, призначити доповідачів, асистентів, опонентів, встановити регламент, систему балів для оцінювання, форми опитування та підведення підсумків.

Подібну схему може мати й захисту проектів під час державної підсумкової атестації з інформатики. Різниця в тому, що тут слід строго дотримуватись відповідної інструкції МОН України. Але процес підготовки державної підсумкової атестації з інформатики у формі захисту проектів має чисто індивідуальний характер, учень працює самостійно, але під наглядом учителя.

Вище було достатньо переконливо обґрунтовано, що групові форми роботи можуть використовуватись лише епізодично, *найефективнішими є самостійні форми навчальної діяльності*. При будь-якій нагоді їм слід надавати перевагу, а випускники загальноосвітньої школи повинні бути повністю готовими добувати знання та інтелектуально удосконалюватись самостійно.

Під час одного із занять ОШППД слухачі спостерігали, як учень працював над проектом, який можна назвати "Розв'язування олімпіадної задачі з програмування". Він складав загальний план: оголошував та описував відповідно до умови задачі дані й шукані величини, обдумував математичну модель задачі, описував, уточнював, доводив та оцінював алгоритм, писав та відлагоджував програму, для чого придумував відповідні умові задачі тести, нарешті виконував програму з метою одержання результатів. Робота над задачею з програмування – це чудовий зразок *самостійної проектної діяльності*. Обґрунтування такого погляду постійно зустрічається в [7]-[12].

IV. Вивчення алгоритмізації та програмування у формі експрес-курсу.

Серед шести актуальних проблем сучасної шкільної інформатики перші дві вказують на труднощі, що виникають при вивченні теми "Комп'ютерне моделювання. Основи алгоритмізації та програмування". Враховуючи, те, що логічно ця тема стоїть окремо і залишається однією з ключових у всьому курсі, виникла необхідність окремо виділити і деталізувати проблеми її вивчення [6].

IV.1 Проблеми вивчення теми “Комп’ютерне моделювання. Основи алгоритмізації та програмування” у школі.

Цілком слушно вважати названу тему *розділом*, адже протягом багатьох років весь курс інформатики майже виключно зводився до вивчення алгоритмів і програм. Крім того, *темою* прийнято називати частину деякого розділу певного предмету чи курсу, тоді, як “Алгоритмізація і програмування” логічно не належить до жодної з частин курсу інформатики, які прийнято скорочено називати АС, СПЗ чи ППЗ¹³ тощо, а враховуючи логічну відособленість від решти навчального матеріалу шкільної інформатики, цю тему доцільно вважати навіть *окремим курсом*. Щоправда, курс, як правило, має значно більшу кількість навчальних годин.

Вже в назві цього розділу чи курсу, що поєднує терміни *комп’ютерне моделювання, основи, алгоритмізація та програмування* натякається на стислість і оглядовість. Це цілком підтверджується відведеною у програмі кількістю навчальних годин (12). Якщо врахувати, що у переважній більшості загальноосвітніх навчальних закладів інформатика вивчається у обсязі універсального профілю, при його викладанні практично перед усіма вчителями виникає багато особливо гострих питань і проблем, які дуже важко вирішити. Розглянемо деякі з них, що, на наш погляд, цілком доречно поставити:

- Який сенс вивчати у школі такий досить абстрактний і формалізований розділ *оглядово*?
- Як “*втиснути*” у 12 навчальних годин названий розділ?
- *Чи можливо* при цьому *досягти* поставленої мети та засвоєння учнями вказаного рівня теоретичних знань і практичних навичок?

Виходячи з того, що учителі не в змозі змінювати програму, ці питання можна вважати риторичними. Але тоді проблема, пов’язана з ними залишиться. Її можна сформулювати так: ***Проблема забезпечення необхідного рівня теоретичних знань і практичних навичок при викладанні розділу курсу основ інформатики “Комп’ютерне моделювання. Основи алгоритмізації та програмування” згідно діючої програми в умовах недостатньої кількості відведених навчальних годин.***

IV.2 Методичне обґрунтування експрес-курсу алгоритмізації та програмування

Спробуємо вказати на досить, на нашу думку, ефективний шлях вирішення поставленої проблеми. Безперечно, її розв’язати можна лише шляхом *максимального ущільнення* навчального матеріалу, *універсалізації* задач, запропонованих для його закріплення, *забезпечення логічної єдності* всього навчального матеріалу та при умові *обов’язкового виконання* учнями *теоретичних і практичних завдань у позаурочний час*. Крім того, практично неможливо обійтись без збільшення кількості відведених для вивчення розділу

¹³ Маються на увазі частини обчислювальної системи: АС – апаратна складова, СПЗ – системне програмне забезпечення, ППЗ – прикладне програмне забезпечення.

навчальних годин. Їх можна взяти за рахунок ущільнення інших тем курсу, яке, виходячи з місцевих умов, допускається авторами програми¹⁴.

Зрозуміло, що при застосуванні до розділу таких умов його справді можна називати окремим *експрес-курсом з основ алгоритмізації та програмування* (ЕКАП) для загальноосвітніх шкіл згідно діючих навчальних планів та програм для універсального профілю¹⁵. Термін “експрес” тут вжито як в традиційному розумінні, тобто “прискорений”, адже програмування за 12-15 навчальних годин – це справді надзвичайно швидко і стисло, так і в розумінні “особливий”, “надзвичайний” та “ефективний”.

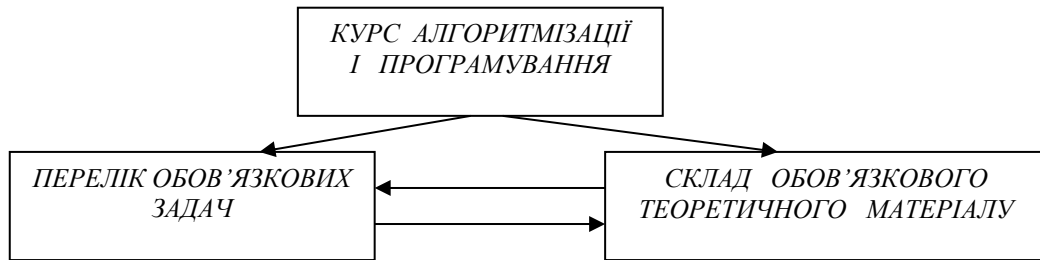
Вважаємо за необхідне наголосити, що нижче буде описано не стислий курс алгоритмізації та програмування для учнів загальноосвітніх шкіл, а методичні поради щодо вивчення окремої теми шкільної інформатики у формі ЕКАПу. Кожен учитель при потребі та бажанні матиме можливість, врахувавши наші методичні поради, самостійно створити курс лекцій і практичних занять, більш чи менш стислий і компактний.

Створити такий ЕКАП не можливо без застосування певних *робочих ідей*, або *принципів*, які покликані допомогти у досягненні ущільнення, універсалізації, логічної єдності і т.д. Тому дуже доречно згадати думку Р.Декарта: “Кожна розв’язана мною задача ставала образом, який застосовувався надалі при розв’язуванні інших задач”. Вона висловлює загальновідому, навіть банальну в наш час дидактичну істину, але все геніальне справді просте. Дійсно, курс обов’язково передбачає розгляд певної кількості задач, розв’язання кожної з яких перш за все потребує математичної постановки. Учням, що в процесі вивчення розділу алгоритмізації та програмування ще тільки знайомляться з новими різновидами мислення, зокрема *алгоритмічним* та *структурним*, при цьому буде необхідно не тільки багато додаткового часу, а й значних інтелектуальних зусиль, спрямованих на подолання психологічних ефектів постійної новизни і невідомості та розпорошення уваги, яка приводить до втрати почуття цільності курсу. Тому ми будемо намагатись розглядати таку *серію задач, де кожна наступна - в певному розумінні буде розширенням чи аналогією попередньої*, іншими словами, з метою економії часу та інтелектуальної напруги спробуємо реалізувати ідею *викладу курсу програмування на одній великій задачі*, або за **принципом нитки** (умовно позначимо його літерою А).

Але будь-який курс повинен охоплювати певне завершене коло основних задач, матимемо це на увазі й при створенні нашого ЕКАПу. Перелік задач повинен відповідати діючій програмі. Крім того, виконуючи її, ми повинні забезпечити необхідний рівень засвоєння мінімального обсягу теоретичного матеріалу, тобто необхідно скласти взаємопов’язані *повний перелік обов’язкових задач і мінімальний обсяг теоретичного матеріалу* (схемаIV.1).

¹⁴ Рекомендуємо доповнити вивчення теми хоча б трьома додатковими навчальними годинами, які слід поставити в кінці кожного з трьох блоків уроків експрес-курсу.

¹⁵ Описані тут ідеї та методи можна адаптувати і до інших варіантів діючих програм з інформатики для загальноосвітніх навчальних закладів.



Наприклад, треба розглянути задачі:

список 1

номер та назва обов'язкові задачі	номер відповідного задачі теоретичного матеріалу
1 Написати лінійну програму обчислювального характеру	1,2,3,4,6
2 Написати діалогову програму	1,2,3,4,5,6
3 Написати програму із розгалуженням	1,2,3,4,5,6,8
4 Написати циклічну програму	1,2,3,4,5,6,7
5 Написати програму визначення суми заданих формулою чисел	1,2,3,4,5,6,7
6 Написати програму визначення суми даних чисел, заданих масивом	1,2,3,4,5,6,7,9,10
7 Написати програму пошуку в лінійних неупорядкованих числових масивах	1,2,3,4,5,6,7,8,9,10)
8 Написати програму пошуку в лінійних упорядкованих числових масивах	1,2,3,4,5,6,7,8,9,10)
9 Написати програму сортування масивів	1,2,3,4,5,6,7,8,9,10,11

У якості відповідного цим задачам обов'язкового теоретичного матеріалу можна взяти теми:

список 2

1. Поняття величини в інформатиці, оголошення і опис величин;
2. Математичні операції та функції над числовими типами та запис математичних виразів у лінійній формі;
3. Команда присвоєння;
4. Процедури вводу/виводу;
5. Символьний та рядковий типи;
6. Коментарі в програмі;
7. Поняття циклу;
8. Поняття команди розгалуження;
9. Поняття масиву;
10. Ввод масиву;
11. Вивод масиву.

Як бачимо, пункти першого списку повинні бути узгоджені при вивченні із деякими пунктами другого списку, причому, з кожним наступним ідуть повтори та наростання обсягу обов'язкового теоретичного матеріалу (у першому списку відповідні пункти другого списку наведені справа в дужках). Таким чином, що цілком природньо, застосування однієї і тієї ж порції теоретичного матеріалу при розгляді різних задач часто повторюється у різних

комбінаціях з іншими порціями, а не рідко дублюється (звернемо увагу хоча б на порції теоретичного матеріалу 1-4, які дублюються в усіх дев'яти наведених задачах). Skorиставшись щойно поміченим, можна досягти значного ущільнення як навчального матеріалу так і часу, відведеного для вивчення. Як видно із списку 1, порції теоретичного матеріалу з меншими номерами на кожному наступному кроці викреслюються, наприклад, починаючи із задачі 3, викреслені п.1-4, починаючи із задачі 7 викреслені п.1-7 і т.д. Це слід розуміти так: викреслені у списку 1 порції теоретичного матеріалу з моменту викреслення більше не вивчаються і не повторюються, наприклад, перед розглядом задачі 8, можна обійтись вивченням тільки двох порцій теоретичного матеріалу: поняття команди розгалуження (№8) та ввід масиву (№10), а з рештою тем (№№1-7 та №9) ознайомлення відбулось при розв'язуванні попередніх семи задач.

Але чи можна обійтись у цій ситуації навіть без повторення тем 1-7 та 9? Відповімо: *не тільки можна, а при вивченні розділу у вигляді ЕКАПу і потрібно*. Тут ми свідомо йдемо проти вікової дидактичної, але, на жаль, в дійсності *не зовсім педагогічної традиції* “повторення – мати навчання”. Чому практично на всіх уроках організовується повторення попереднього навчального матеріалу? Щоб створити максимум комфорту для учнів. Але погодимось, що безкінечне повторення прописних істин – це зовсім не цікаво для тих учнів, які давно засвоїли їх на пам'ять. Скільки таких учнів у класі з n осіб? Якщо цю величину позначити k , то ось строга відповідь: залежно від ситуації $0 \leq k \leq n$. Напрошується цікавий висновок: *у навчальному процесі не доцільно зловживати повторенням раніше вивченого, інакше в учнів можуть виробитись неухважність, байдужість, бездіяльність, безініціативність і т.д., замість повсякденного планування повторення слід постійно слідкувати, щоб k було якомога ближче до n , а на деякому етапі навчання досягало його*.

Справді, чи завжди у повсякденному житті починають роботу з повторення, тренування тощо? Це доцільно, наприклад, спортсменам, співакам тощо, для яких так звана розминка – необхідність. Для більшості ж видів людської діяльності *основні прийоми та навички повинні бути доведені до автоматизму*. Щоб переконатись у цьому, візьмемо будь-яку масову професію, хоча б столяра. Хіба він, починаючи робочий день, повторює навички пиляння, стругання, забивання цвяхів і т.д.? Отже, в ідеалі процес навчання повинен забезпечувати на будь-якому етапі абсолютне засвоєння необхідних знань, умінь та навичок, тому, на нашу думку, вчителям слід організовувати навчальний процес якомога ближче до ідеального.

Поміркуємо, як цього досягти. Відповідь можна знайти в учителів-новаторів, наприклад у І.П. Волкова¹⁶. Під впливом одного із його виступів ще в першій половині восьмидесятих років ХХ століття закарбувалась чітка схема розвитку творчих здібностей та виявлення професійних нахилів учнів:

¹⁶ І.П. Волков, учитель малювання та праці з м. Реутово Московської області, кандидат педагогічних наук, заслужений учитель школи РСФСР. Його основна новаторська ідея - створення механізму реалізації творчих здібностей дітей і здійснення педагогічного керівництва процесом розвитку індивідуальності кожного школяра.

- 1) вивчаючи будь-який курс, предмет чи окрему тему слід виявити *основні поняття* та *їх властивості* (у І.П. Волкова, наприклад, інструменти у майстерні по дереву та основні навички роботи з ними);
- 2) досконало вивчити виявлені основні (робочі) поняття та до автоматизму відпрацювати стандартні прийоми їх використання (у майстерні – це правила техніки безпеки при використанні ножа, пилки, стамески тощо та стандартні прийоми їх застосування при роботі з деревом);
- 3) відпрацювати комплексне застосування робочих понять (наприклад, одночасне використання стамески і молотка);
- 4) створити умови для *самостійної творчої роботи* у рамках теми з вибраними поняттями та їх властивостями (створення виробу із дерева).

По суті, ми підійшли до другого робочого принципу (**Б**), необхідного для створення ЕКАПу. Коротко його можна сформулювати так: ***забезпечувати навчання не на рівні запам'ятання а на рівні розуміння, щоб після закріплення зникла потреба в повторенні раніше вивченого.***

На користь цього принципу наведемо досить тривіальний приклад. Вчителі інформатики знають, як важко призвичаїти учнів до строгості мови програмування, застерегти від багаторазового повторення банальних помилок порушення її синтаксису, дотримання структури програми тощо. Візьмемо фрагмент довільної програми, що містить блок вводу величини *a* з виводом підказки:

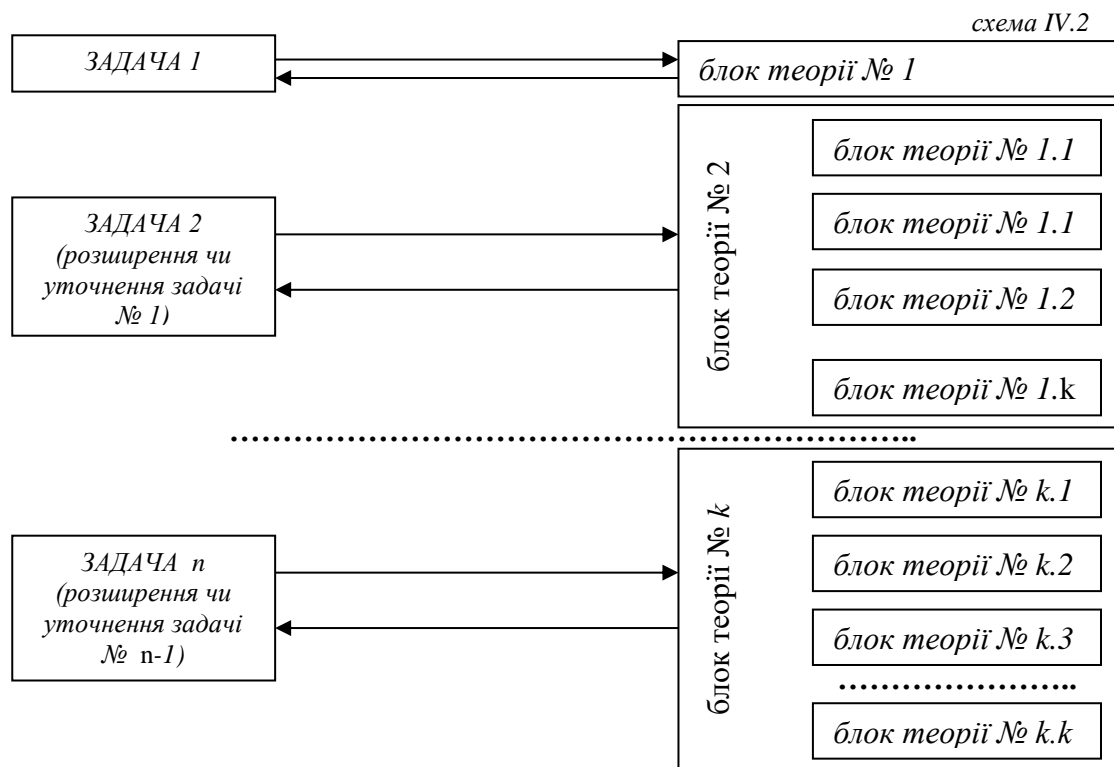
```
{...} var a:real; {...}
begin
  {...} Write('a?');Read(a); {...}
end.
```

Можна спостерігати, як у подібних ситуаціях окремі учні на протязі багатьох уроків набирають “різні варіанти“:

- 1) *Write(a?);Read(a);*
- 2) *Write('a?');Read('a');*
- 3) *Write('a?') Read(a)* тощо.

Як подолати раз і назавжди допущення таких помилок, щоб не тратити час на кожному уроці на повторення формату та різних варіантів команд вводу і виводу? Дуже просто. Дати учням можливість “набити“ усі “шишки“ відразу, виконавши наведені варіанти в середовищі системи програмування. Учень повинен *усвідомити, а не запам'ятати* правильний запис.

Для цього у кожного є свій ліміт повторень ситуацій. Очевидно, що краще цей ліміт вичерпати відразу. Читачеві нічого не залишається, як погодитись, що після використання описаного дидактичного прийому важко буде знайти учня, якому для уникнення помилок необхідне повторення.

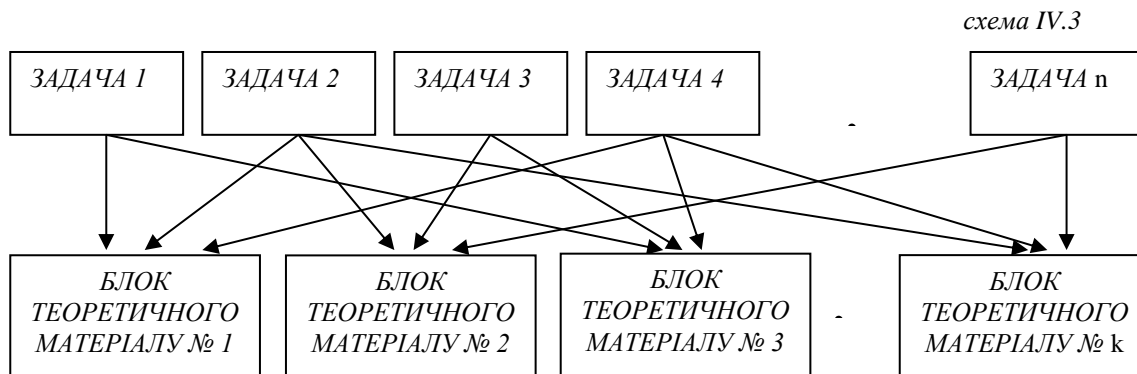


Усе це дасть значну економію часу і дозволить навчати без надмірних інтелектуальних напружень. А наведена щойно схема розвитку творчих здібностей учнів зовсім не є інноваційною, вона лише вимагає будувати ЕКАП по строгості й логічності подібно аксіоматичному методу побудови курсу геометрії.

Безперечно, на практиці впровадження принципів **А** та **Б** не таке прозоре і зрозуміле, тому додатково скористаємось схемами 2 та 3.

Якщо при вивченні алгоритмізації і програмування дотримуватись наведеної вище схеми IV.2, то можна зняти психологічні складності, отже підвищити якість засвоєння навчального матеріалу. Звичайно, при підборі задач необхідно уявляти весь курс і починати із останньої задачі. Нехай це *задача n* про пошук у відсортованому лінійному числовому масиві. Логічно найближча до неї задача про сортування лінійного числового масиву, тому назвемо її *задачею n-1* і будемо вважати попередньою. Для задачі *n-1* попередньою може бути задача *n-2* про відшукування найменшого елемента в лінійному числовому масиві і т.д. Міркуючи таким чином, поступово можна дійти до задачі *1*. Але задачу *n* можна сформулювати настільки детально, що вона міститиме всі попередні - від задачі *n-1* до першої включно. На кожному етапі деталізації чи ускладнення поточної задачі можна підключати необхідний блок теоретичного матеріалу. В кінці-кінців весь курс буде присвячений розгляду однієї глобальної задачі з поступовим нарощуванням блоку теоретичного матеріалу, як і передбачено принципом **А** (див. схему IV.2).

Застосовувати **A** можна лише умовно, адже постійно доповнюючи умову задачі, ми одержуватимемо інші задачі (наприклад, із списку 1). Крім того, можна укрупнити ЕКАП, використавши не одну, а кілька базових задач, які не зводяться одна до одної. Уявимо, що в курсі їх буде n . Звернемо увагу на те, що і блоки теоретичного матеріалу тоді в задачах будуть застосовуватись не так, як показано у списку 1, а більш хаотично (див. схему IV.3).



Як уже зазначалось, для забезпечення вивчення алгоритмізації та програмування в прискореному і ефективному режимі, тобто режимі експрес-курсу, необхідні додаткові резерви часу і обов’язкова організація самостійного виконання учнями необхідних теоретичних і практичних завдань. Ми домовились застосовувати робочі принципи, два з яких (**A** та **B**) уже сформульовані. Але по відношенню до учнів вони мають інтерактивний, тобто спонукальний із зовні, характер. У той же час будь-який курс повинен *не стільки викладатись, як вивчатись*. Отже необхідна ще хоча б одна робоча ідея, покликана бути *внутрішнім імперативом* або *стимулятором* для учнів. Назвемо її принципом **B** – *орієнтувати учнів на самонавчання*.

Один мудрець вказав три шляхи удосконалення: *легкий – наслідування, важкий – досвід і шляхетний – міркування*. Проаналізуємо процес навчання. Кажуть, що воно буває пасивне, активне та інтерактивне. З цим погодитись ніяк не можна. Хіба не безглуздо стверджувати, що існує пасивне навчання? *Навчання – це процес, в якому одночасно і активно приймають участь або дві особи – учитель і учень, або одна особа - учень*. Якщо учень пасивний (читай - відсутній), то це зовсім не навчання, причому, можна сказати “по визначенню“. Отже, **будь-яке навчання активне**. Активність, тобто мотивація і стимуляція можуть бути внутрішніми чи зовнішніми. Інтерактивний - стимульований із зовні. Тому інтерактивні елементи в навчально-виховному процесі корисно застосовувати не постійно, а епізодично, наприклад, для актуалізації, концентрації, закріплення чи повторення, які в задуманому нами експрес-курсі повинні відігравати допоміжну роль. Слід завжди враховувати “протипоказання“ інтерактивних методів, адже, як окремий вид маніпуляцій свідомістю, вони при надмірному “вживанні“ можуть привести до формування в учнів ефекту (а то й дефекту) *залежності*, тобто *нездатності самостійно мислити, виконувати роботу, приймати рішення* тощо.

Дуже корисно на першому, ознайомлювальному уроці запитати учнів: “Чому ви сьогодні прийшли до кабінету інформатики?”. Як правило, останні від несподіванки не знають, що відповісти. Ось зразки відповідей: “вивчати комп’ютери”, “згідно розкладу”, “трати в комп’ютерні ігри”. Але, бачачи, що вчитель чекає іншої відповіді, учні можуть “гадати” далі: “слухати”, “думати”, “щоб нас навчили” тощо. Нарешті почується відповідь: “учитись”. Це буде тим, що потрібно. Далі слід запитати: “Що означає *учитись*?” Скоріше всього, учні не зможуть правильно відповісти, бо вони ще не розуміють, що *учитись* – це самостійно, без учителя. Тоді їм слід пояснити те, що сказане в двох попередніх абзацах, переконати, що без орієнтації на самонавчання, вони ніколи не знатимуть більше за вчителя, будуть обмежені його знаннями та здібностями. І тільки після цього учні можуть, якщо не сформулювати, то хоча б зрозуміти формулу: *учитись – це самостійно шукати запитання та відповіді на них*.

Щойно ми продемонстрували ефект несподіванки, висловлений ще одним древнім філософом. Взагалі, в навчально-пізнавальному процесі повинно бути якомога більше несподіваних ситуацій, це приводять до пізнання через здивування. Назвемо його принципом **Г** - *пізнавати через здивування*, який разом із принципом **В** повинен стати *мотиваційною основою* ЕКАПу. Для забезпечення повноти системи взятих принципів, до них слід долучити ще один – *принцип творчого співробітництва* учнів з учителем (**Д**).

У наступному розділі пропонується орієнтовний ЕКАП з програмування мовою Паскаль, адже вище ми домовились розглядати методіку вивчення теми “Алгоритмізація і програмування” у такій формі, а створення експрес-курсу – це, відповідно до умов і бажань, справа учителя. Кожна с тем, що називаються *Уроками*, розрахована для вивчення протягом одного уроку, але при обов’язковій умові самостійних відповідей на запитання та виконання завдань, які пропонуються в кінці кожної теми. Курс не слід розглядати, як збірник поурочних планів учителя. Взавши пропонований ЕКАП за зразок, учителі повинні розуміти, що введення в дію принципів **А-Д** для учнів повинно бути поступовим, обережним, послідовним і не помітним. Наприклад, повністю відмовитись від систематичних повторень вивченого раніше (принцип **Б**) можна лише при забезпеченні свідомого дотримання принципу **А** у поєднанні з мотиваційними принципами (**В-Д**).

IV.3 Висновки.

Як зазначалось в другому пункті, пропонований ЕКАП повинен відповідати чинній програмі з основ інформатики для універсального профілю [1], тому будуватимемо його синхронно затвердженому МОН України підручнику [15]. Але, дотримуючись вимоги програми щодо кількості відведених навчальних годин і форми експрес-курсу, ми змушені шукати шляхи ущільнення.

У темі “Комп’ютерне моделювання. Основи алгоритмізації та програмування” фактично не відображається *власне програмування*, якщо не брати до уваги кількох окремих згадок терміну “програма”, зокрема: “поняття програми”, “структура опису алгоритму навчальною алгоритмічною мовою та

мовою програмування“ і “описання найпростіших алгоритмів навчальною алгоритмічною мовою та мовою програмування“.

Підручник [2] містить три розділи: 3. “Етапи розв’язування задач за допомогою комп’ютера“, 4. “Вступ до алгоритмізації“ та 5. “Початки програмування“. Як і частина програми про інформаційну модель, розділ 5 підручника дуже розпливчатий і не конкретний. Там багаторазово у різних контекстах згадується термін “модель“, але, на наш погляд, так і нема однозначних відповідей про відмінність понять інформаційної і математичної моделей, комп’ютерного моделювання і побудови моделі, у підручнику немає чіткого і однозначного трактування етапів розв’язування задачі з допомогою комп’ютера. Ми намагатимемось лаконічно і прозоро сформулювати логічний ланцюжок *інформаційна модель – математичне моделювання – етапи комп’ютерного моделювання*

На наш погляд, фактично відсутній чіткий перехід *алгоритм – програма*. Окремий розгляд, як це зроблено в підручнику, вступу до алгоритмізації та початків програмування в умовах обов’язковості практичної форми вивчення теми також не доцільний. Не відходячи від істини, можна вважати, що *програма – це запис алгоритму розв’язання задачі мовою програмування*. Отже є сенс у ЕКАП об’єднати розділи “ Вступ до алгоритмізації “ та “ Початки програмування“.

Але для окремого розгляду алгоритмізації та програмування, що традиційно і цілком природньо робиться, є важливі причини. При розгляді питань алгоритмізації *основна увага концентрується на реалізації математичної моделі задачі в алгоритмі*, тобто домінує логічна лінія. Завдяки цьому при вивченні програмування *стає можливим зосередитись виключно на вивченні мови програмування і нюансах запису алгоритму у вигляді програми*. Підручники з мов програмування, як правило дуже об’ємні й детальні, очевидно, що шкільною програмою не передбачено їх вивчення. Отже виникає ще одна проблема: *як в ознайомлювальному курсі засвоїти основи мови програмування?* Ми також не бачимо можливості в нашому експрес-курсі для більш чи менш детального викладання основ мови *Pascal*. Тому нижче будемо розглядати лише вузлові питання запису програм, причому, як уже акцентувалось, по можливості об’єднуючи опис алгоритму і програми. Щоб заповнити прогалини в описі конструкцій *Pascal*, які не важко помітити при дуже стислому викладі матеріалу, ми будемо формулювати питання і ставити завдання, наведені в кінці кожного уроку, так, щоб учень, працюючи з ними, міг самостійно “відкривати“ елементарні відомості, опис яких відсутній. Виконуючи завдання, учень буде змушений користуватись електронною довідкою, і таким чином отримувати достатньо повне уявлення про мову програмування. В зв’язку з цим зручно користуватись посилатись на версію *PascalABC*¹⁷.

¹⁷ На рівні програми ЗОШ вона цілком відповідає найчастіше вживаним версіям *Pascal*. Крім того ця версія надає багато переваг, зокрема, допускає “довгі імена“, використовує зручнішу систему числових типів, має стандартні підпрограми, наприклад функцію *Power(x,y)*, має русифіковані меню, довідку та

Піднявши питання перегляду ролі і місця теми “Комп’ютерне моделювання. Основи алгоритмізації та програмування” у курсі сучасної шкільної інформатики, ми переслідували ще одну важливу мету. Як уже зазначалось, згідно діючих програм ця тема вивчається фактично оглядово, що затрудняє для учнів її логічне усвідомлення в цілому. Але при цьому, в рекомендованих МОН України підручниках та навчальних посібниках продовжує відображатись класична, обтяжена деталями, схема подання і закріплення навчального матеріалу з алгоритмізації та програмування, яка багато в чому є калькою технологічного, призначеного для майбутніх студентів технічних вузів, підходу до програмування. Але з позицій шкільного вчителя основ інформатики видно проблеми:

- а) чіткості методологічних та філософських засад вивчення курсу;
- б) підпорядкованості вивчення алгоритмізації та програмування простій і логічній загальній схемі;
- в) слабкості органічного зв’язку із основною частиною курсу основ інформатики.

Вирішення проблеми а) бачиться в усвідомленні та реалізації тези “інформатика іде від математики” та вивченні теми раніше на один рік, тобто в першому семестрі 10-го класу, відразу після ознайомлення з основним системним програмним забезпеченням (операційні системи та сервісне програмне забезпечення) і вивчення простого текстового редактора.

Не виходячи із ліміту часу, передбаченого діючими програмами, подолати проблему б) можна, змінивши хронологію та акценти всередині цієї теми за такою схемою:

1. *Поняття про алгоритми і найпростіші програми.*

- 1.1 Поняття інформаційної та математичної моделі. Етапи розв’язування задач з допомогою комп’ютера.
- 1.2 Алгоритми, їх властивості та виконавці. Система команд виконавця. Форми подання алгоритмів. Поняття програми.
- 1.3 Структура програми. Опис величин мовою програмування. Лінійні програми опрацювання величин. Команди вводу, виводу та присвоєння.
- 1.4 Виконання програми та аналіз її правильності. Покрокове виконання і відлагодження програм.

2. *Структуровані алгоритми та їх програмна реалізація.*

- 2.1 Діалогові програми. Структури повторення та їх графічні схеми.
- 2.2 Структури розгалуження та їх графічні схеми.
- 2.3 Конструювання алгоритмів методами покрокової деталізації. Підпрограми і вказівки їх виклику. Формальні та фактичні параметри.
- 2.4 Структуровані алгоритми.

3. *Структуровані величини та їх опрацювання.*

- 3.1 Структуровані типи величин. Табличні величини (масиви). Лінійні масиви та алгоритми їх опрацювання.
- 3.2 Алгоритми пошуку елементів у таблиці.
- 3.3 Алгоритми упорядкування лінійних таблиць.
- 3.4 Рядкові величини та алгоритми їх опрацювання.

ін. Щодо потужності, то практика підтверджує, що ця версія дозволяє реалізувати навіть непересічні “олімпіадні” задачі.

Згідно цієї схеми тема розбита на три частини по чотири уроки. Перша з них присвячена формуванню понять алгоритма і програми. Як буде видно з наступного розділу “Експрес-курс алгоритмізації та програмування”, вже в першій частині наявні дещо відмінні від канонічних трактування алгоритму, команди, програми, процедури, функції, багато уваги приділяється вводу/виводу (з цим пов’язані міркування про лінійні та діалогові алгоритми і програми, в ознайомлювальному плані розглядається поняття текстового файлу та питання використання зовнішніх файлів для вводу/виводу). Це зроблено з тим, щоб учні мали змогу краще уявити структуру і логіку теми та її місце в загальній системі знань.

Акцентується увага на терміні “структура”, зокрема, дається чітке поняття *структури алгоритму та програми*. Далі структурованість також є вузловим поняттям, у другій частині ідеться про *структуровані алгоритми*, а в третій про *структуровані типи*.

Ми вважаємо, що навіть при оглядовому вивченні теми повинні детально і чітко розглядатись не тільки питання математичної постановки задачі, а й питання відлагодження та виконання програми, цьому присвячений урок 4.

У другій частині вузловим є питання конструювання програми відповідно принципу покрокової деталізації. Умисно поставлені акценти на конструюванні методом “згори донизу”, пов’язуючи його з поняттям про стандартні підпрограми та “знизу догори”, розвиваючи ідею створення процедур і функцій користувача.

Логічним підсумком другої частини є останній урок, де вивченням циклу з параметром закінчується формування уявлень про структуру повторення, що дає змогу розглядати програми, які містять усі базові структури алгоритмів (БСА). Ми вважаємо за доцільне на відміну того, як це робиться в підручниках, БЗА розглядати окремо, вводячи їх поступово, відповідно до ускладнення задач. Це допоможе учням усвідомити суть структурного програмування взагалі.

В третій частині розглядаються найважливіші структуровані типи, зокрема масиви і рядки. У двох перших частинах вони лише згадуються та епізодично використовуються на прикладі типу *string*. Але, доводиться визнати, що, виходячи з ліміту навчального часу, у ЕКАП немає змоги для ознайомлення із записами та множинним типом, а також з рештою порядкових типів, вказівниками та процедурними типами.

Щодо проблеми в) можна сказати наступне. Необхідного органічного зв’язку із основною частиною курсу основ інформатики (інформаційно-комунікаційні технології) можна досягти, по-перше, вирішивши проблеми а)-б), по-друге, забезпечивши наступне після теми “Комп’ютерне моделювання. Основи алгоритмізації та програмування” знайомство із об’єктно орієнтованим програмуванням (ООП) та візуальним проектуванням (ВП) і узгодивши, а в окремих випадках навіть підпорядкувавши, вивчення прикладного програмного

забезпечення загального і навчального призначення (ППЗП, ППЗНП) із темою “Комп’ютерне моделювання. Основи алгоритмізації та програмування”.

Кожен урок ЕКАПу є окремою темою, що, здебільшого, не можна вивчити за одну календарну годину, тому, як уже наголошувалось, слід шукати додаткові резерви часу. Але уроки написані так, що на розсуд учителя, їх можна вивчати фрагментарно. Для кращого закріплення у вигляді не обов’язкового додатку в кінці ЕКАПу є практикум із шести практичних робіт, які охоплюють прикладні і практичні задачі.

Поряд із цим відмітимо, що ЕКАП, або його ідея, може успішно застосовуватись і при значно ширшому та глибшому вивченні теми. Його можна розглядати, як *вступ до основ програмування*.

V. Міжпредметні зв’язки та математична складова курсу основ інформатики.

V.1 Значення міжпредметних зв’язків для інформатики.

Однією із пріоритетних у дослідницькій роботі ОШППД була проблема міжпредметних зв’язків інформатики з іншими навчальними предметами. Вони можуть мати різні ступені складності та концентрації, але наявні лише тоді, коли на уроці *акцентується увага на деяких елементах, що належать іншому предмету*. Застосування комп’ютерної презентації на уроці біології або історії зовсім не говорить про міжпредметні зв’язки з інформатикою, тут має місце звичайне застосування засобів однієї галузі для пояснення чи дослідження в іншій, зокрема ІКТ. Міжпредметні зв’язки можуть бути *епізодичними* чи *систематичними* як в межах окремої теми, так і в межах одного уроку. Їх форми достатньо досліджені методистами.

Для інформатики дуже актуальними є міжпредметні зв’язки з математикою, фізикою, англійською, українською та російською мовами, причому, у першу чергу з математикою. Вони завжди дуже різноманітні та багатогранні, тому підлягають детальному розгляду. При вивченні міжпредметних зв’язків їх слід представляти у вигляді орієнтованого графа і розглядати попарно. Досліджуючи зв’язки між інформатикою та математикою, необхідно вивчати окремо зв’язки *інформатика→математика* та *математика→інформатика*.

Немає сумніву, що математична складова домінує не тільки в темі “Математичне моделювання. Алгоритмізація та програмування”, а й у темі “Інформаційно-комп’ютерні технології”, тому вона дуже важлива у курсі основ інформатики в цілому. За влучним висловом А.П. Єршова, *інформатика іде від математики*, тому при вивченні інформатики слід використовувати будь-яку, в тому числі не пряму, а опосередковану нагоду підключення математичного компонента. Навіть розглядаючи перші теми, наприклад, питання про вимірювання інформації, корисно глибоко вивчити позиційні системи числення, акцентувати увагу на питанні про оптимальність розміру базової комірки пам’яті в 1 байт. При цьому слід довести, що 8-бітова комірка може зберігати $2^8=256$ варіантів двійкових кодів і зробити висновок про її оптимальність для зберігання інформації. Це приклад *прямого зв’язку математика→інформатика*. Вивчаючи об’єкти у *Microsoft Word*, корисно

звернути увагу на роботу з редактором формул *Equation*, пропонуючи завдання на створення та редагування математичних виразів. Тут спостерігається *опосередкований зв'язок інформатика→математика*. Прикладів прямого та опосередкованого використання математичного апарату на уроках інформатики можна навести дуже багато.

За відомим висловом М.В. Ломоносова, математику ще й тому слід вивчати, що вона розум до ладу приводить. Це – постулат, який ніким не ставиться під сумнів. Математику, хоча б в обсязі загальноосвітньої школи, вивчають всі люди, бо вона є невід'ємною складовою середньої освіти.

Але важливо постійно думати над розширенням традиційних меж використання елементів математики при вивченні інформатики. Зауважимо, що цілі математичні теорії проникли у всі галузі людської діяльності як на науковому, так і на побутовому рівні. У першу чергу це стосується арифметики та логіки. Наприклад, у побуті часто користуються логічним апаратом, не думаючи, що тут має місце пряме застосування математики. На наш погляд, інформатика, як прикладна галузь знань, *дає своєрідне уточнення та узагальнення не тільки таких понять, як інформація та алгоритм, а й багатьох загальноосвітніх термінів, від лінгвістичних, наприклад, "алфавіт" та "мова", до природничих і математичних, наприклад, "число", "величина", "тип", або "вид", "вираз", "формула", "функція", "процедура" тощо.* Іншими словами, *інформатика, покликана робити наступний за математикою крок у "приведенні розуму до ладу".*

Про що тут ідеться? Візьмемо поняття "алфавіт". У початкових класах його формують у вузькому розумінні, лише як множину знаків, що позначають звуки конкретної національної мови. Але чи правильно, коли дорослі люди так розуміють поняття "алфавіт" до кінця життя? Навіть у XXI столітті ми спостерігаємо, до яких тяжких для людства наслідків приводить довільне трактування історичних, політичних, економічних і взагалі – наукових понять. Тому без перебільшення можна стверджувати, що все починається з дуже приблизного уявлення про алфавіт. На уроках інформатики при вивченні мови програмування учні дізнаються, що алфавіт у *широкому розумінні* – це набір усіх символів, необхідних для висловлення думок. А потім переконуються, що, до українського алфавіту на рівних правах, наприклад, з буквою "а", входять арабські цифри, розділові знаки і т.д. Якщо формування такого уявлення про алфавіт не передбачено в початкових класах та на уроках рідної мови, то вивчення мови програмування доцільне хоча б з огляду на це. Але, на наше глибоке переконання, будь-який учитель-класовод чи мовник повинен у свій час узагальнити учням поняття алфавіту, бо неможливе розуміння процесу пізнання без принципу "від споглядання до узагальнення", адже аналіз та синтез – це α і ω мислительського процесу.

Порівняймо поняття величини чи функції у математиці та фізиці – з одного боку, та в інформатиці – з іншого. Величина у першому випадку має характеристики: ім'я та значення. Про тип величини майже не згадується, адже з точки зору математики і фізики величини фактично бувають лише числові,

правда існує поділ на скалярні та векторні. З програмування учні дізнаються про кодування величин, про потребу їх оголошення та опису, про різноманіття типів, зокрема у мові Паскаль, у якій існує по кілька цілочисельних та дробових типів. На перший погляд це дивує, але коли пояснити про представлення величини у пам'яті комп'ютера, здивування проходить. Ім'я величини в математиці не обговорюється. Час – t , швидкість – v і т.д. Мови про позначення величини, як деякого набору допустимих символів, нема. Але про це доводиться говорити в інформатиці, де ім'я, чи ідентифікатор може бути не тільки у величини, а й у файлу, папки, диску, програми тощо. Коли пояснюєш, що ім'я, в тому числі і лідини – це лише позначення, деякий набір символів, у багатьох виникає протиріччя, пов'язане із засвоєною раніше абсурдною і шкідливою для наукового світогляду містикою імен. Але учні перестають дивуватись і погоджуються, коли їм говорять, що ім'я “Володимир” для американського індіанця так само є лише набором звуків, як для нас “Чингачук”. Таким чином реалізується *процес пізнання через здивування*. Лише сміх замість здивування викликало повідомлення, що кандидатом у президенти в жовтні 2009 р. став громадянин В. Гуменюк, який для реєстрації змінив своє прізвище на “Противсіх”). Отже, щоб не виникало дилетантських уявлень, необхідно ставити наукові (читай логічні, математичні) акценти.

Необхідно хоча б коротко пригадати поняття функції. Після досить детального ознайомлення із ним у школі, багатьом студентам на лекціях з математичного аналізу лише завдяки великим зусиллям вдається зрозуміти класичні означення функції. Але, щоб досягти глибини усвідомлення, у математиці уявлення про функцію розглядається дуже вузько. В інформатиці необхідно акцентувати, що функція - це не лише залежність між двома змінними, при якій кожному значенню однієї відповідає єдине значення другої, а й *дія*, яка певне значення аргументу перетворює у цілком визначене значення функції. Для учнів цікаво, що термін “дія” тут слід розуміти, як “автомат”, подібно тому, як електром'ясорубка – це автомат для перетворення м'яса у фарш. У процесі таких міркувань свідомість учнів робить колосальний світоглядний скачок, багато різних за формою явищ зводяться, так би мовити, “до спільного знаменника”. А якщо пригадати, що математика, навіть вища, практично обмежується розглядом функцій двох числових аргументів, а інформатика, точніше розділ “Алгоритмізація і програмування”, розглядає поняття функції багатьох аргументів, причому, не тільки числових типів, а саме поняття функції у певному розумінні зводиться до поняття *процедури*, то можна сказати, що абстрактне математичне поняття функції в інформатиці набуває живий, зрозумілий без складного математичного апарату образ. От і спробуй не погодитись, що інформатика, як сказано дуже доречно вище, доповнює математику у “приведенні розуму до ладу”, тобто у формуванні наукового світогляду.

Є цілий ряд тем, у яких математичний компонент домінуючий. Красномовним прикладом є тема “електронні таблиці”, адже основне завдання табличного процесора *Microsoft Excel* - це автоматизація обчислень. Але і тут,

якщо не слідувати сліпо за підручником [15], є невичерпні додаткові можливості непрямого використання математики. Треба лише творчо підходити до вибору завдань. Для опису формування *робочого листа таблиці Microsoft Excel*, дій з *майстрами і шаблонами* у підручнику розглянуто, задачі на рух та про визначення кількості мікробів у класі, що припадають на одного учня у різний час і при різній температурі повітря. Ефективнішими при вивченні формул та майстра функцій були б приклади створення таблиці, що містить, наприклад, довільну кількість квадратних рівнянь і їх розв'язків, чи таблиці Брадіса для визначення значень тригонометричних функцій (синус-косинус, тангенс-котангенс) для кута α ($0^\circ < \alpha < 90^\circ$), причому не тільки чотиризначних, а й для більшої розрядності. Дуже цінним математичним моментом при виконанні цього завдання є обчислення та відображення кутів у градусній мірі. Взагалі ж цінність міжпредметних зв'язків інформатики з математикою у можливостях проведення чисельних експериментів, досліджень, аналізу та синтезу, побудови математичних моделей, як пропедевтичний етап для програмування.

Таких зразків прямого й доцільного використання математики при вивченні будь-яких розділів інформатики можна навести дуже багато. Так само широко на уроках математики можна включати можливості ІКТ та відомості з основ інформатики. Ось ще приклади. Приблизно одночасно у восьми класах на уроках алгебри вивчають арифметичний квадратний корінь, ірраціональні та дійсні числа, у геометрії задачі на побудови вписаних чи описаних трикутників та чотирикутників, а на уроках інформатики побудову графіків та діаграм. Все це можна ефективно поєднати. Демонструючи побудову у *Microsoft Word* малюнків, графіків функцій, зразки розв'язань варіантів контрольної роботи з геометрії чи розбір завдань олімпіади з математики, можна піднести інтерес до програмного забезпечення та вивчення основ інформатики, показати якісно відмінну інформатику від "зависання" в інтернеті або від "різання" до виснаження у комп'ютерні ігри.

V.2 Бінарні уроки.

Отже, можна вести мову про *уроки інформатики із математичними елементами* та *уроки математики із застосуванням інформатики*. Найвищий ступінь міжпредметних зв'язків досягається в умовах проведення *бінарних уроків*. При вивченні інформатики бінарні уроки заслуговують на особливу увагу, тому їм майже цілком було присвячено восьме заняття ОШППД. Але такий тип уроків слід відрізняти від звичайних *міжпредметних* та *інтегрованих* уроків.

Міжпредметний урок – це форма занять, яка застосовується на всіх ступенях навчання. Його особливості – *навчальний матеріал ілюструється відомостями з інших предметів*, забезпечує при цьому синхронність навчання по темах кількох предметів, що перетинаються, які розділені в часі на місяці або роки. Міжпредметний урок може вести один учитель.

Інтегрований урок – це заняття, на якому *визначена тема розглядається з різних точок зору*, засобами деяких предметів чи курсів. Ведуть його кілька

учителів. Наприклад: пресс-конференція, конференція-захист, круглий стіл тощо.

У методичній літературі *бінарні уроки* та їх завдання трактуються по різному. Переважає застаріле, надто спрощене уявлення, коли під бінарними маються на увазі всі уроки із елементами міжпредметних зв'язків, уроки з використанням інформаційно-комп'ютерних технологій, навчаючих програм тощо.

Інший погляд на бінарні уроки трактує їх, як варіанти *нетрадиційних уроків*¹⁸, що часто використовуються в інноваційній роботі та спрямовані на *об'єднання окремих тем суміжних предметів у рамках одного уроку*, наприклад фізики та біології, математики та фізики, математики та хімії, хімії та біології. Інакше кажучи, бінарним уроком вважається навчальне заняття, яке об'єднує вивчення тем двох предметів одного циклу (або освітньої галузі) в одному уроці. З таким трактуванням слід погодитись, але із деякими застереженнями. Адже дві різні теми не можна вивчати *одночасно і паралельно*. Їх можна *об'єднати в одну, ширшу тему, або вивчати послідовно, застосувавши результати вивчення першої для вивчення другої*. Тобто, у бінарних уроках *виклад чи дослідження проблеми одного предмету знаходить продовження в іншому*, міжпредметні зв'язки реалізуються в процесі викладання дисциплін однієї освітньої галузі.

Традиційно вважається, що бінарний урок як правило проводять два учителі, а тому *основною метою бінарних уроків вважається розвиток співробітництва педагогів і формування в учнів переконання в зв'язності та єдності предметів*, що не може становити особливої цінності для процесу навчання. Дійсно, чи має *принципове значення* для вивчення окремих предметів *співробітництво двох педагогів*? Відчуття зв'язності та єдності предметів також не може бути самоціллю, адже воно природньо виникає на етапі узагальнення і є функцією міцності засвоєння знань, умінь та навичок.

Дотримуючись вищеописаних ознак та вимог, інколи можна "створити" (що й зустрічається в методичній літературі) досить незвичайний, зовні а то й по змісту навіть казуїстичний зразок уроку: учитель математики проводить зі своїм колегою по фізкультурі бінарний урок, наприклад, "Естафета с математичними завданнями". Пропонується перекинутись уперед через голову, взяти картку, пробігти по перевернутій лаві, зупинитись, визначити, чи стандартний вигляд одночлена записано на картці, і покласти картку на підлогу. Наступні вправи естафети: добігти до парти, записати одночлен в стандартному вигляді, назад рухатись стрибками на правій нозі по лаві, перекинутись уперед через голову.

Згідно опису Ю.А. Самаріна даний вид заняття цікавий тим, що можна активізувати мислительську енергію через фізичні вправи. Заняття проходять на музичному фоні. Учні одержують по декілька оцінок за урок.

Правда там же можна зустріти застереження, що такі бінарні уроки не можуть застосовуватись часто у педагогічній практиці через їхню складність у

¹⁸ Бінарний урок за Ю.А. Самаріним).

підготовці, а також через протипоказання: в застосуванні бінарних уроків є неузгодженість та несумісність педагогів (протипоказань можна знайти ще).

Виходячи з вищесказаного, на нашу думку, можна так сформулювати поняття бінарного уроку: *Бінарними* слід вважати лише такі уроки з міжпредметними зв'язками із *двох предметів*, коли питання, що розглядаються в темі уроку, *стосуються обох предметів і між собою логічно поєднані*, тобто, **бінарні уроки поєднують два предмети в одній темі**.

Наприклад, можна говорити про бінарні уроки *інформатика – математика*. Безперечно, на кожному бінарному уроці повинна бути встановлена пріоритетність одного із предметів, іншими словами, можуть бути бінарні уроки *інформатика–математика і математика–інформатика*. Компоненти уроку, який має вищий пріоритет назвемо *домінантними*, а іншого – *підпорядкованими*. Але *урок*, в тому числі й бінарний, *повинен проводити, як правило, один учитель*.

Зразок бінарного уроку з математики та інформатики описаний у [19]. Але там, як указано в темі, ідеться про урок геометрії “Паралельність прямих і площин у просторі“ з використанням навчальної програми “Репетитор по математике Кирилла и Мефодия“ та графічного редактора *Paint*, тобто у темі уроку не передбачено розгляд питань з інформатики, а питання про графічні системи, формати графічних файлів, формування кольорів, які в ході уроку все ж ставляться, та самотійна робота за варіантами на побудову стереометричних фігур з допомогою *Paint*, не пов'язані з темою “Паралельність прямих і площин у просторі“. Тому тут скоріше ідеться про урок з використання ІКТ, ніж про бінарний урок.

V.3 Зразки бінарних уроків.

Наведемо приклади бінарних уроків *математика–інформатика* та *інформатика–математика*, які відповідають сформульованому вище уявленню про них.

*бінарний урок¹⁹: математика – інформатика²⁰,
проводиться в кабінеті ОІОТ або в кабінеті ІКТН*

тема: Тригонометричні таблиці для обчислення значень синуса, косинуса, тангенса і котангенса гострих кутів.

мета: Розглянути створення і застосування тригонометричних таблиць кутового аргумента.

обладнання та література:

- М.І. Бурда, Л.М. Савченко, Геометрія, навчальний посібник для 8-9 класівшкіл з поглибленим вивченням математики, Київ, “Освіта“, 2004 р.;

¹⁹ Урок для 8 класу, в якому вивчається інформатика відповідно [17].

²⁰ Домінантні елементи (математика) виділено суцільною рамкою, підпорядковані елементи (основи інформатики) виділено пунктирною рамкою.

- І.Т. Зарецька, А.М. Гуржій, О.Ю. Соколов, Інформатика, частина II, підручник для 10-11 класу загальноосвітніх навчальних закладів, Київ, "Форум", 2004 р.;
- *Microsoft Office Excel 2003*.
- П'ятизначні тригонометричні таблиці гострих кутів (додатки 1-2).
- Таблиця *Sin-Cos-Tg-Ctg.cls*

ПЛАН УРОКУ

1. Повторення

- а) Повторити поняття синуса, косинуса, тангенса і котангенса гострих кутів числового та кутового аргументів.
- б) Повторити таблицю значень синуса, косинуса, тангенса та котангенса $30^\circ, 45^\circ, 60^\circ$:

	<i>sin</i>	<i>cos</i>	<i>tg</i>	<i>ctg</i>
30°	$\frac{1}{2}=0,5$	$\frac{\sqrt{3}}{2}\approx 0,8660$	$\frac{\sqrt{3}}{3}\approx 0,5774$	$\sqrt{3}\approx 1,7321$
45°	$\frac{\sqrt{2}}{2}\approx 0,7071$	$\frac{\sqrt{2}}{2}\approx 0,7071$	1	1
60°	$\frac{\sqrt{3}}{2}\approx 0,8660$	$\frac{1}{2}=0,5$	$\sqrt{3}\approx 1,7321$	$\frac{\sqrt{3}}{2}\approx 0,5774$

- в) Повторити формули переведення кутів з градусної міри в радіанну і

навпаки:

$$360^\circ = 2\pi \text{ рад} \Rightarrow \begin{cases} 1^\circ = \frac{\pi}{180} \text{ рад} \Rightarrow n^\circ = \frac{n \cdot \pi}{180} \text{ рад}; \\ 1 \text{ рад} = \left(\frac{180}{\pi}\right)^\circ \Rightarrow k \text{ рад} = k \cdot \left(\frac{180}{\pi}\right)^\circ \end{cases}$$

(1)

- г) Повторити формули переведення кутів з *градусів* у *мінути* і навпаки:

$$1^\circ = 60' \Rightarrow \begin{cases} 1' = \left(\frac{1}{60}\right)^\circ \Rightarrow n' = \left(\frac{n}{60}\right)^\circ; \\ k^\circ = 60k' \end{cases} \quad (2)$$

- д) Повторити вставку формул у комірки таблиць *Excel*.

Звернути увагу на те, що формула записується у *рядку формул* і починається значком "=" (чому?!), а після її занесення у *комірці* відображається значення, обчислене за формулою.

	А	В	С
1	мінути	градуси	
2	3	0,05	
3	12	0,2	
4	13	0,216667	
5	14	0,233333	
6	15	0,25	

2. Вивчення нового матеріалу

I. Постановка та розв'язання проблем:

Розглянути на прикладах застосування тригонометричних таблиць для гострих кутів:

а) таблиці у текстовому форматі: *Sin-Cos; Tg-Ctg* – додатки 2-3

– Як обчислити значення \sin , \cos , tg , ctg для кутів:

1) 12° ; 2) $12^\circ 6'$?

– Чи можна обчислити значення \sin , \cos , tg , ctg для кутів:

3) $12^\circ 7'$; 4) $12^\circ 10'$ з допомогою даних друкованих таблиць? Чому?

б) Зміни в таблицях текстового формату.

Слід звернути увагу на те, що подібні традиційні друковані таблиці, як

таблиця 1

СИНУСИ										поправки:				
0	6	12	18	24	30	36	42	48	54	1'	2'	3'		
0	0	0,00175	0,00349	0,00524	0,00698	0,00873	0,01047	0,01222	0,01396	0,01571	89	0,00058	0,00104	0,00174

показано нижче, (див. таблицю 1) мають поправки на 1', 2', та 3'.

Проте практично такої потреби немає, адже ці поправки можна легко розрахувати вручну, знайшовши різницю двох сусідніх значень і поділивши її на 6.

Тригонометричні таблиці, наведені в додатках 2-3, можна легко створити з допомогою *Excel* (таблиця 2).

в) таблиці у форматі *Excel*: *Sin-Cos-Tg-Ctg* – *Excel*

таблиця 2

D4							D5						
fx = SIN((A4+12/60)*ПИ()/180)							fx = SIN((A5+D2/60)*ПИ()/180)						
	A	B	C	D	E	F		A	B	C	D	E	F
1						СИН	1						СИН
2		0	6	12	18	24	2		0	6	7	18	24
3	0	0	0,00175	0,00349	0,00524	0,00698	3	0	0	0,00175	0,00204	0,00524	0,00698
4	1	0,01763	0,0192	0,02094	0,02269	0,02443	4	1	0,01763	0,0192	0,01949	0,02269	0,02443
							5	2	0,0349	0,0364	0,03693	0,04013	0,04188

Як видно з таблиці 2, в рядку формул застосовано формули (1) та (2), що розглядаються при повторенні²¹.

II. Зміни в таблицях формату *Excel*.

В електронних тригонометричних таблицях, створених з допомогою *Excel*, у рядку формул можна змінити значення, що додається до аргументу в стовпчику А, (порівняйте таблиці 2-3), наприклад, в таблиці 3 значення комірки B_3 дорівнює 2', відповідно B_5 дорівнює 60'.

²¹ Тут реалізується основний принцип, що лежить в основі сформульованого вище уявлення про бінарний урок.

C5	=SIN((A5+B5/60)*ПИ()/180)					таблиця3
	A	B	C	D	E	F
1	Синус, косинус, тангенс і котангенс гострого кута					
2	градуси	хвилини	синус	косинус	тангенс	котангенс
3	3	2	0,052917	0,998599	0,052991	18,87107
4	30	0	0,5	0,866025	0,57735	1,732051
5	59	60	0,866025	0,5	1,732051	0,57735

3. Закріплення

а) Користуючись таблицями (додатки 2-3), визначити $\sin 2^{\circ} 60'$; $\sin 3^{\circ} 0'$.
Результати порівняти;

б) З допомогою *Excel* модернізувати таблицю 2, яка містить *sin-cos* кутів, значення яких відрізняються на $3'$;

в) Модернізувати таблицю 2 так, щоб вона містила значення *sin-cos* з указаною кількістю знаків після коми.

г) Перевірити правильність формул $\sin^2 \alpha + \cos^2 \alpha = 1$; $\operatorname{tg} \alpha = \frac{\sin \alpha}{\cos \alpha}$; $\operatorname{ctg} \alpha = \frac{\cos \alpha}{\sin \alpha}$;

для значень α : 1) 12° ; 2) $12^{\circ} 6'$; 3) $12^{\circ} 7'$; 4) $12^{\circ} 10'$.

4. Висновки та підбиття підсумків

Слід звернути увагу на складність і громіздкість процесу створення тригонометричних таблиць без використання *Excel*, на можливості інтенсифікації обчислювальних процесів з допомогою комп'ютерів. Урок має комбінований характер, але бажано не упустити жодної можливості оцінки роботи учнів як з математики, так і з інформатики.

5. Домашнє завдання²²

додаток 1

П'ЯТИЗНАЧНІ ТРИГОНОМЕТРИЧНІ ТАБЛИЦІ СИНУСІВ І КОСИНУСІВ

(фрагментарно)

СИНУСИ

	0'	6'	12'	18'	24'	30'	36'	42'	48'	54'	
0°	0	0,00175	0,00349	0,00524	0,00698	0,00873	0,01047	0,01222	0,01396	0,01571	89°
1°	0,01745	0,0192	0,02094	0,02269	0,02443	0,02618	0,02792	0,02967	0,03141	0,03316	88°
2°	0,0349	0,03664	0,03839	0,04013	0,04188	0,04362	0,04536	0,04711	0,04885	0,05059	87°
3°	0,05234	0,05408	0,05582	0,05756	0,05931	0,06105	0,06279	0,06453	0,06627	0,06802	86°
4°	0,06976	0,0715	0,07324	0,07498	0,07672	0,07846	0,0802	0,08194	0,08368	0,08542	85°
5°	0,08716	0,08889	0,09063	0,09237	0,09411	0,09585	0,09758	0,09932	0,10106	0,10279	84°
6°	0,10453	0,10626	0,108	0,10973	0,11147	0,1132	0,11494	0,11667	0,1184	0,12014	83°
7°	0,12187	0,1236	0,12533	0,12706	0,1288	0,13053	0,13226	0,13399	0,13572	0,13744	82°
8°	0,13917	0,1409	0,14263	0,14436	0,14608	0,14781	0,14954	0,15126	0,15299	0,15471	81°
9°	0,15643	0,15816	0,15988	0,1616	0,16333	0,16505	0,16677	0,16849	0,17021	0,17193	80°
10°	0,17365	0,17537	0,17708	0,1788	0,18052	0,18224	0,18395	0,18567	0,18738	0,1891	79°
81°	0,98769	0,98796	0,98823	0,98849	0,98876	0,98902	0,98927	0,98953	0,98978	0,99002	8°

²² Конкретно можна визначити лише врахувавши обставини, у яких проведено урок, тому ми віднесемо це до компетенції учителя, що проводив урок.

82°	0,99027	0,99051	0,99075	0,99098	0,99122	0,99144	0,99167	0,99189	0,99211	0,99233	7°
83°	0,99255	0,99276	0,99297	0,99317	0,99337	0,99357	0,99377	0,99396	0,99415	0,99434	6°
84°	0,99452	0,9947	0,99488	0,99506	0,99523	0,9954	0,99556	0,99572	0,99588	0,99604	5°
85°	0,99619	0,99635	0,99649	0,99664	0,99678	0,99692	0,99705	0,99719	0,99731	0,99744	4°
86°	0,99756	0,99768	0,9978	0,99792	0,99803	0,99813	0,99824	0,99834	0,99844	0,99854	3°
88°	0,99939	0,99945	0,99951	0,99956	0,99961	0,99966	0,9997	0,99974	0,99978	0,99982	1°
89°	0,99985	0,99988	0,9999	0,99993	0,99995	0,99996	0,99998	0,99999	0,99999	1	0°
	54'	48'	42'	36'	30'	24'	18'	12'	6'	0'	

КОСИНУСИ

додаток 2

**П'ЯТИЗНАЧНІ ТРИГОНОМЕТРИЧНІ ТАБЛИЦІ
ТАНГЕНСІВ І КОТАНГЕНСІВ
(фрагментарно)**

ТАНГЕНСИ											
	0'	6'	12'	18'	24'	30'	36'	42'	48'	54'	
0°	0	0,00175	0,00349	0,00524	0,00698	0,00873	0,01047	0,01222	0,01396	0,01571	89°
1°	0,01746	0,0192	0,02095	0,02269	0,02444	0,02619	0,02793	0,02968	0,03143	0,03317	88°
2°	0,03492	0,03667	0,03842	0,04016	0,04191	0,04366	0,04541	0,04716	0,04891	0,05066	87°
3°	0,05241	0,05416	0,05591	0,05766	0,05941	0,06116	0,06291	0,06467	0,06642	0,06817	86°
4°	0,06993	0,07168	0,07344	0,07519	0,07695	0,0787	0,08046	0,08221	0,08397	0,08573	85°
5°	0,08749	0,08925	0,09101	0,09277	0,09453	0,09629	0,09805	0,09981	0,10158	0,10334	84°
6°	0,1051	0,10687	0,10863	0,1104	0,11217	0,11394	0,1157	0,11747	0,11924	0,12101	83°
7°	0,12278	0,12456	0,12633	0,1281	0,12988	0,13165	0,13343	0,13521	0,13698	0,13876	82°
8°	0,14054	0,14232	0,1441	0,14588	0,14767	0,14945	0,15124	0,15302	0,15481	0,1566	81°
9°	0,15838	0,16017	0,16196	0,16376	0,16555	0,16734	0,16914	0,17093	0,17273	0,17453	80°
10°	0,17633	0,17813	0,17993	0,18173	0,18353	0,18534	0,18714	0,18895	0,19076	0,19257	79°
81°	6,31375	6,38587	6,45961	6,53503	6,61219	6,69116	6,77199	6,85475	6,93952	7,02637	8°
82°	7,11537	7,20661	7,30018	7,39616	7,49465	7,59575	7,69957	7,80622	7,91582	8,02848	7°
83°	8,14435	8,26355	8,38625	8,51259	8,64275	8,77689	8,9152	9,05789	9,20516	9,35724	6°
84°	9,51436	9,6768	9,84482	10,0187	10,1988	10,3854	10,5789	10,7797	10,9882	11,2048	5°
85°	11,4301	11,6645	11,9087	12,1632	12,4288	12,7062	12,9962	13,2996	13,6174	13,9507	4°
86°	14,3007	14,6685	15,0557	15,4638	15,8945	16,3499	16,8319	17,3432	17,8863	18,4645	3°
88°	28,6363	30,1446	31,8205	33,6935	35,8006	38,1885	40,9174	44,0661	47,7395	52,0807	1°
89°	57,29	63,6567	71,6151	81,847	95,4895	114,589	143,237	190,984	286,478	572,957	0°
	54'	48'	42'	36'	30'	24'	18'	12'	6'	0'	
КОТАНГЕНСИ											

*бінарний урок: інформатика–математика²³,
проводиться в кабінеті ОІОТ*

тема: Математична постановка задачі. Етапи розв'язування задач з допомогою комп'ютера.

мета: Розглянути питання про моделювання; Вивчити поняття інформаційної та математичної моделі; Сформувати навички математичної постановки задачі.

обладнання та література:

²³ Домінантні елементи (інформатика) виділено суцільною рамкою, підпорядковані елементи (математика) виділено пунктирною рамкою.

- І.Т. Зарецька, А.М. Гуржій, О.Ю. Соколов, Інформатика, частина II, підручник для 10-11 класу загальноосвітніх навчальних закладів, Київ, "Форум", 2004 р.;
- *Microsoft Office Word 2003, Microsoft Office Excel 2003.*
- Таблиця "Характеристики величин"

ПЛАН УРОКУ

1. Повторення та актуалізація

а) Розв'язати у зошиті поваріантно задачі:

I варіант

Натрапивши на капустяне поле, що складається з 40 рядків, у кожному з яких росте по 200 капустин, Заць вирішив влаштувати бенкет для своїх Вухатих друзів. Скільки Вухатих друзів зможе запросити Заць на бенкет, якщо за ніч вони з'їдають по 16 качанів капусти?

Розв'язання:

$$40 * 200 / 16 = 500.$$

Відповідь: 500 Вухатих друзів

Висновок: Задачі ідентичні по змісту, бо обидві розв'язуються за формулою $S=a*b$, але друга додатково потребує позначення даних (аргументів) та результату. Узагальнення умови подібних задач, необхідне для їх розв'язування, називають *математичною постановкою*, яка складається з двох етапів:

- 1) опис параметрів (аргументів та результатів);
- 2) побудова математичної моделі.

II варіант.

За даними довжиною і шириною прямокутника визначити його площу. Обчислити, якщо довжина дорівнює 40 см, ширина дорівнює 25 см.

Розв'язання:

Позначимо довжину прямокутника a , ширину b , площу S .

$$S = a * b, \text{ при } a = 40, b = 25$$

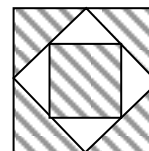
$$S = 40 * 25 = 1000 \text{ (см}^2\text{)}.$$

Відповідь: 1000 см²

б) За даним зразком у текстовому документі Word побудувати поваріантно малюнки при даних n – кількість сторін правильного многокутника та k - кількість вписаних многокутників:

I варіант: $n = 5; k = 3.$

II варіант: $n = 4; k = 5.$



в) Користуючись Microsoft Excel, самостійно заповнити таблицю:

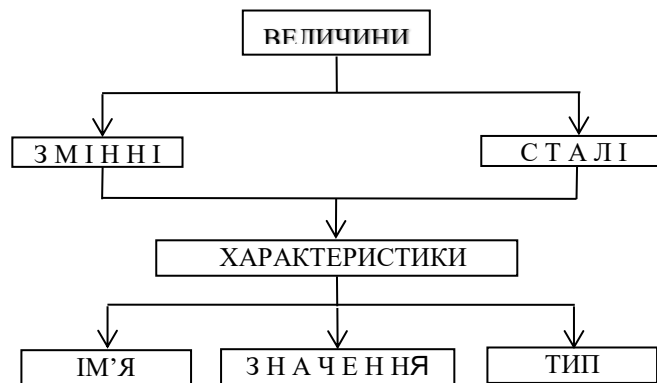
a	b	S
2	12	
4	3	
6	-6	
8	-15	
10	-24	
12	-33	

2. Вивчення нового матеріалу

Під *моделлю* прийнято розуміти *точну або подібну копію деякого об'єкта*, яку можна взяти за зразок, наприклад, модель (марку) мобільного телефону, зменшену, але діючу копію літака тощо.

Інформаційною моделлю будемо вважати *точний опис* з допомогою тексту або аудіо-візуальних засобів *деякого матеріального чи абстрактного об'єкта або явища*, тобто *повну інформацію* про нього.

Кожна модель має *параметри*, тобто *величини*, що виражають її форму, розміри і т.д. Величини діляться на постійні (константи) та змінні і мають три характеристики: *ім'я* (інакше - позначення або ідентифікатор), *тип* та *значення*.



Математичною моделлю називається *опис деякого об'єкта чи явища засобами математики*, тобто у вигляді формул.

Але для запису формул необхідно встановити, тобто описати величини, які будуть даними (*аргументами*), допоміжними та шуканими (*результатами*). Як було сказано в п.1, опис величин та побудова математичної моделі називається *математичною постановкою задачі*. Математична постановка – це перший етап розв'язування задачі, що передуює визначенню її результатів. На уроках математики, фізики та інших предметів визначення результатів виконують особисто, тобто вручну, інколи для проміжних обчислень використовуючи калькулятор чи якусь спеціальну комп'ютерну програму, наприклад *Microsoft Excel*.

Існують задачі, що передбачають автоматизацію вводу аргументів, обчислень та виводу результатів через їх дуже велику кількість, тому розв'язуються з допомогою комп'ютера в автоматичному режимі. Процес розв'язування таких задач складається з кількох етапів, зокрема:

- a) Виконання математичної постановки задачі;
- b) Опису, дослідження та доведення алгоритму;
- c) Складання та тестування (перевірка правильності) програми;
- d) Виконання програми і одержання результатів (відпові задачі).

Після математичної постановки задачі у “ручному” режимі виконуються необхідні обчислення і одержуються результати. При розв'язуванні задачі з допомогою комп'ютера в автоматичному режимі потреби обчислювати особисто немає, замість цього слід виконати п. b) - d).

3. Закріплення

Приклад 1 (Сума) Обчислити суму перших 2500 натуральних чисел. Виконати математичну постановку розв'язання цієї задачі для n перших натуральних чисел.

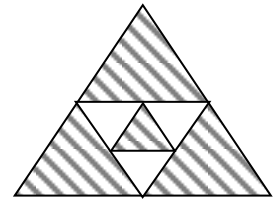
Уявивши вираз для обчислення суми перших 2500 натуральних чисел: $1+2+3+4+5+\dots+99+100+101+\dots+1000+1001+\dots+2499+2500$, та час, потрібний для обчислення вручну, доведеться погодитись, що розв'язування цієї задачі слід почати з другої частини, тобто з математичної постановки.

Математична постановка:

- *Опис величин.* Нехай n – кількість доданків, а S_n – їх сума. Обидві величини повинні бути натуральними числами.
- *Математична модель.* Записавши шукану суму та підмітивши, що суми доданків, однаково віддалених від кінців, рівні між собою, а їх кількість у двічі менша кількості доданків, одержимо: $S_{2500} = (1+2500) + (2+2499) + \dots + (1250+1251)$, після узагальнення, одержимо математичну модель: $S_n = (1+n) \cdot n/2$.

Обчислення. При $n = 2500$ $S_n = (1+2500) \cdot 2500/2 = 3126250$.

Приклад 2 (Фігура) Виконати математичну постановку задачі обчислення площі заштрихованої фігури, що складається з n вписаних правильних трикутників, якщо сторона зовнішнього трикутника a і заштриховано трикутники з непарними номерами (на малюнку 1 $n = 3$)



Математична постановка:

- *Опис величин.* Нехай n – кількість доданків, a – сторона зовнішнього правильного трикутника, а s_n – площа заштрихованої фігури. Додатково позначимо площу зовнішнього трикутника s_0 . На відміну від попередніх прикладів, якщо n – натуральне, то a , s_n та s_0 – повинні бути дійсними числами.
- *Математична модель.* Легко помітити, що площа вписаного трикутника завжди в четверо менша за площу описаного трикутника, тому: $s_n = s_0 - s_0/4 + s_0/16 - s_0/64 + s_0/256 - \dots$. Але ця формула з двома суттєвими недоліками: у виразі відсутній параметр n , вона має два варіанти – для парних і непарних n , крім того, s_0 слід виразити через a , тобто її слід узагальнити. Для цього достатньо помітити, що доданки становлять геометричну прогресію з першим членом s_0 та знаменником, рівним $-1/4$. Отже:

$$s_n = \frac{s_0 \left(\left(-\frac{1}{4} \right)^n - 1 \right)}{-\frac{1}{4} - 1}, \text{ при } s_0 = \frac{a^2 \sqrt{3}}{4}.$$

Тому остаточний вигляд формули такий: $s_n = \frac{a^2 \sqrt{3} \left(\left(-\frac{1}{4} \right)^n - 1 \right)}{5}$.

Наведені формули – це два варіанти математичної моделі даної задачі. Але завжди слід визначати остаточний, найбільш загальний варіант, тому математичною моделлю задачі будемо другу формулу.

I. Усні питання:

- I.1 Чи можна вважати інформаційну модель математичною моделлю?
- I.2 Чи можна вважати математичну модель інформаційною моделлю?
- I.3 Яка різниця між інформаційною та математичною моделями?
- I.4 Чи є в прикладах II.1-II.3 допоміжні величини, тобто ті, які не можна назвати ні аргументами, ні результатами?
- I.5 Чи можна на власний розсуд вказувати типи величин, виконуючи математичну постановку задачі?
- I.6 Чи можуть аргументи бути константами? змінними?
- I.7 Чи можуть результати бути константами? змінними?
- I.8 Чи можуть допоміжні величини бути константами? змінними?

II. *Самостійне завдання (у зошитах)*: За даним радіусом кола визначити його довжину та площу відповідного круга. Виконати математичну постановку та для даного значення аргумента обчислити результати.

4. Домашнє завдання

- 1) Повторити послідовності, арифметичну та геометричну прогресії;
- 2) Вивчити поняття математичної моделі, математичну постановку задачі;
- 3) Виконати завдання:

Завдання 1 Обчислити кількість усіх діагоналей для опуклого n -кутника при даному n . Виконати математичну постановку розв'язання задачі.

Завдання 2 Виконати математичну постановку задачі обчислення площі заштрихованої фігури, що складається з n вписаних квадратів, якщо сторона зовнішнього квадрата a і заштриховано квадрати з непарними номерами (на малюнку 2 $n = 3$)

V.4 Зв'язки інформатики з виховною роботою.

Навчальний процес нерозривний з виховною роботою, думка про можливість "чистого" виховання помилкова. Більше того, воно, як самоціль, не має сенсу, бо сучасне суспільство чим далі стає все більш прагматичним. Часи, коли ідеологічна складова у формуванні особистості була вирішальною, минули. На їх зміну прийшла нова, постромантична епоха. Сучасні школярі, поряд із байдужістю та інфантильністю, дуже амбіційні й практичні. Тому не варто сподіватись, що гуманізація, демократизація чи лібералізація навчально-виховного процесу створить нове, сучасне покоління. У наш час свідомість людей об'єктивно дрейфує від моральних законів та філософських категорій зразка "совість" до відвертого визнання жорстоких законів природи зразка

”боротьби за виживання”. Це означає, що знання та практичні навички, які покликані приносити матеріальну користь у майбутньому, сьогодні більш цінні, ніж порядність і вихованість. Констатувати це нелегко, але факти – річ незаперечна.

Така сувора, але справедлива реальність приводять до висновків:

- 1) Навчання без виховання не ефективне.
- 2) Виховання – це сукупність прийомів, які полегшують навчання та позитивне ставлення до праці.
- 3) Виховання можливе тільки у поєднанні з навчанням та працею.

Погодившись із такими висновками, слід задуматись на суттю та масштабами виховного компонента у навчально-виховному процесі. Адже і в минулі часи не приносило успіху на уроках розв’язування задач виховного характеру. Сьогоднішнім учням слід пропонувати завдання, які розвивають компетентність, впевненість, практицизм і водночас привчають до оптимізму та реалізму.

Якщо бути відвертим і чесним, то дванадцятибальна система оцінювання мало змінила мотивацію до навчання, вона подібна інфляції, коли коштів більше, але варті вони менше. Учні не стали відноситись до зароблених балів, як до досягнень, які можна примножити, вони розуміють, коли ставлять 3 бали, треба випросити ще один бал, щоб вискочити з небезпечної зони, адже 1-3 бали – це та ж колишня двійка.

Як же перевести учнів на ”госпрозрахунок”, щоб вони самостійно давали собі реальну оцінку, прогнозували успіхи та планували їх збільшення? Можливостей дуже багато, корисними в цьому можуть бути ІКТ та інформатика. У *Excel* легко створити таку таблицю:

	A	B	C	D	E	F	G
1				алгебра			
2	прізвище	уроки	пропуски	здібності	робота на уроці	робота дома	якість навчання
3	Бережна В.	99	37	7	0,7	0,7	2,1
4	Васильєва А.	99	2	9	0,7	0,8	4,9
5	Гелевешко І.	99	8	8	0,8	0,8	4,7
6	Дерев'янчук Г.	99	6	8	0,8	0,8	4,8
7	Кавпеченко Т.	99	25	8	0,5	0,7	2,1
8	середнє значення	99	15,6	8	0,7	0,76	3,7

Ця таблиця на прикладі предмета ”інформатика” за формулою $(B4-C4)/B4 * D4 * E4 * F4$ визначає показник якості навчання. Цілком зрозуміло, що якість навчання прямо пропорційна числовим значенням $(B4-C4)/B4$, $D4$, $E4$ та $F4$, де $(B4-C4)/B4$ – відношення кількості відвіданих уроків до кількості проведених уроків (виставляється із класного журналу), $D4$ – значення у

процентах здібностей до вибраного предмету, *E4* та *F4* – процентні показники якості домашньої самопідготовки та робочої активності під час уроків.

Два останні параметри виставляються учителем, здібності бажано, щоб об'єктивно і самокритично виставив собі кожен учень. Тоді одержана якість навчання буде цілком легітимною.

Деякі параметри у певній мірі спірні, у першу чергу це стосується показника здібностей. Прийнято вважати, що всі учні здібні, але чи безсумнівне таке твердження? Адже, якщо бути чесним і об'єктивним, то для будь-кого можна встановити, наприклад, у 12-бальній системі його здібності, іншими словами, потенціал до вибраного предмету. Наприклад, загальнонавчаний геній О.С. Пушкін не мав здібностей до математики. Не існує інструментів для абсолютного встановлення здібностей. Це стосується і відомого коефіцієнта IQ. Крім того, здібності можуть бути з певних причин приховані або розкриті лише частково. Нарешті, здібності можуть розвиватись та удосконалюватись, бо їх складовими є пам'ять, абстрагування, інтуїція, швидкість оперування і т.д.

Для точнішого встановлення якості навчання можна вводити додаткові параметри чи удосконалювати формулу. Але з допомогою *Excel* можна не тільки здійснювати встановлення якості навчання, а й прогнозувати та планувати її. А це може оперативно використовуватись для зведення статистичних даних про рівень навчального процесу, допомагати учневі, класному керівникові, батькам та адміністрації школи, бути дієвим стимулятором у виховному процесі.

Проблеми виховання не планувалось вивчати на заняттях ОШППД, тому цей зразок застосування ІКТ є лише ілюстрацією розширення можливостей комп'ютерів за межі демонстраційних функцій, про що говорилося ще у першому розділі.

VI. Методична робота учителя інформатики.

Учительська робота полягає в організації навчання, тому кожен учитель постійно користується спеціальними галузями наукових знань – *методикою* навчання та *дидактикою*.

Методика – це система правил, виклад методів навчання чому-небудь або виконання якоїсь роботи. **Метод** – (від грец. *methodos* – шлях дослідження, теорія, вчення), *спосіб досягнення якої-небудь мети, розв'язання конкретної задачі, сукупність прийомів або операцій практичного чи теоретичного засвоєння (пізнання) дійсності.*

Дидактика (від грец. *didaktikós* – той, хто повчає) – це розділ педагогіки, теорія освіти і навчання. Вона описує закономірності засвоєння знань, умінь та навичок і формування переконань, визначає об'єм і структуру змісту освіти, удосконалює методи й організаційні форми навчання, виховний вплив навчального процесу на учнів.

Методика викладання основ інформатики розглядає мету, зміст, засоби, методи і організаційні форми викладання шкільного курсу інформатики. Слід зауважити, що результати сучасної і класичної дидактики конкретизуються у контексті викладання основ інформатики. І навпаки,

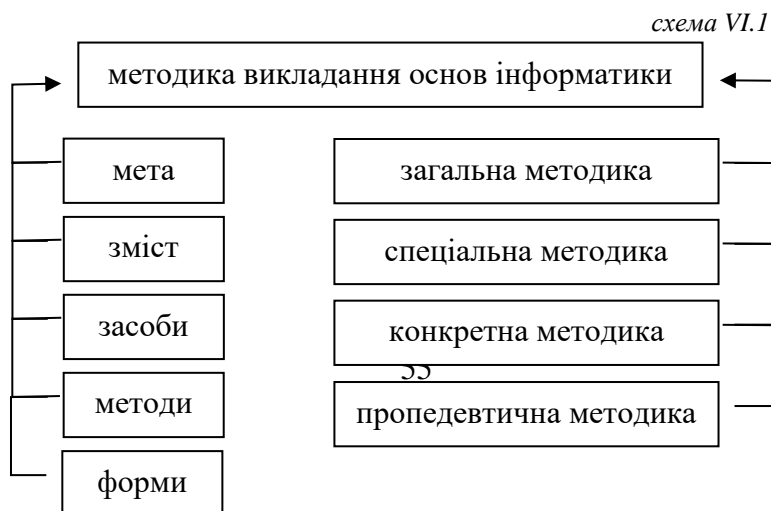
проходить осмислення конкретного досвіду, досягнень і проблем практики викладання в школі із позицій загальної дидактики (див. схему VI.1).

Нажаль, значна частина учителів та викладачів надто спрощено і буквально підходять до використання надбань методичної та дидактичної думки, тобто *ознайомлюються з фаховими періодичними виданнями, сліпо виконують інструкції* щодо календарно-тематичного планування, проведення уроків, обліку знань, умінь та навичок, застосування інтерактивних методів чи комп'ютерних технологій навчання. Але в жодній інструкції немає повних і відповідних конкретним умовам вказівок, як слід поступати у даному випадку. Тому *кожен учитель повинен бути в тій чи іншій мірі методистом*, уміти без інструкцій, творчо й самостійно вирішувати нагальні педагогічні та дидактичні завдання. Іншими словами, *у кожного учителя інформатики повинна бути чітко визначена система методичної роботи*. Вона повинна охоплювати загальні й спеціальні аспекти викладання предмету та конкретну методику. Необхідність активної методичної роботи учителя основ інформатики зумовлена ще й тим, що, як правило, у школі, районному відділі освіти та районному методичному кабінеті відсутні спеціалісти з методики викладання цього предмету.

Загальні методи викладання – це підходи до проведення практичних занять, застосування інтерактивних технологій навчання, міжпредметних зв'язків тощо. У цьому контексті на заняттях ОШППД ми, наприклад, розглядали концепцію так званого *безпаперового методу проведення уроків інформатики* (БМПУ), тобто застосування через локальну мережу електронних конспектів та сценаріїв уроків (ЕКУ та ЕСУ), електронних дидактичних засобів (ЕДЗ, у тому числі КЕДЗ), електронні форми проведення тестування, практичних і залікових робіт тощо.

Спеціальна методика викладання інформатики передбачає вироблення й застосування підходів до вивчення операційної системи *Windows*, мов програмування, алгоритмів тощо. Ми, зокрема, реалізуємо це у формулюванні конкретних проблем шкільної інформатики та пошуках шляхів їх подолання (згідно планів роботи ОШППД на 2007-2008 та 2008-2009 н.р.), у розробці так званого експрес-курсу алгоритмізації та програмування [6] і т.д.

Конкретна методика викладання основ інформатики складається з окремих питань загальної методики (наприклад, адаптація і розробка програм, планування уроків в окремих класах і з окремих тем), окремих питань спеціальної методики (наприклад, проведення уроків-семінарів, бінарних

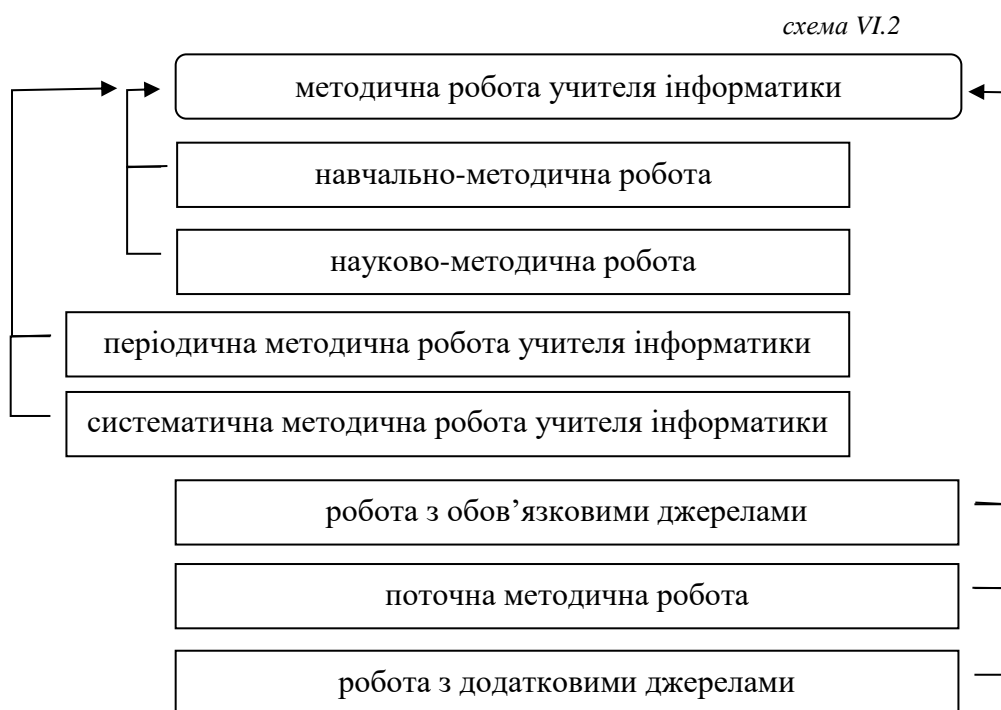


уроків, розв'язування олімпіадних задач на переборні методи і т.д.).

Слід розрізняти також методики викладання *пропедевтичних* (підготовчих) курсів ІКТ, комп'ютерної графіки, Web-дизайну тощо та *спеціальні методики*, наприклад, курсу алгоритмізації і програмування, або розв'язування олімпіадних задач з інформатики.

Враховуючи вищесказане, слід погодитись, що кожен учитель основ інформатики у системі власної методичної роботи повинен мати уявлення про її форми, обсяги і періодичність (циклограму).

Форми методичної роботи учителя основ інформатики можуть бути найрізноманітнішими. Вони залежать від рівня *освіти* (дипломований чи атестований), *навантаження* (основний спеціаліст чи сумісник), *досвіду роботи* (категорія, педстаж) тощо. Тому будемо мати на увазі найбільш важливі і загальні (схема VI.2).



1 Методична робота учителя основ інформатики починається з вибору програм і підручників та складання календарно-тематичних планів. Адже, не зважаючи на велику кількість базових програм для універсального і спеціалізованих профілів, факультативних і профільних занять, вибрати ту, яка в повному обсязі відповідає умовам конкретної школи, дуже важко. Враховуючи це, календарно-тематичне планування для кожного класу, універсального чи іншого профілю необхідно забезпечити відповідною пояснювальною запискою.

Значної і кваліфікованої методичної роботи вимагає також адаптація рекомендованих МОН України навчальних підручників до вимог діючих навчальних програм з основ інформатики. Враховуючи те, що підручники [2] місцями не відповідають діючим програмам[1], не зручні у користуванні (нумерація розділів, параграфів і пунктів, обсяг і розпорошеність навчального матеріалу і т.д.) та виходячи з концепції БМПУ бажано розробити серію дидактичних матеріалів у вигляді стислого викладу навчального матеріалу,

помістивши їх в окремі папки, наприклад, “Тези до уроків інформатики“. Подібним чином можна підготувати серії залікових та тестових завдань чи інші ЕДЗ.

Цей вид методичної роботи можна назвати *роботою з обов'язковими документами та джералами*.

2 Схожа за формою, але відмінна по змісту є робота з додатковими джерелами. По формі ця робота пов'язана із самоосвітою, але по суті – це одержання нової інформації з обов'язковим впровадженням у практику²⁴. Будь-які додаткові джерела необхідно вивчити, оцінити, узагальнити і адаптувати до конкретних умов (ідеться про узгодженням з психолого-педагогічними та віковими особливостями учнів тощо).

3 Створення саморобних паперових і електронних навчальних документів (дидактичних засобів) – це окремий вид методичної роботи учителя, інформатики у тому числі. Методична складова такого виду роботи полягає у створенні інструкцій використання дидактичних засобів та в плануванні їх застосування. Близькою за змістом і формою є робота над створенням та впровадженням тестових та контролюючих засобів. Для учителя інформатики – це, як правило, електронні засоби. Описаний вид методичної роботи можна назвати *поточною*.

4 Описані вище форми методичної роботи можна об'єднати під загальною назвою – *систематичні*, тобто такі, що виконуються повсякденно, безперервно. Але необхідні також окремі види методичної роботи, які проводяться лише один раз, чи багаторазово, навіть з деяким періодом. Учитель інформатики може написати фахову або методичну статтю, курсову роботу під час навчання на курсах, методичний посібник для учителів, або навчальний посібник для учнів. Кожні п'ять років ми готуємо так звану “папку“ для чергової атестації, по сучасному – портфолію, куди “вкладаємо“ методичні матеріали різного характеру, змісту і призначення. Це своєрідна збірка власних методичних матеріалів.

Класифікацію методичної роботи учителя інформатики можна продовжити, але вона, залежно від педстажу, досвіду та напрямків роботи може бути більш чи менш широкою і глибокою, тому замість єдиного реєстру тут пропонується спрощений орієнтовний перелік.

5 І все ж, враховуючи не стільки обов'язковий, як творчий характер методичної роботи, слід погодитись, що у кожного учителя повинна бути своя близька чи улюблена тема, можливо навіть на рівні своєрідного “хобі“. Наприклад, одні учителі більш цікавляться вивченням комп'ютерних мереж, баз даних, Web-дизайну чи комп'ютерної графіки, іншим ближча позакласна робота з обдарованими учнями, олімпіадні задачі, робота в МАН, або об'єктно-орієнтоване програмування. Лише на такому ґрунті можуть виникати серйозні методичні розвідки чи завершені навчально-методичні роботи. Це галузь

²⁴ Слід розрізняти фахову і методичну самоосвіту. Останню можна вважати інтравертивною методичною роботою. Тут же ідеться про екстравертивну методичну роботу, тобто створення та впровадження *методів і організаційних форм* викладання шкільного курсу інформатики.

науково-методичної роботи. У якості прикладів можна згадати теми, розглянуті на заняттях ОШППД:

- Якою бути шкільній інформатиці завтра.
- Письмовий опис алгоритмів як невід’ємна складова розв’язку задач із програмування в школі.
- Метод динамічного програмування у школі.
- Алгоритмічні підходи до розв’язування задач “довгої арифметики”.
- Від задачі до задачі.
- КЕДЗ в системі ІТН.
- Роздуми про проблеми шкільних олімпіад з інформатики.

У минулому навчальному році, розглядаючи метод проектів, ми виділяли окремий вид – *індивідуальний проект*. Таким може бути робота над олімпіадною задачею. Після лекції планується проведення практичного заняття по створенню проекту “Олімпіадна задача з інформатики”.

Отже, класифікацію методичної роботи можна представити так:

1. Методична робота з обов’язковими документами та джерелами (освітніми стандартами, програмами, підручниками, інструкціями, інструктивно-методичними записками, календарно-тематичне планування тощо);
2. Методична робота з додатковими та спеціальними документами і джерелами (вивчення і впровадження в навчально-виховний процес нових форм і методів навчання та виховання);
3. Поточна методична робота (підготовка поточних методичних та дидактичних засобів навчання);
4. Періодична методична робота (опис і узагальнення власного та вивчення і впровадження запозиченого методичного досвіду);
5. Науково-методична робота.

VI.1 Робота з дидактичними матеріалами як складова методичної роботи.

У пункті II.2 наведено приклад електронного конспекту уроку та розглянуто етапи розробки й використання таких електронних дидактичних засобів (ЕДЗ). Треба зауважити, що набагато ефективніше використовувати серію ЕКУ, яка охоплює цілу тему. Такі матеріали вище називались комплексними електронними дидактичними засобами (КЕДЗ). Але їх створення вимагає значних затрат часу, тому доцільно об’єднувати процеси створення та використання.

Дійсно, легко помітити що електронний конспект уроку – це своєрідний модуль, який складається з окремих, цілком автономних блоків, записаних у додатках або окремих файлах і поєднаних гіперпосиланнями. Наведемо перелік блоків згаданого вище прикладу:

- а) План уроку (ядро);
- б) Тема, мета, обладнання, література (додаток 1);
- в) Таблиця ”Структура програми на *PasABC*” (додаток 2);
- г) Таблиця ”Оголошення і опис величин” (додаток 3);
- д) Таблиця ”Команда вводу” (додаток 4);
- е) Таблиця ”Команда присвоєння” (додаток 5);
- ж) Таблиця ”Команда виводу” (додаток 6);

- з) Вікно програми на *PasABC* (додаток 7);
- и) Завдання до практичної роботи (додаток 8).

Саме ця обставина дозволяє об'єднувати процеси створення та використання ЕКУ. Поки він не створений, учитель може використати звичайний поурочний план, а перераховані вище блоки застосувати роздільно, як невеликі електронні дидактичні засоби.

Для зразка нижче наведено електронні заготовки до уроку на тему "Операційна система Windows. Об'єкти файлової системи та дерево каталогів. Навігація в пам'яті комп'ютера".

ОПЕРАЦІЙНА СИСТЕМА WINDOWS

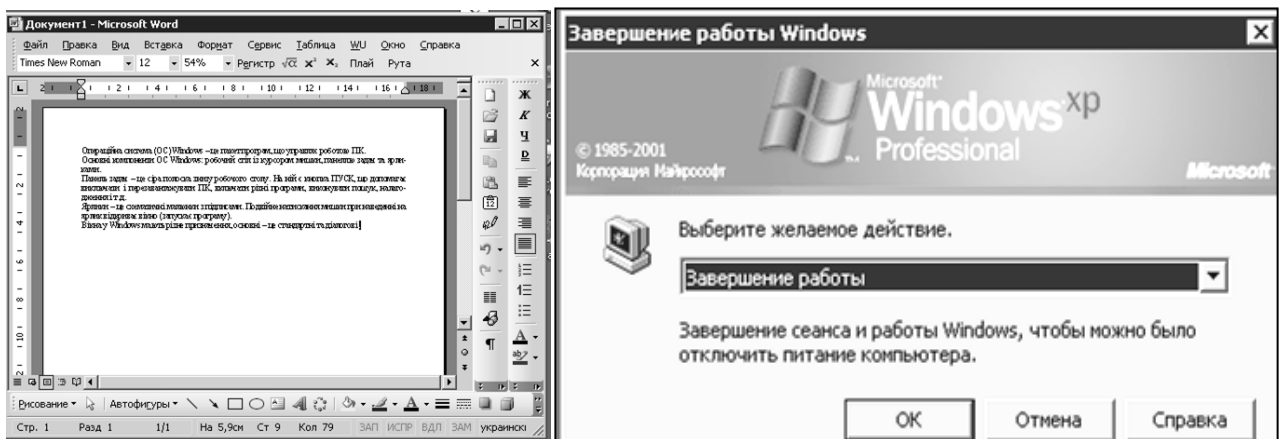
Операційна система (ОС) Windows – це пакет програм, що управляють роботою ПК.

Основні візуальні компоненти ОС Windows: робочий стіл із курсором мишки, панеллю задач та ярликами.

Панель задач – це сіра полоса знизу робочого столу. На ній є кнопка **ПУСК**, що допомагає виключати і перезавантажувати ПК, включати різні програми, виконувати пошук, налагодження і т.д.

Ярлики – це схематичні малюнки з підписами. Подвійне натискання мишки при наведенні на ярлик *відкриває вікно (запускає програму)*.

Вікна у Windows мають різне призначення, основні – це *стандартні та діалогові*.



ОБ'ЄКТИ ФАЙЛОВОЇ СИСТЕМИ ТА ДЕРЕВО КАТАЛОГІВ (10)

Файл – це логічно поєднана порція інформації, яка під деяким іменем збережена на носії пам'яті ПК (диску).

Ім'я файлу – це набір допустимих символів.

приклади: Іваненко.doc; Іваненко.bmp

Чорний Д.doc; Чорний Д.bmp

Windows.exe – виконувані файли, програми.

недопустимі символи: “.”; “.”;

Останні символи імені файлу, які починаються “.” називаються розширенням імені файлу. Розширення пишеться латиницею.

Каталог (папка, директорій) – об’єкт, збережений в пам’яті ПК під деяким іменем, і містить папки (підкаталоги) та файли.

Каталог, що містить всередині інші каталоги називається старшим каталогом, найстарший каталог на диску чи в пристрої називається кореневим каталогом і позначається “\”.

приклад: D: \ klass \ 7_a \ Group_1.

Папка Group_1 є підкаталогом папки 7_a, яка в свою чергу міститься в папці klass, що є підкаталогом кореневого каталогу.

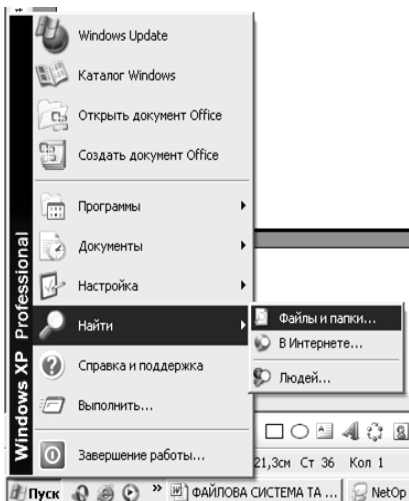
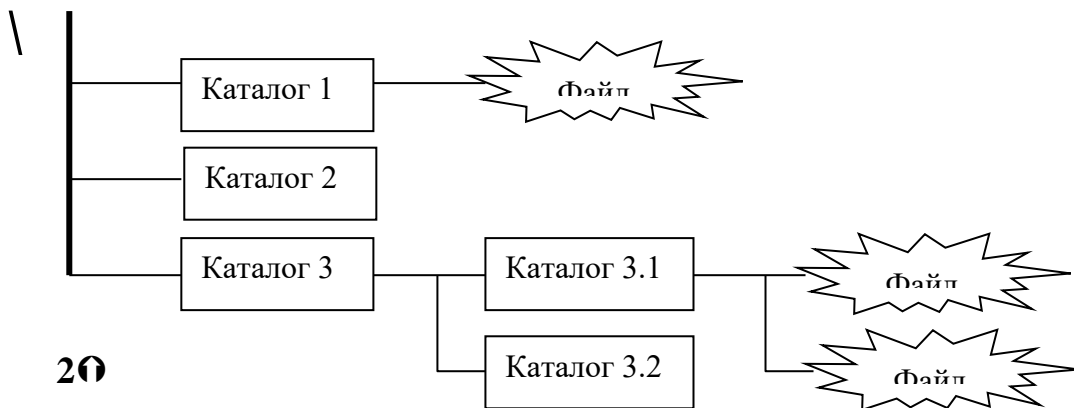
Дерево каталогів 20 – це система каталогів на диску.

НАВІГАЦІЯ В ПАМ’ЯТІ КОМП’ЮТЕРА (ПК) 30

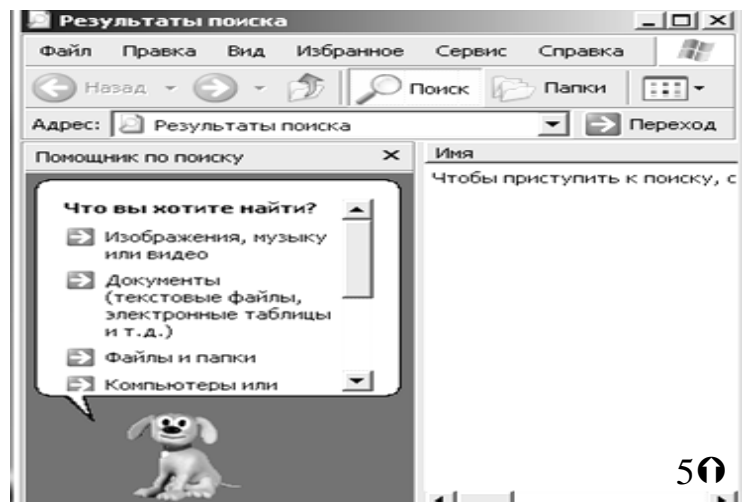
Під навігацією розуміють “подорож” по пам’яті комп’ютера, тобто переходи з диска на диск, відкриття папок та вихід з них, створення, копіювання і вставка файлів чи папок, а також пошук 40

Пошук файлів і папок. Щоб знайти потрібні файл чи папку, треба виконати команду НАЙТИ і інші дії у вікні “Результаты поиска” 50. Команду НАЙТИ можна виконати з допомогою панелі задач, або з допомогою ПРОВОДНИКА 60.

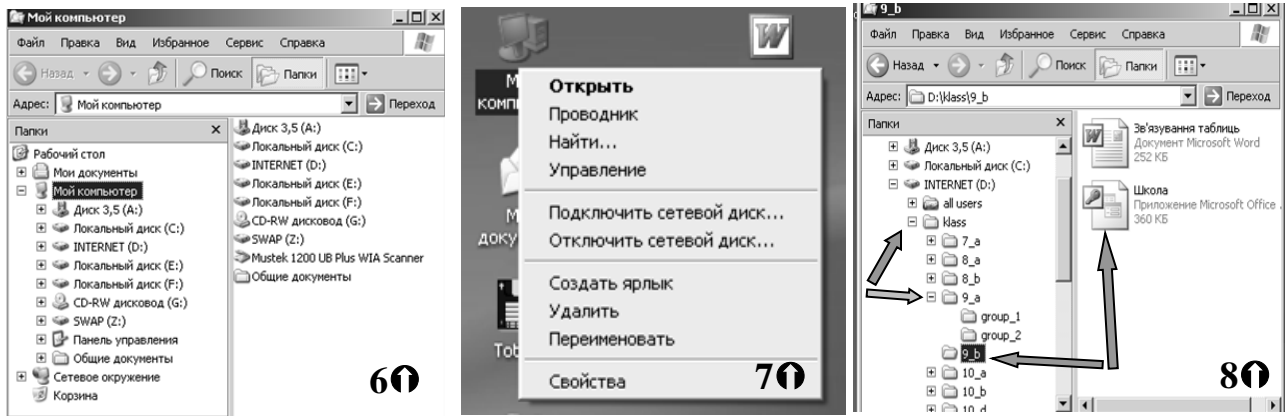
ПРОВОДНИК 70 – це вікно, яке практично співпадає з вікном МОЙ КОМП’ЮТЕР, але зліва має панель “Папки” 80, що відображає дерево каталогів активного диска чи пристрою.



40



50



Домашнє завдання: § 7, § 9-10, повт. § 6

ПИТАННЯ ДО ЗАЛІКУ:

1. Файлова система та дерево каталогів: 1 Ⓜ
 - Поняття файлу. Ім'я та розширення імені файлу. Шлях до файлу. Повне ім'я файлу. Виконувані файли;
 - Дерево каталогів. Кореневий каталог. Підкаталоги;
2. Створення, копіювання, переміщення, перейменування та стирання файлів і каталогів. Вхід в каталог і вихід з нього. Відкриття та закриття файлів.
3. Навігація в пам'яті ПК. Провідник:
 - Пошук файлів і папок;
 - Знаки “ * “ та “ ? “ в іменах файлів і папок; 3 Ⓜ

Ці заготовки електронних дидактичних засобів займають лише дві сторінки, містять великий обсяг навчального матеріалу. Для зручності користування вони зв'язані тимчасовими гіперпосиланнями. Ними можна вигідно замінити підручник, адже подібна тема у підручнику, як правило, займає у 5-10 разів більше сторінок. З часом такі заготовки можуть бути матеріалом для конструювання ЕКУ або КЕДЗ.

VII. Особливості викладання основ інформатики в 9-у класі.

2009-10 н.р особливий тим, що вперше введено навчальний предмет "Інформатика" в дев'яті класи. Це пов'язано не тільки з переходом на 12-річне навчання, а й із зміною концепції вивчення ІКТ в загальноосвітній школі. У зв'язку з цим введена нова програма [2] та кілька альтернативних підручників [18], [19], [20]. Поверховий погляд на ці джерела не виявляє нічого особливого: ті ж вступні теми, традиційний порядок вивчення інформатики на першому році вивчення інформатики, приблизно такий же розподіл навчальних годин. Правда, підсилено присвячений операційній системі розділ, збільшено обсяг навчального матеріалу по темі "Комп'ютерні мережі", широко представлено тему "Комп'ютерна графіка", яку до цього вивчали в 10-11 класах оглядово.

Але ще при першому, загальному, знайомстві з новим курсом виникає цілий ряд суттєвих питань. Ознайомлення з критеріями оцінювання навчальних досягнень учнів викликає сумнів, що за діючою програмою, при встановленій

кількості навчальних годин та з допомогою рекомендованих МОН України підручників можливо досягти високого рівня навчальних досягнень, адже він передбачає, щоб знання, вміння і навички учня відповідали вимогам державної програми *у повному обсязі*.

Щоб переконатись у цьому, досить виписати всі поняття та їхні особливості, які відносяться до окремо вибраної теми та зіставити їх із кількістю відведених програмою навчальних годин. Крім того слід враховувати вікові особливості учнів 9 класу, їх уміння визначати й опрацьовувати навчальну, та сприймати наукову інформацію. Для прикладу візьмемо тему "Поняття файлу, каталогу. Ім'я файлу та каталогу, розширення імені файлу. Імена зовнішніх запам'ятовуючих пристроїв, шлях до файлу", яка є частиною теми "Системне програмне забезпечення" (додаток 1). Протягом одного уроку необхідно вивчити та закріпити достатньо абстрактні поняття (див. програму [2]):

- файлу;
- імені файлу;
- розширення імені файлу;
- повного імені файлу;
- каталогу;
- кореневого каталогу;
- підкаталогу;
- старшого каталогу;
- дерева каталогів;
- зовнішнього запам'ятовуючого пристрою.

Особливістю вивчення основ інформатики в умовах практичної форми проведення навчальних занять є необхідність унаочнити перераховані поняття з допомогою комп'ютера. В умовах групових занять це вимагає відведення 3-5 хв. на кожне з них, тобто всього 30-45 хв.

У запропонованому календарно-тематичному плані з інформатики для 9 класу відповідно до діючої програми вивчення названої теми включено в урок № 6 "Загальні відомості про системне, службове та прикладне програмне забезпечення. Класифікація, основні функції та складові операційних систем. Поняття про ядро операційної системи, інтерфейс користувача, драйвери та утиліти. Різновиди інтерфейсу користувача. *Поняття файлової системи, відмінності між поширеними файловими системами. Поняття файлу, каталогу. Ім'я файлу та каталогу, розширення імені файлу. Імена зовнішніх запам'ятовуючих пристроїв, шлях до файлу*", отже, якщо враховувати важливість і ємкість понять першої частини теми "Загальні відомості про системне, службове та прикладне програмне забезпечення. Класифікація, основні функції та складові операційних систем. Поняття про ядро операційної системи, інтерфейс користувача, драйвери та утиліти. Різновиди інтерфейсу користувача" на вивчення другої частини теми можна відвести не більше 50% часу уроку, тобто до 25 хв. Проведений аналіз переконує, що *вивчення вищеназваної теми в умовах плану і програми можливе лише в оглядовому*

вигляді, що недопустимо. Відомо, що освіта – це те, що залишається після навчання. Можливо однією з причин низького рівня сучасної освіти є перенасиченість програм та підручників з одного боку та недостатня кількість відведених програмою навчальних годин з іншого? Якщо це так, то, мабуть, варто звузити програмний матеріал і вивчати лише те, що обов'язково повинно залишитись після навчання.

У діючій програмі існує, на наш погляд, чимало "вузьких місць". Звернемо увагу на відсутність чіткої послідовності вивчення понять та вироблення умінь і навичок. Щоб побачити їх, досить ще раз подивимось на тему "Системне програмне забезпечення", на вивчення якої відводиться 7 годин. Тема включає три практичні роботи²⁵. "Розкиданість" навчального матеріалу теми видно, як кажуть, неозброєним оком.

МОН України рекомендує для вивчення інформатики в 9 класі 4 підручники різних авторських колективів. Наявність широкого вибору підручників безперечно є позитивним фактом. Але слід погодитись, що кожен авторський колектив має право на власну інтерпретацію курсу, а в умовах існування недоліків програми, про що сказано вище, різні підручники в більшій чи меншій мірі, але обов'язково спотворюють курс у цілому. Наприклад, у підручнику [18] поняття інтерфейсу користувача ОС *Windows* та об'єктів файлової системи описано коротко й чітко:

Інтерфейс користувача – сукупність засобів, які забезпечують обмін даними між користувачем і ОС.

Об'єкти файлової системи: папки, файли, ярлики.

Файлова система – структура збереження даних на зовнішніх носіях і сукупність програм, які забезпечують роботу з цією структурою. Як правило, операційна система може працювати з кількома файловими системами ([18], стор.77).

У [20] ці ж поняття подано поширено й описово:

Інтерфейс (діалог, обмін інформацією) між користувачем та комп'ютером.

WIMP-інтерфейс (*Window* – вікно, *Image* – образ, *Menu* – меню, *Pointer* - указівник), або **графічний інтерфейс**.

Використовуючи цей вид інтерфейсу, команди комп'ютеру надають за допомогою графічних образів (меню, вікон та інших елементів). Саме цей вид інтерфейсу зреалізовано в ОС *Windows*.

Об'єкти файлової системи перераховуються так: папки, файли і інші об'єкти.

Файлова система – частина будь-якої операційної системи, що відповідає за організацію зберігання та доступу до інформації на будь-яких її носіях ([20], стор. 129).

Спроба оптимальної підходу до опису інтерфейсу користувача у [19] (п.5.6, стор. 54): вступне пояснення, означення, графічна ілюстрація, детальний опис. Проте, чіткості, як у [18], чи повноти, як у [20] немає. Ось означення:

Інтерфейс користувача – це сукупність правил і засобів, застосовуючи які користувач може розв'язувати задачі за допомогою комп'ютера.

²⁵ Для зручності перша практична робота та відповідний їй матеріал виділено підкресленим курсивом, друга практична робота з відповідним матеріалом виділено курсивом.

Слово "правил" підкреслено, адже воно лише заплутує учня, на наш погляд, досить обійтись терміном "засобів", як у [18]. Завдання інтерфейсу користувача – забезпечувати обмін інформацією між користувачем та ОС, а не розпливчате "розв'язування задач", як у [19].

Ілюстрація поняття "командний рядок" (мал.5.4) відноситься до *MS DOS*, тому неактуальна, а "графічний інтерфейс" описано формально і не наочно.

У п. 6.4 перераховано та описано об'єкти операційної системи: робочий стіл, програми, папки, документи, ярлики, панель завдань, кнопка ПУСК, головне меню системи, кошик та вказівник мишки. Але поняття об'єкти файлової системи, передбачене програмою у темі 3 "Системне програмне забезпечення", у [19] розглядається п.17 розділу 5.

У розділі VI (Методична робота учителя інформатики) наведено приклад заготовки для електронного конспекту уроку "Операційна система Windows. Об'єкти файлової системи та дерево каталогів. Навігація в пам'яті комп'ютера", де зовсім відсутній опис файлової системи, а мова йдеться лише про об'єкти файлової системи, зокрема про файли та каталоги. На наш погляд, для учнів 9 класу такої інформації цілком досить, адже вони все одно не зможуть повністю усвідомити терміни зразка FAT32 та NTFS.

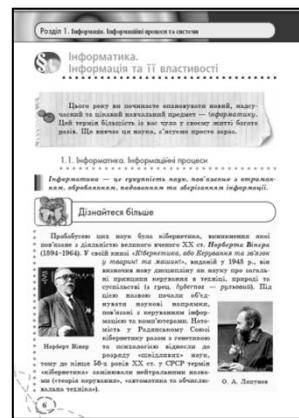
Тут ми розглядаємо питання конкретної методики, але без урахування загальних методологічних засад заглибитись у конкретику не можливо. Прикро, але авторські колективи нових підручників змушені не тільки балансувати між своїм баченням предмета та вимогами державних стандартів, освітніх концепцій і програм, а ще часто обмежені умовами участі в конкурсі підручників, необхідністю виграти тендер тощо. Підручник – це не засіб колективного навчання, він призначений для допомоги при індивідуальному опануванні навчальним матеріалом, навіть якщо клас використовує підручник на уроці під керівництвом учителя. Тому він не повинен бути переобтяженим, наприклад, елементами інтерактивного навчання, які належать до методики і можуть бути присутніми у методичному посібнику для учителя. Чи знизилась би якість підручника [19], якби завдання 1: "Наведіть приклади інформаційних систем. Складіть список п'яти інформаційних систем" було не в рубриці "Працюємо в парах" (стор. 32)? Те ж питання можна поставити щодо завдання 1 рубрики "Працюємо самостійно": "Поясніть кожний із термінів, які використовують для тлумачення комп'ютера як програмно-керованого пристрою, призначеного для автоматизованого опрацювання даних" (там же). У підручнику повинні бути питання та завдання, призначені для учнів. Їх можна поділити на простіші та складніші. Але на цьому й закінчується функція авторів, як використати питання та завдання нехай "болить голова" викладача, адже це вже методика. Автори були б "такими ласкавими", якби до свого підручника додали методичний посібник. Звідси напрашується закономірний висновок: замість кількох альтернативних підручників МОН України слід фінансувати один підручник, і обов'язковий до нього методичний посібник. Якщо знайдеться інший авторський колектив, який запропонує альтернативу,

нехай приймає участь у конкурсі за власний рахунок і буде винагороджений, коли виграє його. Підручник виданий державним коштом повинен бути еталоном.

Незаперечна істина, що підручником користуватись тим легше, чим він стисліший. Крім того, неможливо спростувати відому істину "Короткість – сестра таланту". Але чи можна назвати стислими діючі підручники з інформатики для 9 класу, які всього на 35 уроків містять по кілька сотень сторінок (наприклад, [19] – 344 сторінки)? Ще гостріше питання: чи легко *вперше* викладати інформатику в 9 класі при наявності недосконалої програми, чотирьох громіздких підручників без методичних посібників? Відповідь очевидна, бо питання риторичне.

Поряд із стислістю для ліцензованого підручника не менш важливими є науковість, коректність та логічна послідовність. Якщо уважно переглянути переглянути згадані підручники, можна знайти достатньо прикладів порушення цих принципів. Правда, через розпорошення і перемішаність матеріалу на сторінках підручників не завжди можна чітко відрізнити ці порушення від придирок. Але у подібних ситуаціях більшість сумнівів читача слід зараховувати у негатив авторам. Щоб уникнути голослівності, пропонуємо приклад. Мова ідеться про інформаційну систему. З одного боку вона має дві обов'язкові складові: апаратну та програмну, з іншого боку, в підручнику [19], п.33 на стор.29 стверджується, що сучасний комп'ютер є інформаційною системою. Якщо й погодитись на це, то лише за умови, що до складу комп'ютера, крім апаратури, обов'язково входить програмне забезпечення. Але чи слід дев'ятикласників навантажувати такими небезсумнівними і складними деталями?

На попередніх сторінках здебільшого ми аналізували діючі підручники. Нижче проведено порівняльний аналіз трьох із них: Й.Я. Ривкінда та ін. [18], Н.В. Морзе та ін.[19] та І.Л. Володіної, В.В. Володіна [20]. Всі вони мають у більшій чи меншій мірі загальні вади сучасних підручників. Це і неякісні папір та поліграфія, і громіздкість та переобтяженість зайвою деталізацією, і



необґрунтована данина гуманізації та демократизації навчання, що виражається у зловживанні часто недоцільними ілюстраціями та унаочненнями, і науковий рівень, що не повністю відповідає віковим особливостям учнів 9 класів.

Кожен зайвий фон для тексту, обрамлення або оформлення сторінки чи завиток в нумерації, як проілюстровано вище, розсіює увагу, відволікає від основного, збільшує психологічне навантаження. Сучасні учні не дуже люблять читати та самостійно працювати з книгою, тому всілякі рубрики зразка "Ви дізнаєтесь", "Вивчаємо", "Діємо", "Працюємо в парах", "Поглиблюємо знання", "Обговорюємо", "Дізнайтеся більше", "Перевірте себе" тощо, особливо в умовах цейтноту відведеного програмою часу можуть викликати не сумісні із зацікавленням емоції та мотивацію.

Порівняння підручників проведемо на прикладі вже згаданої теми "Поняття файлу, каталогу. Ім'я файлу та каталогу, розширення імені файлу. Імена зовнішніх запам'ятовуючих пристроїв, шлях до файлу".

підручник	[18]	[19]	[20]
кількість сторінок	11,5	12	10
додаткові поняття ²⁶	12	3	3
відсутні описи ²⁷	3	-	-
додаткові описи	7	4	4
запитання	18	10	13
завдання	9 (з підпунктами 40)	13	19 (в т.ч. 7 додаткових)
практичні роботи	практична (17 п.)	лабораторна (16 п.)	практична (12 п.)

Звідси випливають висновки:

- 1) Кількість сторінок, відведених для цієї теми приблизно однакова, але в усіх підручниках занадто велика для одного уроку.
- 2) В усіх підручниках є додаткові, не передбачені програмою поняття. Але у першому їх значно більше (12), ніж у інших (по 3), тому підручник [18] у цьому відношенні гірший.
- 3) У підручнику [18] відсутній опис трьох понять, передбачених програмою, що також понижує його якість.
- 4) Всі підручники містять додатковими описами (перший – 7, решта – по 4).
- 5) У [20] введені поняття *системних* (службових, створених операційною системою) та *користувацьких* (створених користувачем) **папок**, причому *Робочий стіл* названо головною системною папкою, що містить усі інші папки комп'ютера. Таким чином допущено щонайменше три некоректності. По-перше, Робочий стіл містить *ярлики*, серед яких можуть бути і позначення папок. Ярлики ж – це кнопки запуску додатків та відкриття вікон, у тому числі й папок, файлів чи пристроїв. По-друге, прийнято вважати, що папку можна відкрити, а також вийти з неї. Як відомо, з Робочого столу виходу немає. По-третє, Робочий стіл більш коректно співставляти не з папкою, а з вікном *Windows*, хоча і тут немає повної аналогії.
- 6) Зустрічаються й інші, не зовсім прозорі або не коректні описи. Так у тому ж [20] дано означення поняття формату файлу, перед яким описано назву файлу (стор. 124). Якщо термін "формат" розуміти, як загальний вигляд або

²⁶ Додаткові - це не передбачені програмою поняття та описи (NTFS, CDFS, користувацька папка, папка першого рівня, порожній файл, вигляд папок у Windows Vista, відмінності позначень у Windows XP та Windows Vista, відтворення музичних файлів та відеозаписів і т.д.)

²⁷ Старший каталог, підкаталог, дерево каталогів і т.д.

запис чогось, то слід розрізняти два поняття "формат імені файлу" та "формат файлу", причому, тут важливіше перше.

- 7) Виходячи з практичної форми вивчення курсу інформатики, практичним роботам слід відводити провідну роль, а рекомендовані підручники повинні це забезпечувати. Підручники [18], [20] містять всі 12 практичних робіт, передбачених програмою. Цілком природно, щоб вони повністю відповідали пунктам навчальних досягнень: *розпізнає, класифікує, вміє та використовує*. Але у програмі важко чітко розділити частини цих пунктів, що відносяться до різних практичних робіт. У підручник [19] замість практичних робіт включено лабораторні роботи.

На практичних та лабораторних роботах слід зупинитись окремо. Звернімо увагу на оптимальну кількість та на форму і зміст кожної. Якщо їх вважати заліковими, то передбачена програмою кількість занадто велика (12 робіт на 35 уроків, тобто на кожному третьому-четвертому уроці). Але в програмі не вказано, що всі практичні роботи обов'язкові, тому їх можна поділити на залікові та поточні. У [19] кількість лабораторних робіт вісім, тому усі можна вважати заліковими.

Для заліку форма та зміст лабораторних робіт у [19] безперечно кращі, ніж у решти підручників. Вони оформлені у вигляді таблиці, що містить конкретні завдання, виконання яких легко фіксувати в умовах одночасного контролю групи учнів. Оцінювання можна зробити як під час роботи, так і після її закінчення, адже для кожного питання вказано кількість балів.

Цим перелік зразків відступу від програми, порушення науковості, логічної послідовності та строгості в названих підручниках не обмежується, але очевидно, що у такій ситуації необхідно знайти оптимальний вихід і відповідну методику викладання. Це можливо, наприклад, шляхом стислого викладу матеріалу, подібно експрес-курсу з теми "Алгоритмізація та програмування", представленого засобами електронних дидактичних засобів навчання, зокрема у вигляді електронних сценаріїв уроків (ЕСУ), електронних конспектів уроків (ЕКУ), електронних роздаткових матеріалів (ЕРМ) та електронних тестуючих систем (ЕТС).

Окремої уваги потребує складання календарно-тематичного плану з інформатики для 9 класу. Є необхідність переміщення тем та перерозподілу навчальних годин. Тема кожного уроку потребує стиснення формулювання (додаток 2).

Розуміючи, що змістові зв'язки між багатьма темами курсу є достатньо слабкими, автори у пояснювальній записці зауважують, що вони не мали на меті догматизувати послідовність викладання матеріалу, тобто стандартизація певного порядку їх вивчення є недоцільною. Вчителі інформатики *"можуть змінювати порядок вивчення і обсяг тем курсу залежно від рівня підготовки учнів і технічного оснащення школи, вибудовуючи в такий спосіб найбільш доречну для конкретного навчального закладу або класу траєкторію навчання"*, але при умові, що змінюючи порядок тем курсу, вчитель не порушуватиме *"порядок викладання тем, між якими є суттєві змістові залежності"* [2].

У програмі [2] зазначено, що *”кількість тематичних оцінок встановлюється учителем, програма дозволяє об’єднувати теми, на вивчення яких відведено 2-4 навчальні години. Зокрема доцільно поєднати оцінювання з тем 1 і 2, 3 і 4, 6 і 7. Форму проведення тематичних оцінювань учитель обирає самостійно: контрольні роботи, тестування, комплексні практичні роботи, захист навчальних проектів тощо”*.

Враховуючи різні підходи до вивчення важливих тем курсу у різних підручниках, при складанні календарно-тематичного плану обов’язкова прив’язка до вибраного підручника. Порівняльний аналіз трьох підручників [18], [19] та [20], на наш погляд, надає перевагу підручнику авторського колективу Н.В. Морзе, В.П. Вембер та О.Г. Кузьмінської. Найсуттєвішими аргументами на його користь ми бачимо підходи до вивчення теми *”Комп’ютерна графіка”, ”Службове програмне забезпечення” та ”Основи інтернету”*. Скориставшись порадою у пояснювальній записці [2]: *”якщо у переліку вказано кілька програм певного типу, то це означає, що можна використовувати будь-яку з них, на вибір учителя”*, автори реалізують ідею вивчення векторної графіки на базі вмонтованих графічних можливостей *Microsoft Office*, зокрема у *Microsoft Word* та *Microsoft Power Point*. Беручи до уваги вікові особливості учнів 9 класу, це дійсно простіше і доцільніше, ніж розгляд векторної графіки в *CorelDraw*, яке, як вищий ступінь, можна вивчити в наступні роки. Враховуючи те, що упакування й архівація інформації та антивірусна профілактика у наш час міцно пов’язані з роботою в Інтернеті, природно поєднати теми *”Службове програмне забезпечення” та ”Основи інтернету”*.

При складанні календарно-тематичного плану (додаток 2) тут зближено також теми *”Системне програмне забезпечення” та ”Локальні мережі”*. Незначний, але потрібний перерозподіл навчального часу зроблено також на користь теми *”Апаратне забезпечення інформаційних систем”* за рахунок резерву. Це дозволить детальніше познайомити учнів з роботою пристроїв вводу інформації, адже потреба управляти комп’ютером, в тому числі і через клавіатуру, виникає значно раніше за вивчення текстового редактора.

Рамки матеріалу дуже обмежують методичне дослідження викладання інформатики в 9 класі, тому тут піднято лише найбільш загальні та суттєві питання, подальша робота в цьому напрямку може привести до створення експрес-курсу інформатики для 9 класу та методичного посібника до нього.

додаток 1

Фрагмент програми з інформатики для 9 класу
(тема *” Системне програмне забезпечення ”*, 7 годин)

<p>Загальні відомості про системне, службове та прикладне програмне забезпечення. Класифікація, основні функції та складові операційних систем. Поняття про ядро операційної системи, <i>інтерфейс користувача</i>,</p>	<p>Учень пояснює: відмінність між системним, службовим та прикладним програмним забезпеченням; поняття ядра операційної системи, <i>інтерфейсу користувача</i>, драйвера та утиліти; <i>поняття файлової системи</i>; відмінності між поширеними файловими системами; зміст шляху до файлу; <i>поняття файлу та каталогу</i>;</p>
---	--

<p>драйвери та утиліти. <u>Різновиди інтерфейсу користувача.</u> <i>Поняття файлової системи, відмінності між поширеними файловими системами.</i> <i>Поняття файлу, каталогу. Ім'я файлу та каталогу, розширення імені файлу. Імена зовнішніх запам'ятовуючих пристроїв, шлях до файлу.</i> <u>Робота з основними елементами графічного інтерфейсу користувача операційної системи. Використання вікон, меню, елементів керування.</u> <i>Робота з об'єктами файлової системи: створення, копіювання, перейменування, переміщення та видалення об'єктів.</i> Використання ярликів. Використання буфера обміну. Пошук інформації на комп'ютері. <i>Запуск на виконання програм. Типи файлів. Зв'язок типів файлів з програмами та з розширеннями імен файлів.</i> Використання автономної та онлайнної довідки операційної системи. Встановлення й видалення програм. Відновлення видалених даних. Програма перевірки й очищення дисків. Дефраг-ментація дисків. Контрольні точки відновлення операційної системи. <u>Практична робота №2. Робота з інтерфейсом користувача операційної системи.</u> <u>Практична робота №3. Робота з об'єктами файлової системи.</u> Практична робота №4. Пошук інформації на комп'ютері.</p>	<p><i>поняття типу файлу; поняття та призначення ярликів; необхідність періодичної перевірки та очищення дисків;</i> Учень описує: <i>призначення та основні функції операційної системи; основні правила роботи з об'єктами файлової системи; різновиди інтерфейсу користувача;</i> <i>способи запуску програм на виконання; спосіб відновлення видалених даних; методику встановлення й видалення програмного забезпечення;</i> Учень розпізнає: <i>файли та каталоги; імена, розширення імен та типи файлів; файли, яким зіставлені програми; стандартні імена зовнішніх запам'ятовуючих пристроїв комп'ютера;</i> Учень класифікує: <u>операційні системи за типом інтерфейсу користувача, за кількістю користувачів та програм, що працюють у системі одночасно;</u> Учень вміє: <u>переміщувати, відкривати, розгортати, згортати вікна та змінювати їхній розмір;</u> <i>визначати й записувати шлях до файлу; переходити до файлу за заданим шляхом; виділяти об'єкти та групи об'єктів для виконання операцій над ними; створювати каталоги;</i> <i>створювати ярлики (посилання на файли, каталоги або диски); перейменовувати файли та каталоги; видаляти файли та каталоги;</i> <i>копіювати й переміщувати файли та каталоги з використанням та без використання буфера обміну;</i> <i>запускати на виконання програми; відкривати файли, типи яких зв'язані з програмами;</i> <i>звертатися до служби технічної підтримки виробників операційної системи та прикладного програмного забезпечення;</i> <i>встановлювати й видаляти програми за допомогою спеціальних засобів, що надаються операційною системою;</i> <i>відновлювати видалені файли та папки; створювати резервні копії файлів та папок;</i> <i>створювати контрольні точки відновлення та визначати розклад їх автоматичного створення;</i> <i>повертати стан системних файлів до контрольної точки відновлення;</i> <i>знаходити на комп'ютері необхідну інформацію в автоматизованому режимі;</i> <i>визначати необхідність дефрагментації дисків;</i> Учень використовує: <u>елементи керування для виконання дій в середовищі операційної системи;</u> <u>меню вікна папки та головне меню операційної системи;</u> <i>буфер обміну для копіювання та переміщення файлів, каталогів та ярликів; автономну та онлайнну довідку операційної системи;</i> <i>засоби автоматизованого пошуку інформації на комп'ютері;</i> <i>засоби відновлення стану системних даних;</i> <i>програму перевірки й очищення дисків;</i> <i>програму дефрагментації дисків.</i></p>
--	--

Календарно-тематичне планування курсу інформатики для 9 класу

I семестр

1. Інформація. Інформаційні процеси та системи (2 год.)	
1	Інформаційні процеси. Інформація, її різновиди, подання, представлення та вимірювання.
2	Інформаційні системи та технології, їх розвиток і застосування. Апаратне та програмне забезпечення інформаційної системи. Інформаційна культура та інформатична компетентність.
2. Апаратне забезпечення інформаційних систем (5 год.)	
3	Архітектура персонального комп'ютера. Призначення пристроїв введення, виведення, зберігання та обробки інформації. Робота з ручним маніпулятором mouse.
4	Процесор та запам'ятовуючі пристрої. Периферійні пристрої. Розвиток обчислювальної техніки. Правила техніки безпеки під час роботи на комп'ютері.
5	Клавіатура. Робота з клавіатурним тренажером. Введення тексту.
6	Редагування тексту з допомогою <i>Notepad</i> (блокнота).
7	<i>Практична робота №1</i> . Залік.

Тематичне оцінювання

3. Системне програмне забезпечення (7 год.)	
8	Системне, службове та прикладне програмне забезпечення. Операційні системи. Диски, файли та каталоги.
9	Графічний інтерфейс користувача операційної системи. Вікна, меню, елементи керування. Робота з об'єктами файлової системи.
10	<i>Практична робота №2</i> . Робота з інтерфейсом користувача операційної системи.
11	<i>Практична робота №3</i> . Робота з об'єктами файлової системи.
12	Пошук інформації в комп'ютері. Довідкова система. Виконання програм. Типи файлів та їх зв'язок з програмами.
13	Інсталяція програм та відновлення даних. Сервісні програми.
14	<i>Практична робота №4</i> . Пошук інформації на комп'ютері. Залік.

Тематичне оцінювання

4 Комп'ютерні мережі. Локальні комп'ютерні мережі (3 год.)	
15	Глобальні та локальні комп'ютерні мережі, їх апаратне й програмне забезпечення. Поняття про сервери та клієнти. Мережні протоколи.
16	Доступ до ресурсів локальної мережі, вхід та навігація у ній. Спільне використання файлів, папок та пристроїв.
17	Віддалене керування комп'ютером. <i>Практична робота №7</i> . Спільне використання ресурсів локальної мережі. Залік.

Тематичне та семестрове оцінювання

II семестр

5. Службове програмне забезпечення (3 год.)	
18	Комп'ютерні віруси та їх класифікація. Антивірусні програми та антивірусна політика. <i>Практична робота №5</i> . Захист комп'ютера від вірусів.
19	Стиснення інформації. Програми-архіватори. <i>Практична робота №6</i> . Архівування та розархівування даних.
20	Форматування та копіювання дисків. Запис інформації на оптичні носії. Залік.
6. Основи інтернету. Пошук в інтернеті. (3 год.)	
21	Глобальної мережі та інтернет. Адресація в інтернеті. Функції провайдера та послуги інтернету. Гіпертекстові документи. Навігація в інтернет.
22	Браузери та їх налаштування. Вибір системи кодування. Веб-сторінки, веб-сайти та їх збереження на локальному комп'ютері.
23	Засоби пошуку інформації в Інтернеті. Принципи функціонування веб-каталогів та

	пошукових систем. <i>Практична робота № 8.</i> Пошук інформації в Інтернеті. Залік.
--	---

Тематичне оцінювання

6. Основи роботи з текстовою інформацією. (5 год.)	
24	Системи обробки текстів та середовище текстового редактора. Формати текстових файлів. Робота з текстовими документами.
25	Введення й редагування тексту. Операції з фрагментами тексту. <i>Практична робота №9.</i> Робота з текстовими фрагментами.
26	Послуги текстового редактора. Форматування шрифтів і абзаців. <i>Практична робота №10.</i> Введення, редагування й форматування тексту.
27	Об'єкти в текстовому документі. Написи, заголовки та формули. Залік.

Тематичне оцінювання

7. Графічні зображення. Основи векторної графіки. (3 год.)	
28	Графічні можливості офісних програм. Панель "Малювання". Векторні зображення, їх побудова та обробка.
29	Інструменти малювання. Об'єкти та операції з ними. Автофігури. Параметри об'єктів.
30	Взаємозв'язок між текстовими та графічними компонентами у текстових документах. <i>Практична робота №12.</i> Створення векторних зображень. Залік.

7. Комп'ютерна графіка та її види. Растрова графіка (4 год.)

31	Комп'ютерна графіка та засоби обробки графічних даних. Растрові й векторні зображення та їх властивості. Графічні формати. Робота з графічними файлами.
32	Панель інструментів Paint. Редагування малюнків та створення надписів.
33	Використання головного меню та палітри кольорів. Роздільна здатність, глибина кольору та якість растрових зображень.
34	<i>Практична робота №11.</i> Створення растрових зображень. Залік.

Тематичне та семестрове оцінювання

Резерв навчального часу. (1 год.)	
35	Підсумки за навчальний рік.

VIII. Організація позакласної роботи з основ інформатики.

Позакласна робота з будь-якого навчального предмету має *мету, завдання, особливості, напрямки* (форми) та *ступені* (рівні). Мета – це досягнення ключового, стратегічного завдання: виявлення обдарованих учнів і їх максимальний розвиток у даній галузі. Решта *завдань* мають тактичний характер, вони впливають із ключового й тісно пов'язані між собою. Найважливіші серед них такі: популяризація навчального предмету і зацікавлення ним якомога більшої кількості учнів; виявлення рівня здібностей та підготовленості кожного учня; організація навчальної підготовки; створення атмосфери змагання.

Серед особливостей перш за все слід виділити поєднання гурткового характеру занять та індивідуальної форми навчальної роботи. *Особливостями* позакласної роботи з інформатики є тісні міжпредметні зв'язки, базування на математичній основі та прикладна спрямованість. Її результати – це абстрактні програмні проекти, реалізовані на комп'ютері. Але основну частину курсу інформатики складають інформаційно-комунікаційні технології, тому позакласна робота з інформатики логічно мало пов'язана з основним курсом.

Виходячи з мети, домовимось не відносити до позакласної роботи з інформатики масові заходи розважального характеру зразка вікторин, КВК тощо. Основними формами позакласної роботи вважаються підготовка та участь у предметних олімпіадах і робота в секціях МАН. Але, якщо



поміркувати, то це не напрямки, а *ступені* (див. схему VIII.1).

На місцях часто дотримуються абсурдного правила: відібрати зацікавлених учнів; здібніших спрямувати на розв'язування олімпіадних завдань; решті запропонувати написання рефератів для секцій МАН. Циклічність участі в роботі секції МАН переважно не більше одного навчального року. При таких умовах учень може бути лише озвучувачем (спікером) тез до реферату, адже серйозна дослідницька робота та виконання програмного проекту вимагає тривалого часу і кваліфікованих навичок у вузькій галузі.

Така участь у роботі МАН з одного боку є профанацією самої ідеї, з іншого – фальсифікацією результатів, нарешті своєрідним проявом корупції, адже пов'язана з присудженням призових місць та премій, встановленням рейтингів і пільгових прав для вступу на навчання. Вона не потрібна ні учневі, ні учителю, адже забирає час, приносячи замість практичної користі лише моральну шкоду. Як і будь-які фальсифікації, стверджувати доказово це дуже важко як з технічної, так і з правової сторін, але, враховуючи специфіку – навчально-виховну роботу з підростаючим поколінням, необхідно морально засуджувати. Як доказ, обмежимось констатацією кількох, але красномовних фактів. У конкурсах-захистах МАН беруть участь дослідницькі проекти, які доступні лише поодиноким, особливо обдарованим і підготовленим учням. Отже щорічна кількість учасників конкурсів-захистів може бути хоча б на один

порядок меншою за кількість учасників олімпіад з інформатики, а насправді спостерігається майже однакова кількість. Багато разів доводилось слухати захисти рефератів учнями, які явно не розуміли використаних ними термінів. Безперечно, що за кожною роботою МАН повинен стояти науковий керівник, яким може бути далеко не кожен шкільний учитель. У сільській місцевості роль наукових і педагогічних керівників майже виключно виконують учителі-предметники, причому, часто один керує кількома конкурсантами відразу.

Висновки очевидні, але не будемо поспішати їх робити. Справа у порушенні рівності умов таких конкурсів для елітних і пересічних навчальних закладів, для міської та сільської місцевості. Вищеописаними фактами та висновками не переслідувалась критична мета, вони потрібні у контексті роздумів про організацію позакласної роботи з інформатики лише як несприятливе явище.

Робота в секціях МАН може йти паралельно з поглибленим вивченням предмету, підготовкою до участі в олімпіадних змаганнях, але досягти успіху може лише при організації справжньої індивідуальної дослідницької роботи, що базується на глибокому знанні предмету і високому рівні інтелекту, які досягаються при розв'язуванні великої кількості задач з "родзинкою", тобто олімпіадних.

На наше переконання робота в секціях МАН повинна бути лише там, де є висококваліфіковані кадри наукових та педагогічних керівників. Захист рефератів слід проводити, починаючи з обласного етапу, бо в середньостатистичному районі важко знайти конкурентні між собою проекти.

Таким чином, правильно організована робота секцій Малої академії наук є вищим ступенем або рівнем позакласної роботи з інформатики, а сфера розв'язування олімпіадних задач з програмування у такому випадку стає першим, базовим рівнем і разом з цим найбільш серйозною та масовою формою позакласної роботи.

Правда тут також є важливе застереження. Воно полягає в тому, що програмування у загальноосвітніх школах вивчається дуже обмежено і лише на завершальному етапі навчання. Олімпіади ж проходять у листопаді-березні, коли навіть випускники ще не ознайомлені з курсом програмування. Таким чином олімпіади з інформатики (читай – з програмування), щоб не порушувати рівності умов, про що говорилось вище, доцільно проводити окремо для учнів спеціалізованих шкіл з поглибленим вивченням інформатики та спеціальних гуртків і факультативів, і окремо для учнів загальноосвітніх шкіл.

Але, всупереч цьому, вже на протязі двох десятиліть олімпіади з інформатики проводяться масово для всіх шкіл. Як показує практика, успішно виступають лише ті учні, які на час участі в олімпіаді вже 3-4 роки вивчали програмування поглиблено під керівництвом досвідчених викладачів.

Нижче розглянемо базову форму позакласної роботи з інформатики, тобто організацію підготовки та проведення олімпіад.

VIII.1 Про сучасні олімпіади з інформатики.

Питання про суть і завдання сучасних учнівських олімпіад з інформатики, коли, як та з чого починати підготовку до них, досить очевидне, але зовсім не риторичне. Відповідь має кілька складових. Пропедевтичну роботу з програмування бажано починати приблизно з сьомого класу, бо, з одного боку, в учнів необхідно встигнути сформулювати дуже струнку, чітку і об'ємну наукову систему знань, з іншого – необхідна математична культура мислення, яка з'являється не раніше цього віку.

Необхідно уточнити термін "учнівські олімпіади з інформатики", переглянувши форми і методи їх підготовки та проведення, тобто змінити акценти і узгодження із шкільним предметом "основи інформатики". Спробуємо сформулювати і узагальнити вимоги до олімпіад в традиційному розумінні (далі – просто *олімпіади з інформатики*) та згідно сучасних вимог накреслити шляхи олімпіадної роботи на нижчих (I-III) етапах. Розуміючи, що тема надто широка, обмежимося її фрагментарним оглядом і орієнтовними методичними порадами, не вдаючись до рекомендацій, які повинні спиратись на нормативну базу та педагогічні дослідження.

Як відомо, завдання не тільки для III-IV, а й для нижчих етапів, вимагають знання ряду тем, що виходять за межі шкільних програм з інформатики та математики. Як тут не пригадати слова з пісні: "...кандидат наук и тот над задачей плачет". Учням пропонуються завдання, виконання яких потребує не простої обізнаності, а свідомого розуміння і вільного використання відомостей з теорії множин, аналітичної геометрії, комбінаторики, теорії графів та інших математичних дисциплін, що вивчаються не в середніх навчальних закладах, а у вузах, причому не завжди на перших курсах і переважно на математичних та технічних спеціальностях. Стали необхідними також вміння користуватись рекурсивним методом, методами довгої арифметики, перебору та динамічного програмування, вміння оцінювати програми на ефективність та швидкодію. Наведемо одну характеру цитату 10-15-річної давності [22]: "...метод динамічного програмування для скінченного простору рішень справді досить простий, а його ідея зрозуміла навіть школярам з пересічною підготовкою. Але й годі вимагати від них самостійно додуматись за кілька годин до того, що було останнім словом науки лише 35 років назад". Як видно, вона стосується доступності для учнів методу динамічного програмування. Перш за все вона дуже суперечлива. Чи справді він досить простий, та ще й для школяра з пересічною підготовкою? А як же з "...годі вимагати від них самостійно додуматись за кілька годин"? І, врешті, навіщо учням, навіть під час олімпіад давати завдання, для розв'язування яких необхідно знати, розуміти і вміти застосовувати те, що було останнім словом науки лише кілька десятків років тому? Очевидно, що олімпіади з інформатики з самого початку поміщені в логічний вакуум. Наше завдання – звузити його до тонкої і прозорої плівки, через яку пересічний учень може побачити горизонти програмування, а при бажанні чи потребі досягти їх. Тому позакласну роботу з інформатики слід вести якісно, як додатковий навчальний предмет, причому, не тільки для забезпечення успішної участі учнів у олімпіадах, а й для розвитку в них

алгоритмічної культуру, логічного мислення, комп'ютерної грамотності, підготовки до навчання у вузах. Це проблема відповідності задач математичній підготовленості учасників олімпіади.

Окрема проблема – це *відсутність вікової диференціації*. Адже у програмуванні важко класифікувати задачі, як по віку, так і по складності необхідного теоретичного матеріалу. Через це на олімпіадах з інформатики здебільшого даються однакові задачі для 8-11 класів. Спробуємо знайти один із варіантів розв'язання цієї проблеми, навівши приклади завдань, окремих для 10 та 11 класів.

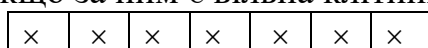
9-10 класи

- 1) *Системи числення (Notations10)*. Дано натуральне число X у четвірковій системі числення. Описати алгоритм його переводу у вісімкову систему числення.
- 2) *Просте число (SimplNumb10)*. Описати алгоритм перевірки, чи буде простим дане натуральне число.
- 3) *Сходишки (Ladder10)*. По східцях (21 сходинка) вгору піднімається десятикласник. Описати алгоритм підрахунку кількості способів дістатись до верхньої сходинки, якщо за один крок він може піднятися не більше, ніж на 3 сходинки.
- 4) *Обмін місцями (ExchangPlac10)*. Потрібно описати стратегію обміну місцями k значків "x" та "o" на полі, що має вигляд смужки з $2k+1$ клітинок (див. мал.), якщо "хрестики" можуть рухатись тільки в напрямку вправо, а "нулики" можуть рухатись тільки в напрямку вліво, стаючи на сусідню клітинку, якщо вона вільна, або "перестрибнувши" через різнойменний значок, якщо за ним є вільна клітинка:



11 клас

- 1) *Системи числення (Notations11)* Дано натуральне число X у n -ковій системі числення. Описати алгоритм його переводу у m -кову систему числення, якщо $m \neq n$ і $2 < n < 16$, $2 < m < 16$.
- 2) *Просте число (SimplNumb11)* Описати алгоритм підрахунку кількості простих чисел, які знаходяться між двома даними натуральними числами M та N .
- 3) *Сходишки (Ladder11)* По східцях ($k \leq 30$ - кількість сходинок) вгору піднімається одинадцятикласник. Описати алгоритм підрахунку кількості способів підняття на верхню сходинку, якщо за один крок він може піднятися не більше, ніж на n сходинок.
- 4) *Обмін місцями (ExchangPlac11)* Потрібно описати стратегію обміну місцями значків "x" та "o" на квадратному полі з $(2k+1) \times (2k+1)$ клітинок (див. мал., $k=3$), якщо кожен "хрестик" може рухатись тільки вправо або тільки вниз, а кожен "нулик" може рухатись тільки вліво або тільки вгору, стаючи на сусідню клітинку, якщо вона вільна, або "перестрибнувши" через різнойменний значок, якщо за ним є вільна клітинка:



×	×	×	×	×	×	×
×	×	×	×	×	×	×
×	×	×		○	○	○
○	○	○	○	○	○	○
○	○	○	○	○	○	○
○	○	○	○	○	○	○

Прокоментуємо наведені завдання. Перш за все в них замість вимоги написати програми згідно з даними технічними умовами, пропонується описати алгоритми, що важливо для теоретичного туру. Кожне завдання можна звести до прозорого, не дуже громіздкого алгоритму, опис якого наближений до звичайного програмного матеріалу з предмету, зводиться до формулювання ідеї, не потребує глибоких теоретичних знань. Нарешті, умови вибрано так, що завдання для 11-го класу є логічним ускладненням завдання для 10-го класу з таким же номером. На наш погляд, обидві сформульовані вище проблеми тут якщо не вирішені цілком, то, принаймні, зведені до мінімуму.

Тут не даремно описані зразки завдань, придатних для теоретичного туру олімпіади. Ми вважаємо, що на I-II етапах, коли регламент залежить від оргкомітету шкільного та районного масштабу, олімпіаду корисно проводити в два тури (I тур – теоретичний, II тур – практичний). При цьому на районному етапі олімпіади слід забезпечити проведення теоретичного туру за кілька днів до практичного, з використанням завдань, складених районним журі. Доцільність такої форми пов'язана з третьою проблемою, яка полягає у невизначеності *співвідношення між теоретичною та практичною частинами роботи над завданнями*. З одного боку не можна ігнорувати роботу над описом математичної моделі, складанням, доведенням та оцінкою алгоритму, з іншого боку, тільки ступінь практичної реалізації у вигляді діючої програми дозволить зробити правильну оцінку.

Офіційні суперечки про те, в якій формі - теоретичній, практичній чи комбінованій слід проводити олімпіади з інформатики на III-IV етапах, давно вважаються завершеними на користь практичної. Тому перевірка письмових робіт і аналіз програм не обов'язкові. Єдиним визнаним продуктом учасника олімпіади є програма. Така позиція загальноприйнята вже . Переконаливо сказано про це в [21]: "*Акцент усе більше зміщується від «краси алгоритму» до «діючої програми»... Журі олімпіад уже не розглядає тексти програм учасників*. Оцінювання діючої програми здійснюється як оцінювання діючого «чорного ящика»: на вхід подаються вхідні дані, а на виході має бути правильний результат. *Хитромудрі обмеження, що містяться в умові, задають необхідний рівень ефективності розв'язання*".

Якщо в умовах задач вимагається описати алгоритм, то, очевидно, постає ще одна проблема, не логічного, як три попередні, а технічного характеру: *як описувати алгоритм*. Цій проблемі присвятимо нижче окремий пункт. Через відсутність однакових умов проведення на місцях, *участь в олімпіадах з інформатики не може бути обов'язковою*, що на практиці де-факто має місце. Отже дані про участь і результати олімпіад з інформатики не

можуть бути обов'язковими компонентами в статистичній звітності та при підведенні підсумків роботи навчальних закладів.

Проведення районних олімпіад з інформатики за зразком обласного етапу не ефективне. Перш за все для організації районного туру, особливо у сільських районах, важко знайти єдиний населений пункт з достатньою кількістю комп'ютерів. Значна частина учнів, особливо 9-10 класів у листопаді-грудні, як уже згадувалось, ще не має достатніх навичок програмування, тому для значної (інколи переважної) кількості учасників практичний тур з використанням комп'ютерів не надто актуальний. Всі етапи Всеукраїнської олімпіади з інформатики мають спільну головну мету: відбір учасників наступного етапу. Але, очевидно, що будь-яка олімпіада покликана також популяризувати предмет, сприяти піднесенню загального рівня знань, умінь та навичок (див. схему VIII.1). У зв'язку з цим можна погодитись, що різні етапи олімпіади з інформатики повинні мати власні локальні завдання. Наприклад, перший (шкільний) етап у першу чергу покликаний виявити тих, хто цікавиться предметом більше за інших та володіє елементами програмування. Другий (районний) етап може відібрати серед призерів I-го етапу тих, хто має здібності до точних наук, схильність абстрактно, логічно, структурно та алгоритмічно мислити, має певні навички програмування. Третій (обласний) етап дає змогу відібрати учнів, які мають значний досвід програмування та перспективи удосконалення. IV-й етап виявляє обдарованих у галузі програмування. Умовно можна об'єднати у дві групи I-II та III-IV етапи. Очевидно, враховуючи це, послідовно від етапу до етапу слід змінювати методику підготовки завдань та умови проведення змагань. Так на IV-у етапі склалась традиція проводити два практичні тури. На наш погляд, як уже було сказано вище, на I-II етапах резонно також проводити два тури, але перший – теоретичний, а другий – практичний.

Олімпіади – це марафон для особливо сильних. Тому після теоретичного туру до наступних змагань слід допускати лише порівняно невелику кількість здатних до такої специфічної сфери діяльності, як програмування. Проведенню районних етапів олімпіади з інформатики є кілька перепон. Серед них першою можна вважати включення районних олімпіад з інформатики, до етапів Всеукраїнської олімпіади, а це не дозволяє змінювати умови проведення. Проте ця перепона легко усувається - можна вважати теоретичний тур попереднім і неофіційним. Залишається лише дозволити в другому турі не брати участь навчальним закладам, які не можуть виставити достатньо підготовлених учасників.

Такі зміни дозволять проводити практичний тур районної олімпіади за всіма правилами, як зауважено в наведеній вище цитаті [21], можна буде змістити акцент від “краси алгоритму ” до “ діючої програми ”, а журі одержить можливість виключно оцінювати дієвість програми, застосувавши в тестах обмеження, що містяться в умові і задають необхідний рівень ефективності розв'язання.

Існує ще одна, можливо, не дуже суттєва перепона впровадженню теоретичного туру. Справа в тому, що графік олімпіад та конкурсів-захистів у секціях МАН досить тісний, тому важко організувати два тури олімпіади з інформатики.

Результати районних олімпіад слід розглядати на семінарі вчителів інформатики, підготувавши методичні рекомендації. Вони можуть бути посібником для вчителів і учнів для наступної позакласної роботи з предмету.

VIII.2 Якою повинна бути олімпіадна задача?

Якщо є діюча при всіх умовах і обмеженнях задачі програма, то ніякі правила опису алгоритмів та оформлення письмової частини роботи не потрібні. Але коли немає діючої програми, доведеться оцінювати написане учнем, інакше перевірка буде формальною.

Після аналізу письмової частини роботи учасника олімпіади з інформатики можна об'єктивно оцінити роботу, як за наявності програми, діючої коректно чи ні, так і без програми взагалі.

Що повинно обов'язково входити до письмової частини олімпіадної роботи? Назвемо та опишемо орієнтовні етапи:

Дослідження умови задачі та оголошення й опис величин;

Опис математичної моделі;

Загальний опис і доведення алгоритму;

Уточнення алгоритму та оцінка його складності;

Щоб розібратись, як доцільно підготувати письмову частину олімпіадної роботи з інформатики, можна рекомендувати ретельно опрацювати дуже цінні в цьому відношенні посібники [23] та [24].

Тут для ілюстрації наведемо інший приклад.

Задача Num. (в більш ускладненому варіанті пропонувалась у 2000-2001 н.р. на обласній олімпіаді в Київській області) Дано натуральне число N , записане n одиницями. Визначити число k - найменшу кількість чисел, в записі яких є лише цифри "3" і які в сумі дорівнюють числу N .

Технічні умови:

вхідний файл *Num.dat* містить натуральне число n ($1 < n < 255$).

вихідний файл *Num.res* повинен містити число k .

a) Дослідження умови задачі та оголошення й опис величин.

1) Згідно умови задачі аргументом повинне бути число n цілого типу ($1 < n < 255$), результатом - число k (найменш можлива кількість доданків, також цілого типу).

2) Згідно ознаки подільності натурального числа на 3 робимо висновок: дане число N представляється у вигляді суми, кожен доданок якої містить лише цифри "3" тоді і тільки тоді, коли кількість цифр числа N ділиться без остачі на число 3. Отже, якщо n ділиться без остачі на 3, то представлення його згідно умови задачі можливе, інакше – ні.

3) Повний перебір всіх варіантів запису суми чисел, що мають у записі лише цифри "3" через обмеження $1 < n < 255$ неефективний.

b) *Опис математичної моделі.*

Для відшукування алгоритму спочатку покладемо $n = 5$ (тобто $N = 11111$) і помітимо, що воно представляється згідно умови задачі у вигляді:

$$N = \underbrace{(3333 + 3333 + 3333)}_{9999} + \underbrace{(333 + 333 + 333)}_{999} + \underbrace{(33 + 33 + 33)}_{99} + \underbrace{(3 + 3 + 3)}_9 + \underbrace{3 + 3}_{\text{остача}} \quad (1)$$

основна частина суми

Як видно з (1), число N має 14 доданків, тобто $k = 14$. Основною частиною суми є число $12 = 34$ (кількість груп доданків, що мають у записі цифру “3”).

Наведена формула може бути узагальнена, і матиме вигляд:

$$N = \underbrace{(33\dots3 + 33\dots3 + 33\dots3)}_{\text{число } n \text{ цифр "9"}} + \dots + \underbrace{(33333 + 33333 + 33333)}_{99999} + \underbrace{(3333 + 3333 + 3333)}_{9999} + \underbrace{(333 + 333 + 333)}_{999} + \underbrace{(33 + 33 + 33)}_{99} + \underbrace{(3 + 3 + 3)}_9 + \text{остача}. \quad (2)$$

Остача – це число, менше за 9, воно може ділитись на 3 без остачі, або мати остачу 1 чи 2. В першому випадку це число додасть до числа k одиницю або двійку, у другому приведе до відповіді: $k = 0$. Очевидно, що остання формула дозволяє отримати найменш можливе значення числа k , тобто цілком відповідає умові задачі і може бути її математичною моделлю.

с) *Загальний опис і доведення алгоритму.*

Алгоритм розв’язування задачі у загальному вигляді буде такий:

```

алг NUM
  арг ціл n
  рез ціл k
  поч ціл основна_частина_суми, остача
  {
    основна_частина_суми
    остача
  } згідно формул (1) та
  ввод n
  якщо mod(n, 3) <> 0
  то k:=0
  інакше основна_частина_суми := 3(n - 1)
  визначити(остача) | допоміжний алгоритм
  k := основна_частина_суми + div(остача, 3)
  все
  вивод k
кін
  
```

Прокоментуємо форму запису алгоритму. Вона може бути довільною, в тому числі словесною чи у вигляді блок-схеми. Головне, щоб вибраний спосіб опису алгоритму був використаний доцільно й грамотно. Нам здається, що краще дотримуватись правил запису на алгоритмічній мові. При цьому слід привчати учнів дотримуватись позначень, введених у *a)* та *b)* що в коментарях (на наш погляд, обов’язкових) допоможе уникнути багатослів’я. Коментарі слід використовувати, як для опису величин, так і в тілі алгоритму.

При оголошенні величин бажано використовувати словесні імена, вживаючи знак “_” для з’єднання кількох слів в одне ім’я, наприклад: *основна_частина_суми*.

Під основним алгоритмом слід повністю за такими ж правилами навести допоміжні алгоритми, записані у вигляді *процедур* чи *функцій*.

Доведення існування та єдиності алгоритму – це теорема, тому воно записується згідно традиційних правил доведення теорем. Але спочатку слід чітко сформулювати умову й висновок, які важливо не сплутати з аргументами й результатами задачі. Для нашої задачі теорему існування її розв’язання можна сформулювати так:

Теорема NUM. Якщо дано натуральне число N , записане n одиницями ($1 < n < 255$), то існує алгоритм, який визначає число k - найменшу кількість чисел, в записі яких є лише цифри "3" і які в сумі дорівнюють числу N . (умова й висновок виділені курсивом)

Доведення випливає з 2), (1) та (2). Дійсно, ця формула представляє число N , як суму найменшої кількості доданків, записаних дев’ятками, а кожен доданок, у свою чергу, виділяє найменш можливу кількість чисел, записаних трійками. *Остача* - це число, менше 9 і з нього легко визначити кількість доданків, рівних 3 із допомогою функції $div(\text{остача}, 3)$. Нарешті, процес віднімання від числа N доданків, записаних лише з допомогою цифр "3" скінченний. Ці міркування повністю доводять теорему NUM.

При підготовці учнів до олімпіад з інформатики важливо уточнити також і поняття "олімпіадна задача". Чим же особливі олімпіадні задачі з інформатики (програмування)? Перш за все вони значно "важчі" від тих, які передбачено розглядати згідно програми вивчення курсу інформатики. Усім відомо висловлення академіка А.П. Єршова "Інформатика йде від математики". Тому задачі слід класифікувати перш за все за типами математичних моделей, тобто відповідністю певним розділам математики. Під основними типами олімпіадних задач за математичними моделями тут розуміються традиційні задачі на повний і удосконалений перебір, рекурсії, методи оптимізації та динамічного програмування, координатний метод, графи, відшукування виграшних стратегій і т.д. (див. [22]). Ефективне вивчення евристики в інформатиці можна провести з допомогою посібника [24].

По рівню складності задачі можна поділити на три типи: прості ("втішальні"), технічні та складні ("непідйомні"). Очевидно, що рівень складності – поняття відносне для різних рівнів олімпіади. Особливо слід виділити так звані задачі-пастки. Вони водночас можуть бути й простими й підвищеної складності. Справа в тому, що при знайденні вдалої математичної моделі така задача стає занадто легкою з точки зору програмування. Як приклад можна навести задачу ([25], IV етап XIII Всеукраїнської олімпіади з інформатики):

Дано послідовність, що складається з $2N$ натуральних чисел, які можна розбити на пари таким чином, що сума чисел у всіх парах буде однаковою. Наприклад, числа послідовності 99, 23, 77, 1 можна поділити на пари $1+99=77+23$. Написати програму, яка для даної послідовності встановлює, чи можна її розбити на пари таким чином, щоб добуток чисел у всіх парах був також однаковим.

Після застосування в математичній моделі відомої всім учням теореми Вієта програма розв’язання цієї задачі записується буквально в кілька рядків.

На олімпіаді одночасно слід пропонувати задачі різного рівня складності, але ні в якому разі не можна давати задач, доведених за рахунок обмежень в умовах “до абсурду”, що в попередні роки часто траплялось, особливо на IV етапі.

Можна поради́ти таку послідовність підготовки:

- Повторення й узагальнення традиційного матеріалу про алгоритми та програми, способи їх запису, основні структури алгоритмів, етапи розв’язування задач на комп’ютері;
- Вивчення основ алгоритмічної мови та мови програмування з розглядом і узагальненням понять числа, величини, процедури, функції, перетворення типів в інформатиці;
- Розв’язування так званих “технічних” задач, яке вимагає мінімум зусиль для побудови математичної моделі задачі, але дозволяє відпрацювати “до автоматизму” навички оперування циклами, розгалуженнями, масивами, процедурами, функціями, кодуванням, файлами вводу/виводу, ресурсами комп’ютера (пам’яттю й швидкодією), директивами компілятора, динамічною пам’яттю і т.д.;
- Розв’язування основних типів олімпіадних задач (за математичними моделями);
- Розв’язування задач на евристику.

У якості “технічних” можна пропонувати задачі на пошук, сортування “довгу арифметику”.

При підготовці до олімпіади замість терміну “кінцевий етап” краще вживати “зрілий етап”. На ньому пропонуються комбіновані в розумінні типів задачі, а учні *обов’язково* повинні вміти працювати у автономному режимі.

IX. Замість післямови.

Вище в оглядовому плані описано зміст роботи обласної школи передового педагогічного досвіду при Київському обласному інституті післядипломної освіти педагогічних кадрів у 2007-2010 роках. Щоб відтворити роботу ОШППД повністю, необхідно навести всі річні плани роботи та плани проведення кожного із 12 занять школи, зміст понад 10 прочитаних лекцій, приблизно 25 проведених практичних та семінарських занять, які не втратять актуальності протягом наступних 3-5 років, тому разом можуть утворити авторські курси підвищення кваліфікації для учителів основ інформатики. До названих матеріалів слід додати велику кількість творчих домашніх завдань найактивніших слухачів, які подавались у електронному вигляді та частково у формі брошур формату А5 обсягом 10-20 сторінок кожна. Але специфіка методичної роботи учителя інформатики така, що більшість дидактичних матеріалів мають демонстраційний електронний формат (наприклад, навчальні презентації), який втрачає основний зміст у вигляді, придатному для друку, тому лише кращі зразки цих матеріалів буде запропоновано для публікації у щорічному методичному збірнику КОШОПК.

Робота ОШППД мала конкретний і актуальний характер, що впливав із повсякденних потреб учителів інформатики. Для вивчення піднімалися проблеми, з одного боку глобального характеру, з іншого цілком доступні для вирішення на місцях силами самих учителів.

Проте, складне поєднання ініційованих проблем має динамічний характер, тому їх розв'язання у повному обсязі можливе лише в ідеалі. Виходячи з цього, керівник ОШППД, допомагаючи слухачам у практичному вирішенні названих вище шести актуальних проблем, намагався також озброїти їх дієвими методичними інструментами, з допомогою яких слухачі у майбутній практиці зможуть самостійно виявляти актуальні питання та шукати способи їх розв'язування. Іншими словами, протягом трьох років кожен активний слухач не тільки і не стільки отримував фахову та методичну допомогу, він також, що головне, ставав у доступній мірі методистом-практиком.

Предмет "основи інформатики" має специфіку, яка полягає у дуже обмеженій кількості навчального часу (всього 70 уроків для універсального профілю) та методичній "віддаленості" від традиційного кола навчальних предметів, що створює не тільки серед учителів, адміністративних та методичних працівників, а й багатьох учнів та їх батьків ілюзорне ототожнення шкільної інформатики із побутовими навичками роботи з комп'ютером, де сьогодні більшість вважають себе спеціалістами. Тому часто вважається, що наявність у школі комп'ютерного класу може автоматично вирішити всі проблеми інноваційних та інформаційно-комунікаційних технологій навчання, використання інтернету, створення шкільних сайтів, забезпечення дистанційного навчання та інтерактивного зв'язку з батьками тощо. Всі ці проблеми неправомірно "зав'язуються" на учителів, який часто викладає інформатику по сумісництву. В роботі ОШППД піднято лише проблеми, що стосуються учителів інформатики, для вирішення щойно названих проблем, які слід адресувати перш за все адміністраціям шкіл та планово-фінансовим органам, необхідні значні фінансування у вигляді оплат послуг спеціальних установ для обслуговування комп'ютерної техніки, послуг інтернету, закупівлі прикладного програмного забезпечення та сервісних програм, при наявності двох комп'ютерних класів у навчальному закладі обов'язкова оплата обслуговуючого персоналу, в тому числі лаборанта та інженера. Замість цього практикується тенденція скорочення ставок лаборантів в кабінетах ІКТ, відміна доплат за шкідливість викладачам інформатики. В таких умовах учитель інформатики, який повинен бути головним спеціалістом навчального закладу з ІКТ, не в змозі ні підтримувати свою фахову кваліфікацію, ні вирішувати актуальні проблеми шкільної інформатики.

Література:

1. Інформатика, навчальні програми для 10-11 класів загальноосвітніх навчальних закладів, "Прем'єр", Запоріжжя, 2003 р.
2. Інформатика, навчальна програма для учнів 9 класу загальноосвітніх навчальних закладів, Інформаційний збірник МОН України, № 19-21, липень 2009 р.
3. В.С. Остапець, Погляд на майбутнє шкільної інформатики, "Комп'ютер у школі та сім'ї", № 7, 2006 р.
4. В.С. Остапець, Комплексні електронні дидактичні засоби в системі інформаційних технологій навчання, методичний посібник для вчителів, "Ризографіка", Бориспіль, 2007 р.
5. В.С. Остапець, Комп'ютерне моделювання й основи алгоритмізації та програмування у формі експрес-курсу, "Інформатика", № 10, 2009 р.
6. В.С. Остапець, Експрес-курс з алгоритмізації та програмування, "Інформатика", №11-13, 2009 р.
7. В.С. Остапець, Письмовий опис алгоритмів як невід'ємна складова розв'язку задач з програмування в школі, "Комп'ютер у школі та сім'ї", № 7, 2003 р.
8. В.С. Остапець, Шкільні олімпіади з інформатики, "Світло", № 1, 2001 р.
9. В.С. Остапець, Використання багаторозрядної арифметики, "Інформатика", № 17, 2005 р.
10. В.С. Остапець, Від задачі до задачі, "Інформатика", № 33-34, 2005 р.
11. В.С. Остапець, Метод динамічного програмування у школі, "Інформатика", № 9, 2008 р.
12. В.С. Остапець, Позакласна робота з програмування у школі, навчально-методичний посібник, "Ризографіка", Бориспіль, 2007 р.
13. Дьяченко В.К. Сотрудничество в обучении: О коллективном способе учебной работы. – М.: Просвещение, 1991 г.
14. Пометун О., Пироженко Л. Интерактивні технології навчання: теорія і практика. – К., 2002 р.
15. І.Т.Зарецька, А.М.Гуржій, О.Ю.Соколов, Інформатика, частина 2, підручник для 10-11 класів загальноосвітніх навчальних закладів, Київ, "Форум", 2004 р.
16. Інформатика, № 10, квітень 2001 р. (Т. Непомняща, Бінарний урок)
17. Інформатика та інформаційні технології в навчальних закладах, № 2-3, 2008 р. Програми для загальноосвітніх класів. Інформатика, 7-9 клас, Жалдак М.І., Морзе Н.В., Мостіпан О.І., Науменко Г.Г.
18. Й.Я. Ривкінд, Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько, Інформатика 9 клас, підручник для загальноосвітніх навчальних закладів, Київ, Генеза, 2009 р.
19. Н.В. Морзе, В.П. Вембер, О.Г. Кузьмінська, Інформатика, підручник для 9 класу, Київ, Школяр, 2009 р.
20. І.Л. Володіна, В.В. Володін, Інформатика 9, підручник для 9 класу загальноосвітніх навчальних закладів, Харків, Гімназія, 2009 р.
21. О.Л. Хижа, Розв'язування задач підвищеної складності з інформатики, Київ, Інформатика, №№ 37-43, 1999 р.
22. О.Б. Рудик, Програма спеціального курсу "Прикладна математика", КМІУВ ім. Б. Грінченка, 2000 р.
23. В.С. Савченко, Разработка алгоритмов: от простого к сложному, Донецк, 1996 р.
24. М.З. Грузман, Эвристика в информатике, Винница, Арбат, 1998 р.
25. В. Бондаренко, С. Жук, Задачі XII Всеукраїнської олімпіади з інформатики та обчислювальної техніки, Комп'ютер у школі та сім'ї, № 3, 1999 р.