

Національна академія педагогічних наук України
Інститут педагогіки

В.В. Лапінський, Л.П. Семко, І.М. Семененко

ІНФОРМАТИКА 9

методичний посібник

Київ
«КОНВІ ПРІНТ»
2018

УДК 373.5.016:004] (072)*9кл.

*Рекомендовано до друку вченою радою
Інституту педагогіки НАПН України
(протокол № 11 від 27.11.2017 р.)*

Рецензенти:

Л. В. Непорожня, провідний науковий співробітник відділу біологічної, хімічної та фізичної освіти, кандидат педагогічних наук, старший науковий співробітник;

В. М. Антоненко, викладач кафедри економічної кібернетики Університету державної фіскальної служби України, кандидат технічних наук, доцент;

О. В. Коновалова, вчитель інформатики I категорії гімназії «Міленіум» №318 м. Києва.

Інформатика 9: методичний посібник В. В. Лапінський, Л. П. Семко, І. М. Семененко; – К., ТОВ «КОНВІ ПРІНТ», 2018. – 64 с.

ISBN 978-617-7724-21-5

У посібнику розглядаються методичні підходи до навчання інформатики у 9-ому класі у закладах освіти з поглибленим вивченням інформатики. Надаються методичні рекомендації щодо організації освітнього процесу та викладання основних тем курсу. Запропоновані методичні рекомендації допоможуть учителям інформатики ефективно реалізувати освітній процес.

УДК 373.5.016:004] (072)*9кл.

© Інститут педагогіки НАПН України, 2018

© В.В.Лапінський, Л.П.Семко, І.М.Семененко, 2018

© КОНВІ ПРІНТ, 2018

ISBN 978-617-7724-21-5

ЗМІСТ

Анотація.....	4
Вступ.....	5
РОЗДІЛ I. ЗАГАЛЬНІ ПЕДАГОГІЧНІ ПІДХОДИ ДО ПОГЛИБЛЕНОГО НАВЧАННЯ ІНФОРМАТИКИ У ДЕВ'ЯТОМУ КЛАСІ	7
1.1. Сучасні аспекти поглибленого навчання інформатики в 9-ому класі загальноосвітніх навчальних закладів природничо- математичного і технічного (інженерного) спрямування	7
РОЗДІЛ II. МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ВИВЧЕННЯ ТЕМ.....	16
2.1. Інформаційні технології у суспільстві (3 год.)	16
2.2. Мережні технології (18 год.)	17
2.3. Основи інформаційної безпеки (4 год.).....	19
2.4. Створення персонального навчального середовища (6 год.)	24
2.5. Електронний документообіг (4 год.)	25
2.6. Комп'ютерні публікації (8 год.).....	28
2.7. Основи алгоритмізації та програмування (56 год.).....	29
2.8. Бази даних (16 год.)	49
ЛІТЕРАТУРА.....	63

Анотація

Методичний посібник «Інформатика 9» розроблено відповідно до навчальної програми з інформатики («Інформатика, 8-9 класи загальноосвітніх навчальних закладів з поглибленим вивченням інформатики», що рекомендована для переважного використання у спеціалізованих школах із поглибленим вивченням предметів технічного (інженерного) циклу і спеціалізованих школах, гімназіях, ліцеях, колегіумах, класах з поглибленим вивченням окремих предметів, зокрема природничо-математичного і технічного (інженерного) спрямування. (Додаток 5 і Додаток 8 до Наказу МОН молодьспорт України № 409 лист від 03.04.2012 року «Про затвердження Типових навчальних планів загальноосвітніх навчальних закладів II ступеня»).

У посібнику розглядаються методичні підходи до навчання інформатики в 9-му класі в школах з поглибленим вивченням предметів природничо-математичного циклу, наводяться навчальна програма, за якою вивчається даний курс, методичні рекомендації щодо організації та викладання основних тем курсу з прикладами розробок деяких уроків інформатики у 9-му класі.

Посібник складається з двох розділів «Загальні педагогічні підходи щодо поглибленого навчання інформатики в 9-ому класі основної школи» та «Методичні рекомендації щодо вивчення тем з поглибленого навчання інформатики в 9-ому класі основної школи».

У першому розділі розглядаються сучасні аспекти навчання поглибленого курсу інформатики у 9-ому класі загальноосвітніх навчальних закладів з поглибленим вивченням предметів природничо-математичного циклу. Зміст першого розділу є теоретичною основою методичних рекомендацій щодо вивчення поглибленого курсу інформатики в 9-ому класі основної школи. У цьому розділі автори аналізують інформатику як навчальний предмет у загальноосвітній школі, розглядають цілі й завдання вивчення інформатики в 9-ому класі в процесі поглибленого навчання інформатики в основній школі і основний наголос роблять на підходах до вивчення інформатики. Інформатична освіта базується на компетентнісній парадигмі, метою і результатом застосування якої є сформованість інформаційно-комунікаційної компетентності учнів.

Другий розділ містить рекомендації щодо викладання поглибленого курсу інформатики у 9-ому класі. У розділі наводяться приклади розробок деяких уроків з інформатики у 9-ому класі, які містять завдання для узагальнення та систематизації окремих тем з поглибленого вивчення інформатики.

Метою посібника є надання методичної допомоги вчителям, які викладають курс інформатики в сучасних умовах.

Запропоновані методичні рекомендації та приклади розробок уроків допоможуть учителям в організації освітнього процесу, оновлення форм і методів навчання на засадах компетентнісного, діяльнісного й особистісно-орієнтованого підходів. Науково обґрунтовані методичні напрацювання цінні тим, що вони є не просто ідеями, а мають практичне втілення.

Посібник стане у нагоді вчителям основної школи, а також тим, хто цікавиться організацією і перспективами навчання інформатики у школі.

Вступ

Перехід до інформаційного суспільства, з одного боку, і спрямованість його на ринкові відносини в Україні, з іншого, сформувавши умови, які торкнулися всіх сфер життя людини. Зазначене викликало потребу перетворення системи освіти з унітарної (авторитарної, трансляційної) на демократичну, гуманітарну, яка спрямована, насамперед, на розвиток кожного школяра — майбутнього члена суспільства. Неухильний розвиток технологій нині ставить також перед освітою нові завдання, починаючи від систематизації світових наукових і технічних досягнень на рівні змісту, приведення у відповідність до них форм, методів і технологій навчання й закінчуючи залученням їх до сфери освіти з інших, непедагогічних сфер світового суспільного простору. Після прийняття нової парадигми освіти (гуманістичної, гуманітарної й особистісно-орієнтованої) і розповсюдження досліджень у галузі педагогіки, об'єктом яких є оновлені форми навчання, педагогічних методів і технологій інноваційного характеру, розвитком науки управління в цілому, постала потреба корегування завдань, що стоять перед розвитком системи освіти в Україні в цілому і перед загальноосвітніми закладами освіти зокрема [2].

Реформування системи освіти полягає не лише в оновленні змісту шкільних предметів. Для вчителя реформування освіти полягає в першу чергу в оновленні методик навчання, осучасненні форм і методів, що використовуються на уроці й позаурочній діяльності.

Відомо, що у галузі високих технологій суттєве оновлення технічних засобів відбувається протягом 3–5 років. Скорочується також і термін використання конкретних інформаційних технологій. Учитель інформатики змушений безперервно слідкувати за розвитком засобів обчислювальної техніки, за появою нових програмних засобів, систем програмування. Необхідність кожні три роки розпочинати майже з нуля не знайома викладачам інших дисциплін. Навіть прикладне програмне забезпечення загального призначення нині змінюється настільки швидко, що неможливо в межах одного предмета ознайомити учнів з роботою навіть більшості таких програмних продуктів. За цих умов особливо має зростати роль фундаментального складника освіти.

Сучасні зміни змісту і структури загальної середньої освіти мають глибинний характер і потребують оновлення всіх складових навчально-виховного процесу. Зрозумілим є й постійне оновлення методичних розробок. Нині настала потреба шукати нові концептуальні підходи до методики навчання користування програмними засобами і необхідність змін у підходах до викладання загальних принципів побудови і функціонування ІКТ. Головним має стати не знання, як виконати ту чи іншу операцію, а розуміння її сутності. Це дозволить учням у подальшій своїй діяльності на базі такого роду знань і навичок досить швидко і безболісно сформувати навички роботи зі спеціалізованим програмним забезпеченням, необхідним у професійній діяльності саме на поточний момент, а згодом й оволодіти його наступними модернізаціями або різновидами. Тобто потрібно формувати загальні концепції, змістові уявлення про роботу з програмним забезпеченням, що і реалізуюватиме фундаменталізм інформатики в поглибленому шкільному курсі [4].

Основним завданням курсу інформатики в 9-ому класі основної школи є формування предметної інформаційно-комунікаційної компетентності як необхідної умови ефективної самореалізації учня в умовах сучасного високотехнологічного суспільства.

Формування основних понять курсу інформатики є досить тривалим процесом, особливість якого полягає в постійному зверненні до раніше засвоєного матеріалу. Така циклічність у вивченні основних понять, повернення до них кожного разу на новому, більш високому рівні пізнання, дозволяє досягати надійного засвоєння їх змісту. Водночас завжди слід мати на увазі головні цілі вивчення інформатики: загальноосвітні, розвивальні й практичні. Досягненню цих цілей буде сприяти дотримання таких методичних принципів, які слід закласти в програмах курсу: принцип системності, паралельності в освоєнні прикладної і фундаментальної складових курсу, самонавчання і взаємонавчання.

Можливими є два основні напрями розвитку курсу інформатики: математизація змісту навчання інформатики й розвиток формального компонента діяльності (центральними поняттями інформатики стають алгоритм і комп'ютер); побудова курсів інформатики від феномена інформації й інформаційних процесів до методів їх вивчення за допомогою інформаційних моделей.

Ці два підходи мають об'єктивний характер і відображають процеси, що відбуваються в усьому світі, але вони далеко не рівноправні з погляду знань, що здобуваються. Найперспективнішим є курс, що об'єднує ці два підходи на основі широкого застосування комп'ютерного моделювання: у 7-их класах — інформаційне моделювання в засобах ІКТ загального призначення, а з 8-го класу — основи комп'ютерного моделювання в середовищі табличного процесора і систем програмування мовою високого рівня.

У програмі курсу інформатики особливу увагу слід зосередити не навколо комп'ютерної діяльності та конкретних версій комп'ютерних програм, а на концептуальних основах інформаційно-комунікаційних технологій та інформаційних процесів, які лежать в основі цих технологій. В умовах насиченості змісту нинішнього курсу інформатики постає проблема поєднання вивчення змісту нового навчального матеріалу з його закріпленням й систематизацією понять.

Автори посібника пропонують своє бачення вирішення цієї проблеми. Пропонований курс спрямовано на подальше формування базових компетентностей використання інформаційно-комунікаційних технологій. Разом з тим, автори вважають, що формування цілісного курсу інформатики в 9-их класах має відбуватися на основі інтеграції змісту навчання навколо використання ефективних способів роботи з інформацією, наповнення навчального матеріалу гуманітарною складовою, адекватного відображення в шкільному курсі сучасного стану фундаментальної науки інформатики, розкриття емоційно-ціннісних і моральних засад, формування і розвитку критичного мислення, активного використання внутрішньо предметних і міжпредметних зв'язків курсу інформатики, навчання узагальнених способів застосування сформованих знань й умінь на практиці [2].

Отже, основними завданнями посібника є:

- ознайомити вчителів з особливостями програми поглибленого навчання інформатики [1];
- допомогти у постановці методичних цілей формування інформаційно-технологічних умінь під час вивчення окремих тем курсу;
- допомогти в узагальненні змісту певної теми, тобто виокремлення інформаційно-технологічних умінь, знань та навичок, необхідних для опанування інформаційних технологій;
- ознайомити учителів з методичними прийомами проведення уроків інформатики..

РОЗДІЛ І. ЗАГАЛЬНІ ПЕДАГОГІЧНІ ПІДХОДИ ДО ПОГЛИБЛЕНОГО НАВЧАННЯ ІНФОРМАТИКИ У ДЕВ'ЯТОМУ КЛАСІ

1.1. Сучасні аспекти поглибленого навчання інформатики в 9-ому класі загальноосвітніх навчальних закладів природничо-математичного і технічного (інженерного) спрямування

У сучасному шкільному курсі інформатики можна виокремити три базові змістові лінії: теоретичної інформатики, основ алгоритмізації та програмування, інформаційно-комунікаційних технологій. Історично з'явившись як освітній предмет, у змісті якого основна увага приділялась основам теоретичної інформатики та алгоритмізації, цей курс з об'єктивних обставин (у зв'язку з розвитком інформаційно-комунікаційних технологій) поступово набував все більш технологічного змісту. Постійне оновлення фактичного наповнення потребує багато часу на його опанування вчителями, тому питанню власне методики викладання приділяється, на жаль, недостатньо уваги.

Разом з тим викладання тем, пов'язаних з питаннями теоретичної інформатики та основ алгоритмізації, не потребує розроблення нових методичних засад. Усталена методика викладання шкільних предметів природничо-математичного циклу – пояснення нового матеріалу, відтворення його учнями, постановка та розв'язання навчально-практичних задач, контроль знань і вмінь учнів – майже повністю забезпечує належний рівень засвоєння змісту освіти з цих предметів [3].

Інформатика — фундаментальна наука про методи, засоби й технології опрацювання інформації. Інформатика є теоретичною платформою розроблення й використання інформаційних і комунікаційних технологій, які є одним із найголовніших досягнень сучасної цивілізації і важливою продуктивною силою сучасного суспільства.

Розвиток інформаційного суспільства зумовлює потребу у нових підходах до навчання інформатики учнів загальноосвітніх навчальних закладів. В умовах швидкого розвитку інформатизації шкільний курс інформатики набуває нового функціонального призначення, спрямованого на формування інформаційної культури, інформаційної компетентності, усвідомлення учнями ролі інформаційних технологій у розвитку сучасного суспільства. Під час навчання в учнів мають бути сформовані як теоретична база знань з основ інформатики, так і вміння й навички ефективного використання сучасних комп'ютерно-інформаційних технологій у навчально-пізнавальній і майбутній професійній діяльності.

Складниками мети навчання, визначеної Державним стандартом щодо галузі «Технології», які безпосередньо пов'язані з навчанням інформатики, є такі завдання:

- формування в учнів уявлень щодо предметно-перетворювальної діяльності людини, світу професій, шляхах отримання, зберігання інформації та способів її опрацювання; здатності до формулювання творчих задумів, усвідомленого дотримання безпечних прийомів роботи та користування інструментами і матеріалами;

- розвиток пізнавальної, художньої і технічної обдарованості, технічного мислення у процесі творчої діяльності, уміння користуватися технічною термінологією, інформацією, поданою художньо та графічно, вміння

використовувати комп'ютер як засіб продуктивної діяльності;

– формування готовності до виконання побутових завдань шляхом застосування алгоритмів та застосування навичок технологічної діяльності у практичних ситуаціях.

Інформатика (найближчий англомовний аналог назви — «Computer Science») — фундаментальна наука про методи, засоби й технології опрацювання інформації, що разом з речовиною й енергією належить до основних понять, на яких будується сучасна наукова картина світу. Інформатика є теоретичною платформою розроблення й використання інформаційних і комунікаційних технологій (ІТ, ІКТ), які є одними із найголовніших досягнень сучасної цивілізації і важливою продуктивною силою сучасного суспільства [5].

Значна частина цілей навчання інформатики є надпредметними, тобто такими, що передбачають формування ключових компетентностей. Зокрема, до таких цілей можна зарахувати формування властивостей особистості, які називають «комп'ютерна грамотність», «інформаційна культура», «computer skills», «abilities», які є необхідними передумовами соціалізації особи у сучасному інформатизованому суспільстві.

Навчання в загальноосвітніх навчальних закладах нині ґрунтується на *компетентісному підході*, відповідно до якого кінцевим результатом навчання інформатики мають стати предметна інформатична та ключові компетентності, зокрема інформаційно-комунікаційна, навчальна, комунікативна, математична, соціальна, громадянська, здоров'язберезувальна.

Оновлення навчальних програм, зокрема навчальної програми курсу інформатики основної школи, нині здійснене таким чином, щоб урахувати необхідність переходу від вузько предметного визначення змістового наповнення процесу й результатів навчання до визначення його результатів на основі компетентісного підходу. У термінах компетентісного підходу до результатів навчання інформатики зазначене описується як формування ключових компетентностей, зокрема цифрової обчислювальної, загальнонаукової і загально технологічної, виокремлених документами Ради Європи як компетентності для навчання протягом життя. Важливою складовою цілей навчання інформатики в школі є формування в учнів компетентностей в основах наук. Адекватне сприйняття багатьох явищ, які відбуваються в сучасному суспільстві, неможливе без ознайомлення з основами програмування, принаймні на рівні описань і виконання простих алгоритмів. Ці компетентності мають бути сформовані на основі набутих у процесі навчання знань, умінь і навичок, досвіду навчальної та життєвої діяльності, вироблених ціннісних орієнтирів.

Інформаційно-комунікаційна компетентність (ІКТ-компетентність) як ключова — це здатність ефективно використовувати інформаційно-комунікаційні технології (ІКТ) у навчальній, дослідницькій і повсякденній діяльності задля вирішення задач, які виникають у реальному житті. Її формування в учнів відбувається як результат навчання інформатики як навчального предмета інваріантної частини навчального плану, навчання курсів за вибором інформатичного, інформаційно-технологічного (ІТ) спрямування та застосування ІКТ в процесі навчання інших предметів, реалізації діяльносного, особистісно орієнтованого та компетентісного підходів у навчально-виховному процесі. Формування ключових компетентностей, зокрема цифрової обчислювальної, загальнонаукової і загально технологічної є предметною частиною мети навчання інформатики. У цьому сенсі вона розгляда-

ється як здатність особи застосовувати в конкретній життєвій або навчальній ситуації набуті знання, уміння, навички, способи діяльності щодо добору та застосування відповідних інформаційних технологій і засобів задля пошуку необхідних відомостей, їх аналізу, організації, перетворення (опрацювання), зберігання, передавання з дотриманням правових й етичних норм. Зазначені дії необхідні для виконання завдань предметної галузі (не обов'язково інформатики) або реального життя, у якому зазвичай виникають завдання, предметну галузь яких визначити досить складно, оскільки вони охоплюють діапазони галузей знань від етики і психології до математики й фізики.

Навчальний предмет «Інформатика» за своєю сутністю є інтегративним, його зміст формується як результат аналізу широкого кола галузевих знань. Чинними програмами, які є результатом творчої роботи груп висококваліфікованих у галузі навчання інформатики фахівців, відображено доступний для засвоєння учнями відповідних вікових категорій зміст, орієнтований на формування як предметних (галузевих) компетентностей, так і на створення підґрунтя для формування надпредметних компетентностей, відображення змістових ліній, що інтегрують знання багатьох предметів і створюють передумови ефективної соціалізації особи в інформаційному суспільстві [5].

Інформатичні компетентності забезпечують виконання особою суспільно корисної перетворювальної діяльності, тому компетентісно орієнтоване навчання основ інформатики (теоретичного компонента) і технологій (діяльничого компонента) нині є суттєвим для формування і розвитку загальної культури суб'єктів навчання, їхніх ціннісно-сміслових установок, здатностей до ефективної соціально-комунікативної взаємодії через мотивацію самоактуалізації, уміння організувати й оцінювати власну навчальну діяльність, використовувати й подавати відомості в графічно-образній формі, спілкуватися, вести конструктивний діалог тощо.

У чинних навчальних програмах поєднано знаневі й діяльнісні компоненти вимог до результатів навчання з компетентнісними компонентами.

Відповідно до вимог чинного Державного стандарту базової і повної загальної середньої освіти зміст предмета «Інформатика», незалежно від конкретизації змісту у навчальних програмах, будується за такими *змістовими лініями*:

- інформація, інформаційні процеси та системи;
- комп'ютер як універсальний пристрій для опрацювання даних;
- комп'ютерні мережі й інфокомунікації;
- комп'ютерне моделювання;
- алгоритмізація й основи програмування;
- базові інформаційні технології.

У навчанні інформатики необхідним є також дотримання наскрізних змістовних ліній освіти, а саме: «Екологічна безпека та сталий розвиток», «Громадянська відповідальність», «Здоров'я і безпека», «Підприємливість та фінансова грамотність», які відображають провідні соціально й особистісно значущі ідеї, визначені у модернізованих навчальних програмах та мають послідовно розкриватися у змісті навчання й плануванні навчально-виховного процесу інформатики.

Удосконалення процесу навчання інформатики і покращання його результатів для учнів як загальноосвітніх, так і спеціалізованих навчальних закладів, може бути здійснене шляхом посилення вимог до фундаментальних складників результатів навчання. Зазначене надасть можливість більш

ефективно формувати ключові компетентності, зокрема шляхом забезпечення для учнів можливості освоювати роботу з новими засобами об'єктно-перетворювальної діяльності, засобами міжособистісної комунікації тощо [1].

За структурою процес навчання інформатики подібний до процесів навчання математики, природничих дисциплін, тому бажано максимально можливо гармонізувати і синхронізувати (на рівні навчального закладу) навчання інформатики з навчанням математики, фізики, хімії, біології, географії, що надасть можливість проведення інтегрованих уроків і виконання міжпредметних проектів.

Разом з математикою, фізикою, хімією, біологією курс інформатики закладає основи природничо-наукового світогляду. Інформатика має велику кількість міжпредметних зв'язків, причому як на рівні понятійного апарату, так і на рівні інструментарію. Багатопредметні знання і способи діяльності (включаючи використання засобів ІКТ), освоєння учнями на базі інформатики, знаходять застосування як у рамках освітнього процесу під час вивчення інших предметів, так і в інших життєвих ситуаціях, стають значущими для формування якостей особистості, тобто орієнтовані на формування метапредметних і особистісних результатів.

Вивчення інформатики у 9-их класах вносить значний вклад у досягнення головних цілей основної загальної освіти, сприяючи:

- формуванню цілісного світогляду, відповідного сучасному рівню розвитку науки і суспільної практики, завдяки розвитку уявлень про інформацію як найважливішому стратегічному ресурсу розвитку особистості, суспільства; розуміння ролі інформаційних процесів в сучасному світі;
- удосконаленню загальнонавчальних і загальнокультурних навичок роботи з інформацією в процесі систематизації й узагальнення наявних і отриманих нових знань, умінь і навичок в галузі інформатики та інформаційно-комунікаційних технологій; розвитку навичок самостійної навчальної діяльності школярів (навчального проектування, моделювання, дослідницької діяльності тощо);
- вихованню відповідального та вибіркового ставлення до інформації з урахуванням правових і етичних аспектів її поширення, вихованню прагнення до продовження освіти і творчої діяльності із застосуванням засобів ІКТ.

Складність принципів функціонування засобів ІТ, неможливість подання вичерпного опису їх роботи без залучення знань, недоступних для сприйняття людиною без спеціальної підготовки, свого часу викликала відмову від намагання хоч якось пояснювати їх, що було відображено в концепції "користувацького підходу" до навчання інформатики у загальноосвітніх навчальних закладах. Сучасний стан розвитку засобів ІТ характерний тим, що значна частина їх засобів (гаджетів і відповідного програмного забезпечення) не вимагає від користувачів спеціального навчання користування ними. Разом з тим, сучасним учням 9-го класу вже доступні багато з можливих пояснень принципів функціонування засобів ІТ. Також слід ураховувати, що використання засобу діяльності (у нашому випадку — комп'ютера) тоді є найбільш ефективним, коли людина розуміє основні принципи його функціонування. Цього можна досягти, якщо подати всім учням максимально спрощені, але достовірні відомості щодо зазначених принципів — це досить важко, не всі учні повністю це зрозуміють, але всім має бути надано шанс цього досягти. Важливість поглибленого вивчення інформатики

у загальноосвітніх навчальних закладах нині посилюється кількома чинниками, найважливішими з яких, безумовно, є економічні. Трансформація суспільства, спрямована на його інформатизацію, для України відображається в тому, що за першу половину 2015 року українські ІТ-компанії експортували товарів і послуг на більш ніж 700 мільйонів доларів. У 2016-2020 рр. очікується, що сукупні податкові надходження від ІТ до державного бюджету складуть 36 млрд. гривень, а в 2020 році внесок ІТ-індустрії у ВВП країни досягне 5,7% [5].

Зміст навчального предмету забезпечує опанування учнями наукових основ інформатики, відповідних фундаментальних понять, принципів побудови й функціонування засобів інформаційних і комунікаційних технологій. Завдання навчання предмету інформатика в основній школі полягають у тому, щоб:

- ✓ сформувати в учнів базові знання про інформацію та інформаційні процеси, значення інформації і знань на сучасному етапі розвитку інформаційного суспільства;

- ✓ надати учням основні історичні відомості про розвиток інформатики і засобів обчислювальної техніки, внесок зарубіжних і вітчизняних учених у їх розвиток;

- ✓ сформувати в учнів уміння використовувати програмні й інформаційно-комунікаційні засоби в навчальній та повсякденній діяльності;

- ✓ забезпечити оволодіння учнями основами інформаційної культури;

- ✓ сформувати вміння творчо виконувати навчальні завдання, розробляти раціональні алгоритми виконання і здійснювати аналіз їх виконання;

- ✓ сформувати в учнів початкові навички програмування, уміння налагоджувати програми й аналізувати отримані результати.

Виконання зазначених завдань забезпечує формування в учнів наукового світогляду, інформаційної культури, алгоритмічного й критичного стилів мислення, розвитку творчих здібностей, умінь і навичок працювати із сучасними засобами інформаційних і комунікаційних технологій.

Набуті учнями в процесі навчання інформатики знання, уміння і навички забезпечать підґрунтя для формування в учнів предметної інформатичної компетентності і ключових компетентностей, передбачених Державним стандартом базової і повної загальної середньої освіти.

Набуті учнями в результаті навчання компетентності надають їм змогу:

- ✓ пояснювати властивості інформації й закономірності інформаційних процесів;

- ✓ оволодіти основними методами наукового пізнання;

- ✓ розуміти наукові основи опрацювання відомостей, застосовувати основні поняття, пов'язані з алгоритмізацією опрацювання даних, управління об'єктами і процесами;

- ✓ бути готовим до активної життєдіяльності в умовах інформаційного суспільства, стати в майбутньому не лише повноцінним членом такого суспільства, а й його творцем;

- ✓ використовувати набуті знання в подальшій навчальній та практичній діяльності в умовах інформаційного суспільства.

Концентричність реалізує ознайомлення учнів з поняттями інформатики і інформаційно-комунікаційними технологіями за всіма змістовими лініями на різних рівнях складності, поступово доповнюючи і розширюючи їх зміст залежно від рівня сформованості загально навчальних навичок, вивченого навчального матеріалу з інших предметів і вікових особливостей розвитку учнів відповідного віку. Таким чином забезпечується поступове

нарощування складності матеріалу, його актуалізація, повторення, закріплення, що сприяє формуванню предметної ІКТ-компетентності та ключових компетентностей.

Зміст навчання інформатики у 9-ому класі в процесі поглибленого вивчення інформатики структуровано за темами із визначенням кількості годин на їх вивчення. Учителю надається право корегувати послідовність вивчення тем залежно від методичної концепції, від конкретних навчальних ситуацій, від рівня підготовки учнів та сформованості у них ІКТ-компетентності. Водночас учитель не може порушувати порядок вивчення тем, між якими є змістові залежності. Запропонована кількість часу на вивчення кожної теми також є орієнтовною, вчитель може її змінювати.

З метою надання рівних можливостей долучення до сучасних технологій максимально можливій кількості учнів в Україні нині діють дві програми поглибленого навчання інформатики в основній школі.

Певна кількість навчальних закладів з поглибленим вивченням інформатики з 2014 року працює за програмою «Інформатика. 5–9 класи. Для навчальних закладів з поглибленим вивченням предметів природничо-математичного циклу», рекомендована МОН України, лист від 17.07.2013 р., №1/11-11636, чинна до 2018 р. (Комп'ютер у школі та сім'ї. — 2012. — №6. — С. 3–14). За цією програмою у 2016/2017 навчальному році вперше працювали восьмі класи (2 год. на тиждень). Вивчення більшості розділів і тем можна організувати, використовуючи підручники, призначені для вивчення інформатики за програмою поглибленого навчання інформатики у 8–9-х класах. У 2017/2018 навчальному році розпочинається поглиблене вивчення інформатики у 9-ому класі за навчальною програмою «Інформатика. 8-9 класи загальноосвітніх навчальних закладів з поглибленим вивченням інформатики», «Затверджено МОН України». Рішення Колегії МОН України від 27.06.2013 року протокол №2/4-2. Навчальною програмою передбачено вивчення поглибленого курсу інформатики 140 год.. 4 години на тиждень вивчення інформатики за рахунок двох годин з інваріантної складової навчального плану та двох годин з варіативної складової. Решту годин варіативної частини навчального плану рекомендується використовувати для навчання (у формі факультативів) розділів математики, які необхідні для кращого сприйняття учнями навчального матеріалу інформатики [1].

Зміст навчального предмета «Інформатика» містить фундаментальну складову, яка реалізується шляхом вивчення основ науки інформатика, має прикладну спрямованість, що реалізується під час виконання відповідних тематичних практичних робіт, виконання індивідуальних і групових навчальних проєктів. Опанування змістом навчального матеріалу з інформатики у 9-ому класі в процесі поглибленого вивчення інформатики забезпечується застосування різних форм (індивідуальної, парної, груповою та колективною) організації діяльності учнів та інноваційних методів навчання.

Успішне впровадження навчального курсу «Інформатика» залежить від обов'язкової саме для нього складової – стандарту можливостей для навчання, в якомузначаються обов'язкові умови та ресурси, потрібні для реалізації державних вимог до рівня загальноосвітньої підготовки учнів середнього шкільного віку.

До обов'язкових умов успішного впровадження курсу належать:

- підготовленість учителів інформатики до поглибленого викладання курсу «Інформатика»;
- забезпечення навчального закладу сучасною комп'ютерною технікою

й програмним забезпеченням відповідно до чинних норм і специфікацій навчальних комп'ютерних комплексів (НКК);

- під'єднання до Інтернету всіх комп'ютерів НКК, якість якого забезпечує виконання кожним учнем завдань роботи із сервісами Інтернету, та наявність у навчальному закладі локальної комп'ютерної мережі.

При вивченні предмета кожний урок проводиться із використанням комп'ютерів, тому на кожному уроці класи діляться на підгрупи так, щоб кожен учень був забезпечений індивідуальним робочим місцем за комп'ютером, але не менш як 8 учнів у підгрупі.

Перелік необхідних програмних засобів:

- операційна система з графічним інтерфейсом;
- програма для запису даних на оптичні носії;
- архіватор;
- антивірусна програма;
- векторний графічний редактор;
- растровий графічний редактор;
- текстовий процесор; • редактор презентацій;
- редактор публікацій;
- програми для опрацювання об'єктів мультимедіа;
- табличний процесор;
- система управління базами даних; програма для створення карт знань;
- програми для опрацювання аудіо- та відеоданих і розроблення поточкових презентацій;
- електронні словники та програми-перекладачі;
- система розпізнавання текстів;
- веб-браузер;
- навчальне середовище виконання алгоритмів;
- програми для розвитку логічного та критичного мислення;
- розвиваючі програми;
- комп'ютерні програми для підтримки вивчення різних навчальних предметів;
- системи програмування.

Вибір певних операційних систем, програмних та апаратних платформ, програмних засобів здійснює вчитель.

Програмою поглибленого вивчення інформатики передбачено розширення і поглиблення змісту теми алгоритмізації і програмування, вивчення якого продовжується у 9-му класу. Саме це дасть можливість вчасно здійснювати ґрунтовну і якісну підготовку учнів до участі в олімпіадах, конкурсах, турнірах, науково-практичних конференціях, конкурсах-захистах науково-дослідницьких робіт різного рівня.

Вивчення підрозділу: "Логічні елементи. Фізичні втілення логічних елементів." не передбачене програмою 8 – 9, але досить важливе для навчання у закладах навчання, які мають на меті підготовку досить широкого науково-інженерного (STEM – спрямування). Тому оглядове подання відповідного навчального матеріалу має бути здійснене на прикладах відомих суб'єктам навчання з життєвого досвіду застосувань логічних пристроїв — замків, систем охорони, систем керування об'єктами (їх рухом, температурою тощо). Якщо у школі викладається (за рахунок годин варіативної частини навчального плану) курс "Робототехніка", або подібний, можна провести кілька інтегрованих уроків.

Досвід роботи шкіл (класів) з поглибленим вивченням інформатики свідчить, що найбільш вдалою формою організації навчального процесу в них є лекційно-практична система, в якій виділяються лекції, практикум з розв'язування задач, семінари та метод проєктів. Причому більше уваги, ніж у звичайних класах, приділяється організації різних видів самостійної роботи учнів. Ураховуючи різні умови для навчання інформатики й інформаційні технології у загальноосвітніх навчальних закладах, у навчальній програмі для кожного класу передбачено до 15% резервного часу. Ці години вчитель може самостійно розподіляти між розділами навчальної програми, використовувати для виконання додаткових навчальних проєктів (у тому числі інтегрованих за змістом з навчальними предметами, що вивчаються поглиблено), проведення інтелектуальних конкурсів і творчих змагань, за рахунок зазначених годин вводити нові теми, зокрема для здійснення допрофільного навчання.

Під час вивчення поглибленого курсу інформатики в 9-ому класі обов'язковим є проведення 24-ох практичних робіт. Зміст таких завдань треба добирати так, щоб тривалість їх виконання не перевищувала 25 хвилин (згідно з санітарними нормами щодо тривалості безперервної роботи за комп'ютером учнів цієї вікової категорії). Виконання учнями практичних робіт на комп'ютері є важливою складовою уроку інформатики. Їх мета може бути різною: формування позитивної мотивації та актуалізації знань; формування вмінь, навичок, здібностей; поточне оцінювання навчальних досягнень тощо.

Практичні роботи, вказані в програмі, можуть виконуватися учнями як індивідуально, так і в групах, відповідно до обраної вчителем методики. Учитель може самостійно визначати форму проведення цих робіт (лабораторні роботи, практикуми, навчальні проєкти, колективна робота в Інтернеті тощо). Лабораторні роботи рекомендуються для самостійного виконання за індивідуальними завданнями. Результатом виконання лабораторної роботи є захист її учнем. Захист лабораторної роботи, на вибір учителя, може бути проведено індивідуально з окремими учнями, або у груповому інтерактивному режимі. Лабораторні роботи передбачено лише з тих тем, які дозволяють учням провести відповідну дослідницьку роботу [4].

У завданнях до практичних робіт вчителю слід передбачати використання актуального для учнів змістовного матеріалу й завдань з інших предметних областей. Також під час вивчення поглибленого курсу інформатики вчителю рекомендується здійснювати тематичне оцінювання в кінці кожної теми навчальної програми, об'єднуючи роботи з невеликих тем (до 5 годин) із наступними підсумковими роботами на кожному 8-10 уроці, а кількість практичних робіт з обов'язковим оцінюванням рекомендуємо встановлювати на рівні 25% від загального обсягу навчального часу, який відводиться на вивчення предмету.

Складність принципів функціонування засобів ІТ, неможливість подання вичерпного опису їх роботи без залучення знань, недоступних для сприйняття людиною без спеціальної підготовки свого часу викликала відмову від намагання хоч якось пояснювати їх, що було відображено в концепції «користувацького підходу» до навчання інформатики у загальноосвітніх навчальних закладах. Сучасний стан розвитку засобів ІТ характерний тим, що значна їх частина не вимагає від користувачів спеціального навчання користування ними. Разом з тим сучасним учням 9 класу вже доступні багато з можливих пояснень принципів функціонування засобів ІТ. Ураховуючи, що використання засобу діяльності (у нашому випадку –

комп'ютера) є найбільш ефективним тоді, коли людина розуміє основні принципи його функціонування, доречно подавати учням максимально спрощені, але достовірні відомості щодо зазначених принципів – це досить важко, не всі учні повністю зрозуміють їх, але всім має бути надано шанс цього досягти.

Оцінювання навчальних досягнень учнів з інформатики протягом навчання здійснюється шляхом тематичного оцінювання. Значення оцінки за тему рекомендуємо визначати так: розрахувати середній бал на основі поточних оцінок, а потім додати його до оцінки за підсумкову роботу, якщо така була проведена і поділити на два. Такий алгоритм пропонується тому, що оцінку за підсумкову роботу не можна прирівнювати до оцінки за поточну роботу, оскільки зміст завдань підсумкової роботи за визначенням відображає зміст розділу (розділів), а її оцінювання є комплексним оцінюванням відповідних навчальних досягнень.

Для ефективного контролю успішності учнів не досить лише виявити, що вони знають й уміють. Оцінювання учня має складатися з двох компонентів — числового балу, який фіксує результат перевірки знань, умінь і навичок школярів, рівня сформованості компонентів компетентностей, і вербального оцінного судження, яке характеризує якість навчальної діяльності учнів, ставлення до навчання, старанність. Обґрунтовуючи оцінку, вчитель повинен аналізувати виявлені знання за формою, змістом, обсягом, а також, що не менш важливо, вказує на прогалини та хиби в знаннях (за їх наявності) і надати рекомендації щодо їх виправлення. Доведення цієї частини оціночного судження до учнів має здійснюватися на етапі «оголошення та обґрунтування оцінок» уроку; в процесі аналізу самостійної (практичної, лабораторної, контрольної) роботи; на індивідуальних консультаціях тощо.

При оцінюванні навчальних досягнень учнів враховуються: характеристики відповіді: правильність, цілісність, повнота, логічність, обґрунтованість; якість знань: осмисленість, глибина, гнучкість, дієвість, системність, узагальненість, міцність; сформованість ключових та предметних компетенцій; рівень володіння розумовими операціями: аналізом, синтезом, порівнянням, абстрагуванням, класифікацією, узагальненням тощо; розвиток творчих умінь (уміння виявляти проблеми, формулювати гіпотези, перевіряти їх).

У процесі проведення уроків з інформатики слід приділяти увагу патріотичному вихованню учнів, розкривати досягнення вітчизняної науки, українських вчених у розробленні обчислювальної техніки і фундаментальних основ кібернетики та інформатики.

Під час підготовки вчителів до уроків радимо використовувати періодичні фахові видання: «Комп'ютер в школі та сім'ї», «Інформатика та інформаційні технології в навчальних закладах», «Інформатика в школі».

РОЗДІЛ II. МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ВИВЧЕННЯ ТЕМ

2.1. Інформаційні технології у суспільстві (3 год.)

Основні терміни та визначення:

Авторське право – галузь права інтелектуальної власності; покликане захищати лише зовнішню форму вираження об'єкта (літературні, художні, аудіо-, відеотвори, комп'ютерні програми, бази даних, карти, фотографічні твори), тобто їх «матеріальне втілення».

Дані – подання фактів, понять або інструкцій у формі, прийнятній для спілкування, інтерпретації, обробки людиною або за допомогою автоматичних засобів

Знання – зафіксована й перевірена практикою інформація, яка використовува-лася і може багаторазово використовуватися для прийняття рішень.

Інформаційна грамотність – здатність розвивати, розмішувати, оцінювати ін-формацію і найбільш ефективно її використовувати.

Інформаційна етика – розглядає проблеми власності, доступу, недоторканно-сті приватного життя, безпеки та спільності інформації.

Інформаційна культура – знання, вміння та навички ефективного пошуку і правомірного використання потрібної інформації.

Інформаційне суспільство – це суспільство, в якому діяльність людей щонай-більше зосереджується на опрацюванні інформації, а матеріальне виробництво та виробництво енергії здійснюють машини.

Інформаційно-комунікаційна компетентність – здатність особистості вико-ристовувати на практиці інформаційно-комунікаційні технології для задоволення власних індивідуальних потреб і розв'язання завдань – (ІКТ-компетентність).

Плагіат – привласнення авторства на чужий твір або на чуже відкриття, винахід чи раціоналізаторську пропозицію, а також використання у своїх працях чужого твору без посилання на автора.

Право інтелектуальної власності – закріплені законом права на результат ін-телектуальної діяльності в промисловій, науковій, художній, виробничій, інформа-ційній та інших галузях.

Прийняття рішень – вибір найкращого в деякому сенсі варіанта дії з можливих на підставі наявної інформації.

Фрілансінг (від англ. freelancer – вільний списоносець, найманець; у перенос-ному значенні – вільний художник) – праця за тимчасовими контрактами, переважно надомна. Поширена форма роботи ІТ-фахівців і журналістів.

За програмою поглибленого курсу інформатики тема «Інформаційні те-хнології у суспільстві» вивчається в 9-ому класі і є оглядовою. На неї за програмою відводиться всього 3 години. Основними завданнями вчителя під час вивчення цієї теми є надавання учням уявлення про інформаційну систему, інформаційні технології, апаратне та програмне забезпечення ін-формаційної системи, інформаційну культуру, інформаційну грамотність та ІКТ- компетентності тощо.

Задача вчителя – навчити учнів описувати різновиди інформаційних те-хнологій, інформаційних систем, інформаційні технології та їх признач-чення, інформаційні технології і системи в освіті.

Вивчаючи розділ, учителю доцільно побудувати урок не стільки на по-відомленні нового матеріалу, а на систематизації знань та класифікації ін-формаційних технологій, основних інформаційних процесів. Інформаційні

технології в освіті мають бути розглянуті на прикладах. Інформаційні технології розкривають нові можливості для розширення доступу до освіти. Дистанційне навчання дає змогу навчатися особам, які не можуть щодня відвідувати заклад освіти.

Інформаційне суспільство ставить перед освітою завдання підготовки громадян, які мають високий рівень інформаційної культури та здатні:

- адаптуватися в змінних умовах інформатизації життя і діяльності людей;
- самостійно отримувати та опрацьовувати необхідні знання;
- усвідомлювати, де і яким чином отримані знання можуть бути застосовані в житті;
- творчо і креативно мислити;
- бути комунікабельними та контактними;
- співпрацювати в групах як безпосередньо, так і дистанційно.
- Проблеми освіти, що розв'язуються засобами ІТ в інформаційному суспільстві:
 - забезпечення неперервності навчання протягом життя та доступності його з будь-якого місця та в будь-який час;
 - доступ до навчальних матеріалів із будь-якого місця та в будь-який час;
 - збільшення доступу до безкоштовного навчання;
 - зниження вартості навчання.
- Важливість інформаційних технологій в освіті визначається за такими якостями:
 - доступ до навчальних ресурсів;
 - мультимедіа підхід в організації навчання;
 - доступ до on-line-бібліотек;
 - зберігання навчальних та навчально-методичних матеріалів в електронних сховищах (застосування хмарних технологій);
 - можливість дистанційного навчання;
 - вільний доступ до джерел інформації;
 - доступ до відкритих навчальних курсів;
 - можливість спілкування учителя з учнями за використання двостороннього відео;
 - відкритий доступ до отримання освіти для людей з обмеженими можливостями.

Під час вивчення розділу "Інформаційно-комунікаційні технології в суспільстві" вчителю слід зосередитися на прикладних завданнях, безпосередньо пов'язаних з соціалізацією людини в інформатизованому суспільстві — плануванням поїздок, у тому числі з відвіданням історичних місць, пошуком вакансій, медикаментів у аптечній мережі тощо. Ознайомлення з геоінформаційними системами можна провести також у формі інтегрованого уроку інформатика + географія.

2.2. Мережні технології (18 год.)

Основні терміни та визначення:

DNS (Data Source Name) – система доменних імен, що забезпечує можливість використання для адресації вузлів мережі текстових імен замість числових кодів.

DNS сервери – програмно-апаратні засоби, які виконують перетворення текстових імен на числові коди.

Виокремлені лінії – лінії зв'язку, що забезпечують обмін повідомленнями без додаткових дій з фізичного комутування ліній зв'язку.

Віртуальний канал зв'язку – канал зв'язку, що утворюється перемиканням ліній зв'язку пристроями, керованими на рівні протоколів сеансового рівня.

Дейтаграма (англ. Datagram) – блок даних, що передається без попереднього налагодження з'єднання і створення віртуального каналу зв'язку.

Інтернет речей (англ. Internet of Things, IoT) – частина мережі Інтернет, що поєднує фізичні об'єкти (речі), тобто пристрої, які мають вбудовані пристрої перетворення сигналів і виконання команд, а також відповідне програмне забезпечення, яке здійснює первинне опрацювання, передавання й обмін даними між фізичним світом і комп'ютерними системами з використанням стандартних протоколів зв'язку (здебільшого – TCP/IP), мають IP-адреси.

Інфокомунікація – передавання даних із використанням цифрових засобів опрацювання інформації й каналів зв'язку.

Канал зв'язку – сукупність апаратно-програмних засобів кодування та декодування повідомлень і ліній зв'язку.

Комп'ютерна мережа – сукупність комп'ютерів, об'єднаних каналами зв'язку.

Комутовані лінії – фізичні лінії зв'язку (проводові, радіоканали) для утворення каналу зв'язку.

Комутування каналів – фізичне перемикання ліній зв'язку для утворення каналу зв'язку.

Комутування пакетів – створення віртуального каналу зв'язку для кожного окремого пакета даних з використанням службових даних, що містяться в пакеті (забезпечується протоколами TCP/IP).

Лінії зв'язку – середовище передавання сигналів у формі електричного струму, електромагнітного поля (радіохвиль, світла, у тому числі інфрачервоного діапазону), звукових або ультразвукових хвиль і відповідні апаратно-програмні засоби.

Постійне з'єднання – з'єднання через виокремлений канал (виокремлену лінію) зв'язку.

Провайдер – посередник між клієнтами та Інтернетом – організація або приватна особа, так званий ISP – Internet Service Provider – постачальник послуг Інтернету.

Телекомунікація – передавання даних з використанням ліній зв'язку.

Метою вивчення теми «Мережні технології» пояснити зміст понять URL-адреса, IP-адреса, доменне ім'я, протокол передавання даних, каналу зв'язку тощо. Сьогодні вміння здійснювати раціональний пошук інформації, користуватися електронними комунікаціями для обміну інформацією, спілкуватися у мережі вкрай необхідні в різних сферах життя і практичної діяльності людини. Тому тема «Мережні технології» є дуже важливою для формування інформаційно-технологічних умінь учнів та їхньої загальної інформаційної культури. У процесі вивчення цього розділу головними завданнями вчителя є:

- навчити учнів наводити приклади URL-адрес, IP-адрес, доменних імен, мережних протоколів, мережних пристроїв;
- навчити учнів описувати способи поєднання до Інтернету й алгоритм розширеного пошуку відомостей в Інтернеті;
- розвивати логічне й алгоритмічне мислення учнів;

- розвивати творчі здібності особистості;
- роз'яснити учням принципи функціонування комп'ютера і роботи з даними;

- навчити учнів добирати засоби для розв'язування типових задач;
- навчити учнів аналізувати і приймати правильні рішення, що є

Розділ має вивчатися досить глибоко, оскільки дуже багато його понять будуть новими навіть для учнів, які поглиблено вивчають інформатику. Особливо це стосується обміну даними в мережі. Необхідно зосередити увагу на суттєвих відмінностях між мережами першого покоління й сучасними, зокрема на переході від комутації ліній зв'язку до комутації пакетів. Безпосередньо пов'язані з цим адресування, система доменних імен, маршрутизація. Тут же слід розглянути проблему обмеження кількості доменних імен і способи її подолання, важливість збільшення адресного простору для Інтернету речей. Бажано вчителю продемонструвати роботу з IP-камерою як приклад речі з власною IP-адресою. Аналізуючи сервіси Інтернет, слід пояснити відмінності між поколіннями веб-ресурсів — від перших гіпертекстових документів мережі й до технологій веб-2.0 і наступних.

Зміст підрозділу «Типи каналів зв'язку і їх основні характеристики» є важливим для розуміння процесів, які відбуваються при передаванні даних у комп'ютерних мережах. Оскільки учні основної школи не мають математичної підготовки, достатньої для розуміння взаємозв'язку швидкості передавання даних і характеристик лінії зв'язку, протягом часу, відведеного на вивчення підрозділу, їм слід надати відомості щодо складових каналу зв'язку, їх фізичних реалізацій. Обов'язковим є формування в учнів базових понять щодо обов'язкових компонентів каналу зв'язку — первинних перетворювачів сигналу, кодерів — декодерів, кінцевих пристроїв (передавач — приймач), фізичної реалізації лінії зв'язку.

Розділ "Веб-технології" може викладатися як продовження відповідного розділу, що вивчався у сьомому класі. Ознайомлення учнів із соціальними сервісами й Інтернет-спільнотами слід проводити з обов'язковим наголосом на можливих негативних наслідках необережного спілкування у мережі. Бажано провести ознайомлення можливостями, що надає сервіс Фейсбук. Слід детально ознайомити учнів з Веб-сторінкою навчального закладу. Особливу увагу слід звернути на пошук навчальної інформації у Вікі-середовищах, роботу в них, пояснити технологію появи нових статей і їх редагування. Також корисно ознайомити дітей з новим сервісом Kiddle (Kiddle.co) — безпечним пошуковиком для дітей, який поки що працює тільки англійською.

Цікавим варіантом проведення такого ознайомлення може бути проведення інтегрованого уроку «інформатика + англійська мова».

2.3. Основи інформаційної безпеки (4 год.)

Основні терміни та визначення:

Cookie – у комп'ютерній термінології поняття, яке використовується для опису інформації у вигляді текстових або двійкових даних, отриманих від сайту на Веб-сервері, яка зберігається у клієнта, тобто браузера, а потім відправляється на той же сайт, якщо сайт повторно відвідати.

HTTPS (Hypertext Transport Protocol Secure) – протокол, який забезпечує конфіденційність обміну даними між сайтом і призначеним для користувача пристроєм.

Web-браузер – програмні засоби для роботи з гіпертекстовими документами World Wide Web.

Автентифікація – перевірка достовірності пред’явленого користувачем ідентифікатора.

Адміністративні засоби – правила користування інформацією, здатні доповнити законодавчі норми там, де це потрібно організації; вони, здебільшого, передбачають застосування інших видів захисту (технічного, програмного) і тільки в такому разі забезпечують досить надійний захист.

Атака – намагання реалізувати інформаційну загрозу.

Джерела загрози – потенційні зловмисники, природні або техногенні впливи, що можуть порушити цілісність, конфіденційність або взагалі знищити інформацію.

Засоби фізичного (технічного) захисту інформації – різного роду механічні, електро- або електронно-механічні пристрої, а також споруди і матеріали, призначені для захисту від несанкціонованого доступу та викрадення інформації, запобігання їй втрат у результаті порушення дієздатності компонентів комп’ютерної системи, стихійних лих, саботажу, диверсій тощо.

Зловмисник – особа, яка починає спробу атаки.

Інтранет – комп’ютерна мережа, призначена для управління інформацією всередині компанії або приватної домашньої мережі.

Інформаційна загроза – потенційна можливість певним чином порушити інформаційну безпеку, чи ступінь імовірності виникнення такого явища (події), наслідком якого можуть бути небажані впливи на інформацію.

Комп’ютерний вірус – певна сукупність виконуваного коду, яка може створювати свої копії (що не обов’язково збігаються з оригіналом) і вмищувати їх у файли, системні області комп’ютерів, комп’ютерні мережі; невелика програма, яка, потрапляючи в комп’ютер, спричинює небажані чи шкідливі дії.

Правові засоби захисту – чинні закони, укази та інші нормативні акти, які регламентують правила користування інформацією і відповідальність за їх порушення, захищають авторські права та регулюють інші питання використання комп’ютерних систем.

Резервне копіювання даних – створення копій важливої інформації, що є на персональному комп’ютері, для збереження її в інших сховищах даних (флеш-накопичувач, жорсткий диск, DVD-диск, хмарний сервіс тощо).

Спам – небажана кореспонденція.

Фішинг – спроба обманом отримати особисту інформацію користувача.

Метою вивчення теми є надання учням уявлення про необхідність захисту даних та інформаційних ресурсів, про загрози, які виникають в процесі використання Інтернету, про призначення антивірусних програм тощо.

Під час вивчення цієї теми вчителів варто приділити увагу вивченню класифікації загроз безпеці та пошкодженню даних у комп’ютерних системах, шкідливим програмам, їх типам, принципам дії та боротьби з ними, поняттям про спам і фішинг, засобам захисту від них тощо. Учні мають навчитися виконувати налаштування параметрів безпеки у веб-браузерах та операційній системі, вміти провести профілактику зараження комп’ютера шкідливими програмами.

У процесі вивчення розділу «Програмне забезпечення та інформаційна безпека» слід звертати увагу на те, що загрози безпеці та цілісності даних у комп’ютерних системах (КС) є наслідком недосконалості як технічного, так і соціального середовища, у якому функціонує КС. До таких загроз відносять можливість доступу сторонніх осіб до устаткування або помилок у програмному забезпеченні. на виконання наскрізної лінії «Громадянська відповідальність» окремо слід наголосити на етичних та правових засадах захисту відомостей і даних, вказати, що належить до норм поведінки, які традиційно склались або складаються в інформаційному суспільстві.

Зміст теми здебільшого знайомий учням, але слід наголосити на тому, що навчальним середовищем можна назвати систему, складові якої – інтелектуальна та матеріальна сприяють досягненню мети навчання, а саме – високої якості освіти. Інтелектуальна складова включає зміст навчання, навчальні технології та методики навчання, які застосовують викладачі, матеріальна – навчальні приміщення, засоби навчання, технічні засоби, підручники, посібники тощо. Необхідними компонентами інформаційно-освітнього середовища є навчальні матеріали, бази даних, ЕОР (підручники, посібники, словники, довідники, сайти, блоги тощо), засоби діагностики рівня навчальних досягнень та управління навчанням тощо. Персональним навчальним середовищем називатимемо сукупність начальних (інтелектуальних, матеріальних, технічних) ресурсів, які використовуються для організації особистої діяльності, навчання та самонавчання; за допомогою яких створюються необхідні умови діяльності, навчання та самонавчання. До персонального навчального середовища кожного студента априорі вже включено навчальне середовище навчального закладу, електронні ресурси Інтернет, власні інформаційні матеріали та інші складники, використання яких необхідне для навчальної діяльності.

Слід мати на увазі, що персональне навчальне середовище можна змінювати й доповнювати самостійно. Для підвищення мобільності й доступності персонального навчального середовища можна рекомендувати застосування Інтернет-середовищ, зокрема хмарних обчислень (англ. cloud computing) – хмарних технологій, їх ще називають просто «хмарами».

Формат хмарних технологій означає, що матеріали зберігаються в центрі опрацювання даних, а не на персональному комп'ютері – це забезпечує користувачеві чи групі користувачів доступ до документів та інших матеріалів через будь-який браузер із будь-яких пристроїв (комп'ютера, планшета, телефону та ін.), де є можливість виходу в Інтернет.

Хмарні технології (ХТ) – технології надання користувачеві через Інтернет обчислювальних ресурсів: від сховищ даних і прикладних програм до центрів комплексного опрацювання даних. Хмарою часто називають сервіс в Інтернеті, де можна зберігати різні дані, в тому числі світліни, музику, документи, відео (взагалі будь-які файли), отримуючи до них доступ з комп'ютера, телефону, телевізора або іншого пристрою, підключеного до Інтернету.

Інтернет-середовища – хмарні технології розрізняють за способами опрацювання даних і рівнями керування користувачем ресурсами хмари. Здебільшого хмарні середовища пропонують пакети: ІААS – інфраструктура як послуга – використання хмарної інфраструктури для самостійного управління ресурсами, мережами (операційні системи, платформи і прикладне програмне забезпечення); SААS (англ. Software as a service) – програмне забезпечення як послуга; PААS (англ. Platform as a Service) – платформа як послуга – бізнес-модель, призначена для надання доступу до хмарних обчислень, де користувач отримує доступ до використання усієї інформаційно-технологічної платформи.

Компанія, яка розробляє і розповсюджує продукцію та послуги під власною торговою маркою у хмарі, називається вендор (англ. vendor, продавець). Популярність ХТ для створення персонального навчального середо-

вища можна пояснити тим, що вони відкривають вільний доступ до програмного забезпечення (різних версій та розробників) із будь-якого місця знаходження; не потребують великих матеріальних затрат; не вимагають наявності необхідної ІКТ-інфраструктури в особистому (платному) користуванні.

Найбільш відомими вендорами є:

Amazon.com – Amazon web services (AWS) пропонує широкий асортимент продуктів – від онлайн-сховищ та інструментів обробки даних до програмного й технічного забезпечення, яке потрібне іншим компаніям, щоб пропонувати свої «хмарні» сервіси:

Amazon EC2 (Elastic Compute Cloud) – передавання та збереження даних, процесорний час, розширення кожного конкретного сервера тощо;

SIMPLEDB – сервіс бази даних з простим інтерфейсом;

Simple Storage Service (S3) – зберігання великих обсягів даних;

CloudFront – розподілена мережа зберігання і доставки контенту;

Elastic MapReduce – система оброблення та аналізу великих обсягів даних.

Microsoft Windows, який пропонує:

OneDrive – середовище для передавання, збереження даних, спільний доступ користувачів;

Office 365 – мережеві версії офісних додатків Word, Excel і PowerPoint;

Azure Services Platform – платформа для збереження даних і використання з Web-додатками на серверах, що містяться в дата-центрі Microsoft тощо.

Dropbox – популярний файлообмінник і ресурс збереження файлів, який є кросплатформним сервером (Windows, Mac, Linux та ін.). Забезпечує завантаження файлів на сервер Dropbox, які можна зробити доступними для інших осіб, здійснює синхронізацію файлів на локальному комп'ютері з файлами в он-лайн сховищі.

Google – Google Drive – призначений для зберігання файлів користувача, інтеграції з іншими інструментами, редагування документів у текстових форматах, форматах електронних таблиць тощо.

Аналізуючи можливості вендорів, можна вибрати найдоцільнішого для створення персонального навчального середовища. У виборі ХТ для побудови персонального навчального середовища та використання Інтернет-середовищ для створення й публікації документів (текстових, графічних, презентацій тощо) бажано дотримуватися таких етапів.

1. Виконати аналіз можливостей вендора ХТ, обґрунтувати їх доцільність і корисність у досягненні цілей персонального навчального середовища.

2. Оцінити якість відібраних ХТ, їх доступність, зрозумілість, а також умови використання, описані вендором.

3. Виконати порівняння та оцінювання характеристик і можливостей ХТ, їх значення для передбачуваних робіт.

4. Оцінити можливості ХТ впливу на підсилення передбачуваної вами діяльності.

5. Оцінити рівень безпечного користування ХТ.

6. Розробити структуру системи ХТ, яка передбачає інтеграцію хмарних сервісів різного призначення.

У поглибленому курсі інформатики метою навчання інформаційної безпеки (ІБ) є формування у старшокласників знань принципів забезпечення інформаційної безпеки, способів забезпечення надійного функціонування засобів інформаційних та комунікаційних технологій (ІКТ).

З розвитком ІКТ збільшується кількість загроз, розширюється і ускладнюється спектр засобів забезпечення ІБ, відповідно, і збільшується потреба у кваліфікованих кадрах у сфері інформаційної безпеки, тому метою навчання ІБ крім соціалізації, є профорієнтація і профільне навчання школярів в області інформаційної безпеки.

У ході навчання інформаційної безпеки важливим елементом соціалізації є розвиток у школярів здібностей до самоврядування, саморегуляції та самокорекції, активізації своїх дій, розвиток розуміння особистої відповідальності за свою долю, цілеспрямованого і безперервного самовдосконалення. Учні в 7-ому класі вже ознайомилися з поняттям інформаційної безпеки, виходячи з широкого сенсу, тобто кажучи про формування системних знань в настільки значущою в сучасному інформаційному суспільстві області, як забезпечення ІБ. Це дозволило нам говорити про ІБ як про стан доступності для суспільства достовірній незабороненій інформації, необхідної для його сталого існування і розвитку, при якому забезпечений надійний захист дітей, особистості, суспільства, держави, міжнародного співтовариства від впливу на них загроз, які у формі організованих інформаційних потоків і спрямованих на деформацію їх інформаційного простору, суспільної та індивідуальної свідомості. Наведемо деякі приклади ІБ: Інформаційна безпека на кожному рівні регулюється нормативно-правовими актами, зокрема: Конституцією, законами країни, міжнародними стандартами (ISO/IEC), державними стандартами (ДСТУ, ГОСТ), галузевими стандартами та іншими. Приклади таких актів в Україні: закони «Про інформацію», «Про захист персональних даних», «Про захист інформацій в інформаційно-телекомунікаційних системах» (ІТС), «Про електронні документи та електронний документообіг», інструкція «Порядок підключення до глобальних мереж передачі даних» та багато інших. Слід вказати, що нормативно-правові документи визначають не лише права, але й відповідальність користувачів за порушення у сфері інформаційної безпеки.

Щоб розповідь учителя не була надмірно монотонною можна під час пояснення матеріалу виконати завдання підручника або організувати невелику дискусію за завданнями підручника.

Як правило, інтерес у школярів викликає розповідь про комп'ютерні віруси. Якщо вчитель вважає, що наявних у підручнику ілюстрацій на цю тему недостатньо, то можна підготувати й інші. На початку уроку під час вивчення теми «Комп'ютерні віруси. Антивірусні програми» вчителю варто запропонувати учням відповісти, що вони знають про біологічні віруси, а також з'ясувати, чому біологічні віруси можуть завдавати шкоди живим істотам. Програми-віруси багато в чому подібні до біологічних вірусів, але вони виникають не самі по собі, а створюються користувачем. Вони загрожують нормальній роботі комп'ютера і можуть спричинити різні проблеми. Часто користувач не помічає їх наявності до того моменту, доки не почнуть зникати важливі дані, або сповільнюється робота комп'ютера. Відповіді на

питання (як поширюються комп'ютерні віруси? Як вони діють? Як запобігти зараженню комп'ютера і як позбутися вірусів?) учні можуть надати, виходячи із власного досвіду. Відповіді можна фіксувати на дошці.

Продовжуючи розпочату розмову, вчитель пояснює, що таке комп'ютерні віруси, описує середовище розповсюдження вірусів.

Далі вчитель знайомить учнів з антивірусними програмами, їх класифікацією та приводить приклади використання. Обговорення прикладів використання антивірусних програм зручно проводити на підставі відповідної таблиці, яка представлена у підручнику.

Вчитель демонструє дію антивірусної програми, яка встановлена на комп'ютері, коментує її функції та призначення елементів інтерфейсу. Слід наголосити на важливість оновлення антивірусних баз будь-якої антивірусної програми, адже постійно створюються нові віруси і якщо їх нема у базі програми, тоді комп'ютер стає відкритим для ураження ними.

Прагнення вчителів досягти найкращих результатів у розв'язанні навчальних, виховних і розвивальних завдань спонукає їх до підготовки й проведення нестандартних уроків, структура яких не узгоджується з алгоритмом класичного чи традиційних уроків. Кожен з таких уроків має свою композицію і власний сценарій. На нестандартному уроці учні навчаються, організовуються, розширюють свої пізнавальні можливості, проходить виховання особистості. Такі уроки сприяють розвитку творчих і аналітичних здібностей учнів, стимулюють і підвищують інтерес до навчання, розвивають уміння приймати правильні рішення в різних ситуаціях.

Особливо потрібні нестандартні уроки в нинішніх умовах, коли суспільство потребує освічених молодих людей, здатних до ефективного аналізу реальної політичної, економічної чи виробничої ситуації, обґрунтованого прийняття єдино правильного рішення і його реалізації. Нестандартний урок – це величезне вікно, через яке у духовний світ дитини вливається життєвий потік уявлень і понять про навколишній світ.

Учні люблять уроки у формі гри, тому що процес гри сповнений несподіванок, результат – завжди таємниця. Учні працюють, переживають, найбільш активно є емоційна сфера. Якщо завдання підібрані цікаві, учні можуть забути про все інше й активно долучитися до роботи на уроці.

Як показує практика, вивчення, закріплення і повторення матеріалу на нестандартних уроках сприяє зміцненню умінь і навичок, підвищенню компетентності учнів з предмета, розвитку загальної ерудиції, і, як кінцевий результат, – забезпечує міцність знань. Такі нестандартні уроки вчитель може запропонувати учням під час вивчення даної теми.

2.4. Створення персонального навчального середовища (6 год.)

Метою вивчення теми «Створення персонального навчального середовища» є сформувати поняття про форми та методи отримання знань у мережі та її засобами.

Під час вивчення даної теми учні повинні мати уявлення про синхронізацію даних, технологію використання каналів новин тощо. Учні повинні навчитися створювати, публікувати документи з використанням безкоштовних інтернет-середовищ, створювати та надавати спільний доступ та редагувати документи з використанням офісних веб-програм, підписуватися на канали новин та використовувати спеціалізоване програмне забезпечення для їх перегляду тощо.

Програмою на вивчення розділу «Створення персонального навчального середовища» відведено 6 уроків, з них передбачається виконання двох практичних робіт на комп'ютері.

Зміст розділу здебільшого знайомий учням, але слід наголосити на тому, що навчальним середовищем можна назвати систему, складові якої — інтелектуальна та матеріальна — сприяють досягненню мети навчання, а саме — високої якості освіти. Для підвищення мобільності й доступності персонального навчального середовища можна рекомендувати застосування Інтернет-середовищ, зокрема хмарних обчислень (англ. cloud computing) — хмарних технологій.

2.5. Електронний документообіг (4 год.)

Основні терміни та визначення:

FineReader – одна з систем оптичного розпізнавання текстів, яка дає змогу розпізнавати тексти, набрані практично будь-якими шрифтами, без попереднього навчання.

Автор електронного документа – фізична або юридична особа, яка створила електронний документ.

Адресат – фізична або юридична особа, якій адресовано електронний документ.

Візуальна форма подання електронного документа – відображення даних, які містить документ, електронними засобами або на папері у формі, прийнятній для сприйняття його змісту.

Графічний редактор – комп'ютерна програма, призначена для створення і редагування зображень на екрані комп'ютера та зберігання їх у багатьох популярних форматах, наприклад JPEG, PNG, GIF, TIFF.

Дані – інформація, подана в формі, прийнятній для її оброблення електронними засобами.

Електронний документ – документ, інформація в якому зафіксована у вигляді електронних даних, включаючи його обов'язкові реквізити.

Електронний документообіг (обіг електронних документів) – сукупність процесів створення, оброблення, надсилання, передавання, одержання, зберігання, використання та знищення електронних документів, які виконуються із застосуванням перевірки цілісності та в разі потреби з підтвердженням факту одержання таких документів.

Електронний словник – комп'ютерна база даних, яка містить словникові статті, що дають змогу швидко знайти потрібні слова (словосполучення, фрази).

Електронний цифровий підпис – вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача. Електронний цифровий підпис накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа.

Копія документа на папері для електронного документа – візуальне подання електронного документа на папері, яке засвідчене в порядку, визначеному законодавством.

Обов'язковий реквізит електронного документа – обов'язкові дані в електронному документі, без яких він не може бути підставою для його обліку і не матиме юридичної сили.

Оригінал електронного документа – електронний примірник документа з обов'язковими реквізитами, у тому числі з електронним підписом автора або підписом, прорівняним до власноручного підпису відповідно до Закону України «Про електронний цифровий підпис».

Посередник – фізична або юридична особа, яка у визначеному законодавством порядку здійснює приймання, передавання (доставку), зберігання, перевірку цілісності електронних документів для задоволення власних потреб або надає відповідні послуги за дорученням інших суб'єктів електронного документообігу.

Програми перекладу забезпечують повний цикл перекладу всього документа: введення початкового тексту, переклад іншою мовою, редагування, форматування й збереження перекладеного тексту.

Ресстраційна картка документа – основна одиниця обліку в системі, яка містить дані для формування статистичних і аналітичних звітів по різних інформаційних зрізах та забезпечує легкість під час пошуку.

Система електронного документообігу – інформаційна система, що охоплює діловодство та підготовку документів і поєднує їх із зовнішнім середовищем електронного обміну.

Система оптичного розпізнавання символів – програма автоматичного введення текстових документів у комп'ютер.

Сканер – пристрій, що використовується для аналізу початкового зображення або тексту, його оцифрування і збереження з метою подальшого опрацювання, зберігання у цифровій формі або виведення на друк.

Суб'єкти електронного документообігу – автор, підписувач, адресат і посередник, які набувають передбачених законом або договором прав і обов'язків у процесі електронного документообігу.

Ознайомлення з темою «Електронний документообіг» дозволяє зрозуміти й усвідомити різновиди робіт, які можуть бути автоматизовані в офісі, характеризувати особливості сканування документів різних типів, можливості перекладу різних документів тощо. Слід розпочати з визначення поняття електронний документообіг (обіг електронних документів) як сукупності процесів створення, опрацювання, надсилання, передавання, одержання, зберігання, використання та знищення електронних документів, які виконуються із застосуванням перевірки цілісності та у разі необхідності з підтвердженням факту одержання таких документів. Вказати на важливість його в сучасному суспільстві.

Оскільки процеси створення текстових документів і рисунків розглянуто вже досить докладно, бажано зосередитися на процесі створення електронної копії паперових документів з використанням систем оптичного розпізнавання текстів. За допомогою сканера досить просто отримати зображення сторінки тексту в графічному файлі. Процес сканування дуже простий, хоча його результати можуть істотно розрізнятися і залежать від якості оригіналу (сканованого документа). Сторінка, що містить текст, поміщається на стіл для оригіналів і запускається програма розпізнавання тексту. У цій програмі вказують параметри сканування і тип текстового файлу. Для прикладу OCR-системи доцільно використати програму АBBYY FineReader. Багато користувачів комп'ютерів, які мають сканер, застосовують для сканування і розпізнавання тексту саме таку програму, можливо різних версій. Особливістю програми FineReader є висока точність розпізнавання і низька чутливість до дефектів друку. Ця система оптичного розпізнавання текстів забезпечує дозволяє розпізнавати тексти, набрані практично будь-якими шрифтами, без попереднього її навчання. Для одержання документа у форматі текстового файлу необхідно провести розпізнавання тексту, тобто перетворити елементи графічного зображення в послідовності текстових символів.

Спочатку необхідно розпізнати структуру розміщення тексту на сторінці: виділити колонки, таблиці, зображення тощо. Далі виділені текстові фрагменти графічного зображення сторінки необхідно перетворити на текст. Після завершення розпізнавання отримуємо текстовий файл, який можна відкрити та редагувати в редакторі Word чи в будь-якій іншій програмі.

Ознайомлення з системами *електронного перекладу і словниками* бажано проводити у вигляді бінарних уроків (іноземна мова + інформатика).

До засобів автоматизації перекладу відносяться два типи програм: електронні словники й програми перекладу.

Електронний словник – словник у комп'ютері або іншому електронному пристрої. Дані словники дозволяють швидко знайти потрібне слово, часто з врахуванням морфології і можливістю пошуку словосполучень (прикладів вживання), а так само з можливістю перекладу його іншими мовами. З технічної ж точки зору, електронний словник – комп'ютерна база даних, яка містить словникові статті, що дозволяють здійснювати швидкий пошук потрібних слів (словосполучень, фраз).

Деякі електронні словники забезпечують звуковий супровід перекладених слів. Серед найпопулярніших програм даного класу слід відзначити ABBYY Lingvo (<http://www.lingvo.ua/uk>). ABBYY Online – це колекція онлайн-сервісів, створених на базі технології ABBYY.

Програми перекладу забезпечують повний цикл перекладу всього документа: введення початкового тексту, переклад на іншу мову, редагування, форматування й збереження перекладеного тексту. Прикладом програм перекладу може бути програма машинного перекладу Language Master (LM). Програма може бути інтегрована в комплект програм Microsoft Office, зокрема, у програму Microsoft Word.

При цьому Microsoft Word набуває нових функціональних можливостей перекладу документів, а LM – розширених засобів редагування й форматування перекладених текстів. До переваг програми LM можна віднести її компактність та простий інтерфейс.

Особливості програми LM такі:

- забезпечується переклад документів з англійської, російської і української мов у будь-якому напрямку;
- є конвертор текстових файлів із формату DOS у формат Windows;
- забезпечується перевірка орфографії для всього тексту або виділеного фрагмента будь-якою з трьох мов. Відкоректований текст зберігається в буфері обміну для подальшої заміни;
- є можливість підключення до програми додаткових словників з економіки та інформатики;
- забезпечується динамічне відслідковування напрямку перекладу (мови оригіналу і перекладу);
- програма перекладає окремі слова, виділений фрагмент тексту або весь текст, а також файл або групи файлів у пакетному режимі;
- забезпечується збереження при перекладі в середовищі Microsoft Word параметрів форматування оригіналу й розташування малюнків, графіків, таблиць та інших об'єктів, вигляду оригіналу;
- забезпечується повноцінне використання можливостей Microsoft Word щодо редагування, форматування й перевірки орфографії.

Система машинного перекладу LM складається з двох компонентів:
незалежного програмного модуля;
модуля розширення Microsoft Word.

Результат перекладу зберігається в буфері обміну і відображається в спеціальному вікні.

Перекладаючи фрагмент, слід мати на увазі, що LM:

- перекладає текст як зв'язані речення з урахуванням усіх граматичних особливостей обох мов;
- неперекладені з різних причин слова (помилки в оригіналі, відсутність у словниках) позначаються у вікні перекладу тексту червоним кольором;
- неперекладені з різних причин слова (помилки в оригіналі, відсутність у словниках) можна додати до списку неперекладених слів;

під час перекладу окремого слова у вікні перекладу тексту відображаються виділене слово і всі значення його перекладу, а в буфері обміну зберігається слово оригіналу;

для перекладу файлів або групи файлів їх необхідно заздалегідь виділити у вікні програми перегляду файлової системи.

2.6. Комп'ютерні публікації (8 год.)

Основні терміни та визначення:

Друковані публікації – книги, брошури, журнали, газети, бюлетені, буклети, листівки тощо, виконані поліграфічним способом.

Електронні публікації – документи (книги, брошури, журнали, газети, бюлетені, буклети, листівки тощо), що розповсюджуються з використанням електронних носіїв даних (компакт-дисків, дискет, флеш-носіїв тощо) або комп'ютерних мереж.

Макети – шаблони публікації одного типу (одного змісту, призначення), які відрізняються між собою лише кольоровим чи шрифтовим оформленням об'єктів.

Написи – об'єкти-контейнери, в яких містяться тексти, текстові поля.

Публікація – оприлюднення певних повідомлень (новин, розпоряджень, указів, наукових досягнень, літературних творів тощо).

Тираж, або наклад – виготовлення певної кількості примірників публікації.

Шаблон – порожня форма документа (можливо – з коментарями про те, як заповнювати кожне поле).

Зазначена тема закладає базове розуміння понять комп'ютерна публікація, шаблон та структура публікацій. Однією з найбільш важливих методичних задач викладання цієї теми є навчити учнів описувати особливості роботи з графічними і текстовими об'єктами під час створення публікацій.

Вивчення розділу «Комп'ютерні публікації» бажано організувати з використанням засобу Microsoft Publisher. Разом з тим можна використати й текстові редактори, які забезпечують портування документів у формат *.html. Створений документ (публікацію) бажано розташувати в локальній мережі, з якої він відкриватиметься як Веб-сторінка (без встановлення будь-якого додаткового програмного забезпечення).

Розпочати виклад теми бажано з розповіді, що після створення якогось корисного продукту, завершення виконання проекту, іноді – навіть тільки після виникнення ідеї його започаткування, виникає необхідність розповісти про це людям, переконати їх у необхідності і корисності результатів виконання проекту, порадитись з ними, тобто *презентувати проект або ідею його започаткування*. Таким чином, оприлюднення певних відомостей у формі, доступній не тільки для ініціатора й виконавців проекту, є однією з необхідних складових проектної діяльності. Таке оприлюднення результатів є також діяльністю, яка полягає в перетворенні певних відомостей у форму, яка забезпечує їх сприйняття іншими людьми.

Об'єктом перетворювальної діяльності в цьому випадку є інформація. Форма подання відомостей щодо цілей, процесу виконання й результатів проекту є результатом, продуктом перетворювальної діяльності людини або групи людей (співвиконавців). Необхідність виконання такої перетворювальної діяльності виникає тоді, коли в процесі виконання проектної діяльності група людей дійшла висновку, що існує деякий спосіб отримання суспільно корисного продукту, більшості людей невідомий, і відомості про це потрібно зробити доступними суспільству.

Найстарішим прикладом такого донесення до широкого загалу результатів проектної-пошукової діяльності можуть бути біблійні притчі, одна з

яких, про сім вгодованих і сім худих корів, є не чім іншим, як викладом моделі оптимального управління державою, отриманої в результаті проєктування на основі узагальнення, осмислення досвіду.

На початку вивчення цієї теми можна провести мотиваційну бесіду та навести приклади з повсякденного й ділового життя, коли з'являється потреба зацікавити людей своїми досягненнями, привернути увагу діяльності організації, розповісти в лаконічній формі про продукт, зробити доповідь на конференції та інше. Учителю має пояснити учням, що в таких випадках оприлюднення спеціальним чином підготовлених відомостей, зокрема – результатів проєкту, може відбуватись у формі *презентації* або *публікації*. Учителю слід нагадати учням, що кілька років тому поняття «публікація» в багатьох людей пов'язувалось виключно з паперовою друкарською продукцією. Зараз публікації все частіше розповсюджуються в електронному вигляді як звичайні комп'ютерні файли, які можна переглядати вдома на персональному комп'ютері, в дорозі на екрані ноутбука або мобільному пристрої.

Важливою відмінністю веб-документа від звичайного електронного документа є його гіпертекстова або гіпермедійна структура. Це означає, що електронний документ може не тільки переглядатись послідовно, сторінка за сторінкою. Гіпертекстові або гіпермедійні документи (веб-сторінка, електронна книга тощо) містять програмні складові, які забезпечують цілеспрямовані переходи як у межах документу, так і за його межі – в простір Інтернет, до іншого документа або програмного засобу.

Робота над створенням комп'ютерних публікацій та їх демонстрація дозволяє учням подати власні думки та результати своєї діяльності у наочній формі, вчить знаходити потрібну інформацію, аналізувати наявний матеріал, робити структурний аналіз та оформлювати її в коректному плані.

2.7. Основи алгоритмізації та програмування (56 год.)

Основні терміни та визначення:

Lazarus – візуальне середовище програмування, призначене для розроблення й виконання програм у текстовому та графічному режимах.

Алгоритмізація – процес розроблення алгоритму.

Виконуваний код – програма, готова до виконання на комп'ютері з відповідним процесором і операційною системою.

Клас – сукупність змінних і методів, що описує множину об'єктів.

Компіляція програми – процес перетворення програми на машинний код.

Масив – структурований тип даних, що складається з фіксованої кількості елементів одного типу.

Метод – програма, що описує поведінку об'єкта.

Метод ітерації – багаторазове виконання однотипної обчислювальної процедури з метою наближення отриманого результату до істинного.

Мова програмування – система позначень, за допомогою яких можна точно записати алгоритм для виконання його комп'ютером.

Об'єктно-орієнтоване програмування – програмування, що ґрунтується на поданні програми у вигляді сукупності об'єктів, кожний із яких є екземпляром певного класу.

Оператор циклу з параметром – оператор, що завершує свою роботу, якщо поточне значення параметра перевищує кінцеве.

Оператор циклу з передумовою – оператор циклу, в якому умова продовження циклу перевіряється перед початком кожного виконання операторів тіла циклу.

Оператор циклу з постумовою – оператор, у якому умова завершення циклу перевіряється після кожного виконання операторів тіла циклу.

Подійне програмування – програмування, у якому виконання програми залежить від певних дій користувача, значень зовнішніх або внутрішніх щодо програми величин.

Простий циклічний алгоритм – циклічний алгоритм, що не містить інших циклів.

Рекурентна залежність – залежність, у якій значення змінної на кожній ітерації обчислюється за допомогою значення цієї ж змінної, отриманого під час виконання попередніх ітерацій.

Розмір масиву (довжина) – кількість елементів масиву.

Розмірність масиву – число індексів в масиві.

Структурований тип даних – це безліч елементів даних (компонент) з одним іменем.

Точність обчислення – міра наближення результату, обчисленого комп'ютером, до точного значення.

Форма – вікно середовища Lazarus для створення інтерфейсу програми або елемент інтерфейсу програми.

Розпочати бажано з того, що навчаючись працювати з електронними таблицями (ЕТ), учні вже опрацьовували блоки комірок. Програма мовою Паскаль також може опрацьовувати велику кількість однотипних даних. Для цього використовують таку структуру (тип) даних як масив. При цьому використовується лише одне ім'я змінної, пов'язаної з ними, а звернення до конкретного значення відбувається за індексом (номером) елемента масиву. Дії над елементами масиву здійснюються шляхом використання імені масиву і значення індексу елемента. Масив є іменованою послідовністю елементів (комірок, груп комірок) пам'яті, в яких зберігаються однотипні дані. Для різних масивів типи даних можуть відрізнятися. Наприклад, один масив може складатися з чисел типу integer, а інший – з чисел типу real.

Масиви бувають статичні або динамічні. Кількість елементів у статичних масивах не змінюється під час виконання програми, а в динамічних масивах кількість елементів може змінюватися. Елементи масиву нумеруються за порядком, але необов'язково починаючи з одиниці (хоча мовою програмування Паскаль найчастіше саме з неї). Порядковий номер елемента масиву називається індексом цього елемента. Слід наголосити, що індекс елемента масиву не є його значенням. Значенням є дані, що зберігаються в масиві, а індекси тільки вказують на них.

Реалізація класичних алгоритмів опрацювання й упорядкування масивів досить добре описана у значній кількості джерел. Разом з тим слід наголосити, що упорядкуванням об'єктів називається розміщення цих об'єктів за зростанням або спаданням відповідно до визначеного лінійного відношення порядку, наприклад такого, як відношення “ \leq ” для чисел. При розв'язуванні задачі упорядкування, як правило, вимагається мінімальне використання допоміжної пам'яті, що вказує на недопустимість використання допоміжних масивів. Методи впорядкування можна поділити на: прями та удосконалені. Прямі методи впорядкування за принципом, покладеним в основу методу, своєю чергою поділяються на: порядкування обміном; упорядкування вставлянням; упорядкування вибором.

Прямі методи на практиці використовуються не часто, оскільки мають досить низьку швидкодію. Удосконалені методи впорядкування базуються на таких же принципах, як і прямі, але використовують спеціальні алгоритми для збільшення швидкості процесу впорядкування.

При переході до символічних величин слід обов'язково актуалізувати поняття типу і необхідність типізації, для того, щоб раціонально розподілити пам'ять комп'ютера і уникнути непередбачуваних ситуацій, що виникають тоді, коли над величинами виконуються неприпустимі дії.

Найпростіший двовимірний масив, або ж матриця – це шахова дошка. Хід білого пішака e2-e4 означає, що він перемістився з однієї клітинки в іншу, і вміст клітинки e4 набув значення «білий пішак». Двовимірні масиви можна уявляти також як таблицю MS Excel, кожна клітинка якої теж має дві координати (індекси), наприклад, D5, H7. Необхідно подати алгоритми послідовного обходу масиву (матриці). Складнощі виникають, зазвичай, внаслідок нерозуміння учнями ролі індексів – учні плутають рядок і стовпчик, оскільки вони, на відміну від ET і гри «морський бій» позначаються числовими індексами. З метою пояснення слід використати написання програми поелементного заповнення масиву числами (або літерами) з клавіатури. Для закріплення бажано використати програмування роботи з елементами діагоналей матриць.

Розробка рекурсивних алгоритмів та їх реалізація у вигляді програм

Перш за все слід пояснити необхідність використання допоміжних алгоритмів, тим, що при написанні програм досить часто виникає потреба виконати однакові дії з різними даними в різних місцях програми. Тому ділянки коду, що виконуються більше одного разу, можна записати один раз, забезпечивши доступ до нього довільну кількість разів. Наступні дії залежать від мови й середовища програмування.

Важливим є пояснення ітераційного процесу виконання програми і використання рекурсії (якщо середовище програмування його забезпечує), відмінностей між ними.

Перед учителем поставлено завдання не просто надати випускнику набір знань, а підготувати учня, який вміє самостійно вирішувати виникаючі перед ним проблеми, здатного відповідати за власний добробут і суспільства в цілому. Для цього потрібно створити необхідні умови для підготовки відповідальної, самостійної і активної особистості, яка володіє основними компетентностями. На нараді концептуальної групи проекту «Стандарт загальної освіти» сформульоване таке робоче визначення поняття «компетентність»: «Готовність учня використовувати засвоєні знання, навчальні вміння та навички, а також способи діяльності в житті для рішення практичних і теоретичних задач». Дане визначення можна спростити до такого — вміння складати і здійснювати програму своєї діяльності. І де, як не на уроках інформатики ці вміння формуються найкраще, і, перш за все, при вивченні теми «Алгоритмізація та програмування».

Тема «Алгоритмізація та програмування» одна з найбільш складних тем при вивченні курсу інформатики. Свого часу були намагання вилучити її з навчальних програм, замінивши традиційний математико-алгоритмічний підхід до вивчення інформатики на інформаційно-технологічний. Орієнтувати шкільний курс тільки на практичне освоєння роботи з текстовими редакторами, електронними таблицями, базами даних тощо помилково. За таких обставин інформатика швидко втратила б значення як самостійна навчальна дисципліна.

Виникає питання: для чого потрібно вивчати програмування в школі?

1. З метою фундаменталізації курсу інформатики. Одним з фундаментальних принципів інформатики є принцип програмного керування роботою комп'ютера. Зрозуміти його неможливо, не знаючи, що таке програма для ЕОМ. Таким чином, програмування є елементом загальноосвітнього змісту інформатики.

2. З погляду профорієнтаційної функції предмета. Будь-який шкільний предмет має поряд з освітньою, виховною і розвиваючою функціями виконувати ще і профорієнтаційну. Програмування є галуззю діяльності, дуже важливою, сучасною і престижною.

А що відповідати учневі, у якого відсутня мотивація до вивчення мов програмування, коли дитина каже: «я ж не буду програмістом». Мотивація для навчання може бути кількох типів: або «переваги над іншими, які я можу отримати просто зараз», або «переваги, які я матиму над іншими у самостійному дорослому житті». Є ще «переваги, які я матиму над собою», але це дуже складна мотивація для переважної більшості учнів.

То дійсно, що відповідати? Головне, щоб учителі зрозуміли для себе, для чого саме ми навчаємо програмування всіх. По своїй суті програмування — це планування роботи комп'ютера задля виконання поставленого завдання. Тож, навчаючи програмування, ми навчаємо плануванню діяльності. Комп'ютер у даному контексті виступає як ідеальний виконавець. Навчившись планувати діяльність ідеального виконавця, людина зможе набагато впевненіше навчитися планувати діяльність далеко неідеальних виконавців — себе, інших людей, підлеглих у майбутньому. Не всі хочуть бути програмістами? Проте переважна більшість бажає бути керівниками. От програмування й учити вправному керуванню. Дуже важливо, щоб процес вивчення програмування був цікавим, корисним та зрозумілим для учнів. Саме у цьому полягає педагогічна майстерність учителя інформатики, щоб використавши відповідні методики, правильно обравши мову програмування для вивчення, зацікавити учнів при вивченні теми «Алгоритмізація та програмування».

Шкільні програми не прив'язують навчальний процес до конкретної мови програмування, що дає змогу вчителям вибирати мову, яку вивчатимуть учні, з декількох альтернатив, з урахуванням власної компетентності, інтересів учнів, їх спрямованості та структури навчання інформатики в школі. Визначальним для вибору мови і середовища програмування як засобів навчання є методична підтримка (методичне забезпечення) навчального процесу. У шкільному курсі, навіть за поглибленого вивчення інформатики, не потрібно вивчати всі можливості мови програмування. Це не потрібно навіть для того, щоб перемогти на предметній олімпіаді. Найважливіше навчитися отримати програмні реалізації певного кола алгоритмів. Для цього непотрібно знати багато службових слів чи назв процедур і функцій. Але потрібно навчитися так висловлюватися розмовною мовою, щоб швидко й легко викласти свої висловлювання мовою програмування

Навчання програмування має базуватися на загальних педагогічних принципах: «від простого до складного», «від відомого до невідомого», «від конкретного до абстрактного». Деякі середовища і мови програмування є досить складними для новачків. Навіть написання простої програми в них вимагає певної кількості дій з налаштування середовища. І може статися

так, що середовище програмування, яке забезпечує високу ефективність роботи професіоналів, зовсім не підходить для початкового вивчення в школі.

Учитель не повинен забувати, що мета навчання – не вивчення мови програмування і не підготовка програмістів, а вивчення способів, алгоритмів і методів програмування при вирішенні завдань, тобто прищеплення учням навичок алгоритмічного мислення.

Одним із засобів формування інтелектуальних умінь можна вважати вивчення об'єктно-орієнтованого програмування, методів роботи в об'єктно-орієнтованих системах візуального програмування. У процесі навчання об'єктно-орієнтованого програмування переважно використовуються традиційні форми навчання, такі як лекції, лабораторні роботи, практичні заняття які добре себе зарекомендували в процесі навчання структурному програмуванню. Учителю необхідно звернути особливу увагу учнів не тільки на розгляд переваг об'єктно-орієнтованого програмування, але і навчити дітей реалізовувати ці переваги на практиці.

Технологія візуального програмування дозволяє будувати інтерфейс майбутньої програми зі спеціальних компонентів, що реалізують необхідні якості. Кількість таких компонентів досить велике. Кожен з них містить готовий програмний код і всі необхідні для роботи дані, що позбавляє програміста від створення того, що вже створено раніше. Практично будь яка робота учня з візуальним середовищем програмування пов'язана з такими поняттями як: об'єкт, властивості об'єкта, множина значень властивості об'єкта, методи об'єкта. Програмісти – початківці схильні до великої кількості помилок, і це добре. Саме на помилках вони вчать. Але помилку ще треба побачити, знайти. Тому ще один критерій вибору середовища та мови програмування — це легка у користуванні система налагодження програми, мінімізація вірогідності появи неявних помилок, пов'язаних, наприклад, з використанням не правильних типів даних.

Аналіз сучасних середовищ створення додатків з графічним інтерфейсом показав, що найбільш популярними на сьогоднішній день можна назвати: Lazarus, Delphi, Visual Basic. Середовище візуального програмування Lazarus поєднує в собі об'єктно-орієнтовані засоби візуального програмування і різні технології, що полегшують і прискорюють створення програми

Середовище Lazarus відносяться до категорії вільного програмного забезпечення, що дозволить зняти всі проблеми нелегального використання ліцензійного ПЗ. У той же час Lazarus за своїми можливостями практично не поступається Delphi і є ідеальним засобом для вивчення мови програмування Pascal у школах.

Учителю варто нагадати учням, що розробка прикладної програми в середовищі **Lazarus** складається з декількох етапів:

1. Аналіз та планування виконання завдання.
2. Підготовка проекту
3. Розміщення компонентів інтерфейсу користувача на формі проекту
4. Написання програми опрацювання подій
5. Тестування та налагодження програми.

На першому етапі, крім визначення вхідних даних і вихідних результатів, необхідно побудувати інформаційну модель розв'язку завдання, визначити методи розв'язування та скласти відповідні алгоритми.

На етапі підготовки проекту варто створити окрему папку для нього.

Після запуску середовища програмування **Lazarus** слід створити новий проект — програму з підтримкою графічного інтерфейсу, зберегти його у створену папку, перевірити успішність компіляції та запуску нового порожнього проекту — це можна зробити також, використовуючи клавішу **F9**. Далі — закрити запущений порожній проект і продовжити його створення. Швидко зберігати всі проміжні зміни можна за допомогою комбінації клавіш **Ctrl+S**.

При створенні графічного інтерфейсу програми слід розмістити компоненти цього інтерфейсу на формі та визначити їх властивості. Щоб об'єкти були розміщені на формі на одному рівні по горизонталі чи вертикалі, використовують автовирівнювання, яке показує розміщення одних компонентів по горизонталі та по вертикалі відносно інших лініями вирівнювання.

Програми опрацювання подій створюються в вікні редактора коду автоматично системою або складаються вручну засобами мови програмування **Free Pascal**.

Створений проект запускають на виконання. Якщо код програми не містить синтаксичних помилок і програма успішно відкомпільована, буде відображено екранну форму. Слід перевірити, чи всі об'єкти екранної форми «реагують» на події так, як це передбачено завданням проекту. За наявності невідповідності поставленим завданням необхідно внести зміни до коду програми чи змінити властивості об'єктів.

У навчальній програмі з інформатики 5-9 класів загальноосвітніх навчальних закладів з поглибленим вивченням предметів природничо-математичного циклу визначені такі передбачувані результати навчання:

Учень (учениця):

- розуміє сутність структурованих типів даних;
- знає типові алгоритми опрацювання даних в одновимірному і двовимірному масивах;
- знає основні методи пошуку даних в одновимірному масиві;
- знає основні методи сортування даних в одновимірному масиві;
- пояснює сутність підпрограм, процедур і функцій;
- розробляє і налагоджує найпростіші програми опрацювання даних у масивах;
- розробляє і налагоджує програми пошуку і сортування даних в одновимірному масиві;
- розробляє і налагоджує найпростіші програми з використанням процедур функції.

На початку вивчення теми учні повинні усвідомити той факт, що при обробці великих обсягів даних без їх організації (структурування) не обійтись. На прикладах використання таблиць вчитель повинен показати, наскільки зручно мати величину з одним ім'ям і багатьма значеннями (основна ознака структурування). Укажіть, що таблицею з однорідними даними часто обійтись неможливо, оскільки доводиться працювати з таблицями, в колонках яких різномірні дані (різні кадрові анкети). Нарешті, не завжди число елементів в структурі можна заздалегідь передбачити; більше того, воно може змінюватися «на ходу». Приклади: чисельність групи в процесі навчання, будь-яка черга. Відповідно, чим більше варіантів організації даних пропонує мова програмування, тим вона більш досконала.

Оскільки Lazarus заснований на Object Pascal, тому розглянемо схему (рис.1) типів даних Турбо Паскаля. Зауважимо, що ця схема може допомогти вчителю із визначенням порядку вивчення теми.



Рисунок 1

На схемі не відображено, що в Турбо Паскалі «дійсний» і «цілий» – це групи типів; розмова про них має відбутися окремо. Також треба зауважити, що такі поширені в додатках динамічні структури даних, як *черга* і *стек*, не є стандартними типами Паскаля, але можуть бути реалізовані за допомогою покажчиків.

Вивчення структурованих типів необхідно розпочати з найбільш традиційного – масиву. Масив – це не що інше, як таблиця. Таблиці можуть бути лінійні та прямокутні, а масиви відповідно – *одновимірні* та *двовимірні*.

Під час розв'язування задач, як правило, використовуються одновимірні та двовимірні масиви. Масиви більшої розмірності на практиці майже не зустрічаються.

Порядок роботи з масивом:

- оголосити про масив у розділі описів, вказавши його розмір і тип елементів, що в нього входять (тобто приготувати місце в пам'яті, де будуть зберігатися значення елементів, власне, створення масиву);
- заповнити необхідними значеннями масив для розв'язування задач;
- якщо треба, вивести масив на екран для зорової перевірки роботи з ним;
- робота з даним масивом;
- виведення отриманих результатів.

Розглянемо ці пункти, що визначають послідовність роботи з масивами детальніше.

1. Для створення масиву можна використати наступні способи:

1 спосіб:

Var

<ім'я масиву>: array [нижня межа..верхня межа] of <тип елементів >;

2 спосіб:

Type

<ім'я типу>: array [нижня межа..верхня межа] of <тип елементів>;

Var

<ім'я масиву>: array of <ім'я типу>;

Отже, задати тип масиву можна двома способами: 1) без використання ключового слова `type`; 2) з використанням ключового слова `type`.

Другий спосіб використовується тоді, коли є необхідність багаторазового використання даного типу для опису змінних у ході виконання програми (коли змінні цих типів необхідно передавати до процедур чи функцій, або коли необхідно описати дані такого типу як локальні в кількох різних процедурах чи функціях).

Наприклад: без використання ключового слова `type`

var

a,b: **array** [1..5] **of** real;

c: **array** [1..5,1..5] **of** real;

з використанням ключового слова `type`

type massiv = **array** [1..5] **of** real; {задано одновимірний масив з 5-ти елементів}

matrix = **array** [1..5,1..5] **of** real;

{задано двовимірний масив (матрицю), що складається із 5-ти рядків, кожний з яких містить 5 елементів}

var

a,b: massiv;

Подібна багатоваріантність прикрашає мову, але учні можуть розгубитися. Для того, щоб на початковому етапі вивчення теми не вносити плутанину, методично правильно буде використовувати лише один спосіб.

Задати кількість елементів масиву не було у переліку пунктів роботи з масивами, але вчитель обов'язково це розкаже учням.

1 спосіб: через текстове вікно `Edit`, попередньо виконати перевірку, щоб текстове вікно не було порожнім:

```
If Edit1.Text <> " then begin n:=StrToInt(Edit1.Text);.....
```

```
end
```

```
else ShowMessage («Ввести кількість елементів»);
```

Функція `StrToInt` — переводить рядковий тип у цілий, отже значення `n` матиме числовий тип

2 спосіб: за допомогою діалогового вікна — вікна введення `InputBox`:

```
n:=StrToInt(InputBox («Масив», «Ввести кількість елементів», «3»));
```

Оскільки масив – це множина однотипних даних, тому їх буде більше ніж 3, на що вказує число в лапках.

Звертання до елементу масиву:

Для лінійного (одновимірного) : <Ім'я масиву>[<його індекс>].

Для двовимірного: <Ім'я_масиву> [<індекс рядка>, <індекс стовпчика>]

Надання елементам масиву значень.

Розглянемо три способи виконання даного завдання, а саме:

1. з клавіатури;
2. випадковим чином;
3. за формулою.

Надання значень елементів масиву з клавіатури можна виконати за допомогою циклу з передумовою, але для того, щоб не ускладнювати роботу учнів, вчителю достатньо показати цикл «`for ... do`». Застосування блоку модифікації робить подальшу обробку масиву більш зручною.

```
for i:=1 to n do
```

```

for j:=1 to m do begin
  write (“введіть A[“i”,”j”]”);
  readln (A[i,j]);

```

end;

На рисунку 2 представлена блок-схема алгоритму введення елементів лінійного масиву X з використанням блоку модифікації.

Надання значень елементів масиву випадковим чином відбувається за допомогою лічильника випадкових чисел, що вмикається в події створення форми OnCreate (Randomize).

For i:=Low(<им’я масиву>) **to** High(<им’я масиву>) **do**

```

  <им’я масиву>[i]:=Random(x)+a;

```

(на проміжку [a,b], де $x=b-a+1$ – цілочисельний проміжок).

Наприклад, випадковим чином із проміжку [K,L]:

```

for i:=1 to n do

```

```

  for j:=1 to m do

```

```

    A[i,j]:=random(L-K)+K;

```

При роботі з масивами зручно користуватись операторами циклів. Наприклад, якщо в масиві цілих чисел mas із десяти елементів значення кожного з елементів повинно дорівнювати квадрату індексу цього елемента, то присвоєнті ці значення елементам масиву можна за допомогою наступних команд:

```

var
  mas:array[1
  ..10] of integer;
  i:integer;
  for
  i:=1 to
  10 do
  begin
    mas[i]:=i*i;
  end;

```

Власне, це і є прикладом заповнення масиву за допомогою формули.

2. Виведення елементів масиву на екран для зорової перевірки. Вивід масиву організовується аналогічно введенню, тільки замість блоку введення елемента масиву буде блок виведення.

Методи виведення елементів одновимірного масиву на екран.

1) виведення у стовпчик:

```

for i:=1 to n do
  writeln(M[i]);

```

2) виведення у рядок:

```

for i:=1 to n do
  write(M[i]:5);

```

При виведенні елементів масиву у рядок бажано зазначити формат виведення, наприклад, `write(M[i]:10:3)` – для дійсних чисел або `write(M[i]:5)` – для цілих.

Виведення двовимірного масиву на екран.

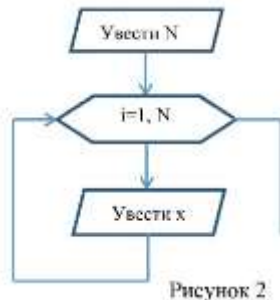


Рисунок 2

```

for i:=1 to n do
begin
  for j:=1 to m do
    write(A[i,j]:8); {виведення в рядок}
    writeln; {перехід на новий рядок}
  end;
end;

```

Учитель знову має нагадати дітям, що для роботи з масивом потрібен оператор повторення. Вочевидь, що у двовимірному масиві необхідно використовувати їх два: один цикл, внутрішній, потрібен для переходу між елементами рядка (тобто по стовпчиках), а другий, зовнішній, – для переміщення між рядками. Якщо в матриці кількість рядків і стовпчиків однакова, то таку матрицю називають **квадратною** (на відміну від звичайної прямокутної таблиці). Тільки в квадратних матрицях існують головна та бічна діагоналі. Учитель обов'язково звертає увагу учнів на те, що в Паскалі, а відповідно і в Lazarus масив не може бути динамічною структурою. Учні часто потрапляють у глухий кут при необхідності, скажімо, видалити елемент з масиву. Видалення елемента зі зрушенням інших ліворуч – операція технічно нескладна, але тоді виходить «дірка» на правому кінці масиву. Прибрати її «на ходу» неможливо за самою природою масиву, і якщо це дійсно необхідно, то треба користуватися іншою структурою.



Знайомити учнів з алгоритмом видалення елемента з масиву вчитель може розпочати з наступного простого завдання. Необхідно видалити третій елемент з масиву X , що складається з 6 елементів. Алгоритм видалення третього елемента полягає в тому, що на місце третього елемента необхідно записати четвертий, на місце четвертого – п'ятий, а на місце п'ятого – шостий.

```
X[3]:=X[4]; X[4]:=X[5]; X[5]:=X[6];
```

Таким чином, всі елементи з третього по п'ятий треба перемістити ліворуч на один, на місце i -го елемента потрібно записати $(i + 1)$ -й.

Блок-схема алгоритму представлена на рисунку 3. Доречно показати учням виконання оберненої дії, тобто – вставку елемента у масив. Вчитель може запропонувати дітям нескладне завдання: вставити число b в масив X (10) між третім і четвертим елементами.

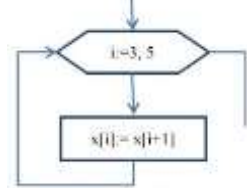


Рисунок 3

Для вирішення цього завдання необхідно всі елементи, починаючи з четвертого, зрушити вправо на один елемент. А потім у четвертий елемент масиву записати b ($X[4] := b$).

Але щоб не втратити сусіднє значення, змішувати треба спочатку десятий на один вправо, потім дев'ятий, восьмий і так до четвертого.

Блок-схема алгоритму вставки «нового четвертого» елемента приведена на рисунку 4, а на рисунку 5 праворуч узагальнена блок-схема вставки числа b у масив $X(N)$ між елементами з номерами m і $m + 1$.

Четвертий пункт із списку, що визначав порядок роботи з масивами, це власне $i \in$ **робота із самим масивом**.

Навички використання масивів закріплюються за допомогою вирішення типових задач. До них відносяться: організація поелементного введення і виведення лінійного або двовимірного масиву (простий цикл), підрахунок числа додатних або з іншою властивістю елементів числового масиву, знаходження найбільшого елемента масиву (цикл з вкладеним розгалуженням) тощо. Потім переходять до завдань складніших, які вимагають організації структур типу «цикл в циклі», і більш складним: упорядкувати лінійний числовий масив за зростанням або за спаданням, знайти найбільший елемент в двовимірному масиві тощо.

Вчителеві не варто обмежуватися завданнями, в яких масиви мають числову природу. Прикладом завдання на використання символічного масиву може слугувати така задача: «на схемі кінозалу зірочками позначені місця, на які квитки на сеанс продані, рисками – вільні, підрахувати число проданих квитків.» При роботі з масивами найчастіше доводиться виконувати

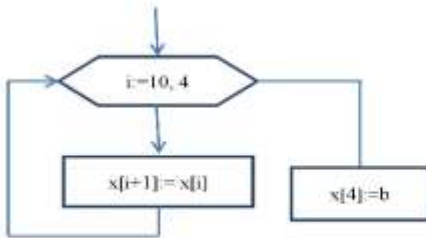


Рисунок 4

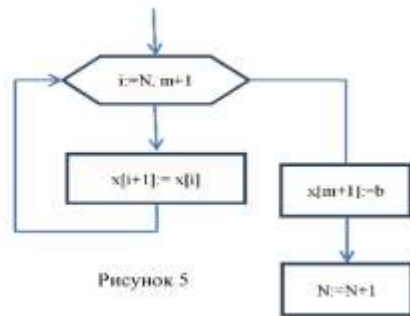


Рисунок 5

операції сортування та пошуку даних. Такі задачі мають велике практичне значення, тому від ефективності реалізації цих операцій часто залежить ефективність всієї програми.

Сортування – це процес перегрупування даних у деякому заданому порядку. Основна мета сортування – полегшити пошук потрібної інформації у заданій послідовності даних.

Методи сортування масивів можна розбити на три категорії:

1. Сортування за допомогою **включення** (метод *прямої вставки* – послідовно вибирається елемент з масиву і після визначення його місця у впорядкованому наборі даних вставляється безпосередньо на своє місце).

2. Сортування за допомогою **вибору** (метод *прямого вибору* – в масиві вибирається елемент з певними властивостями, наприклад, мінімум або максимум, а потім вибраний елемент становиться на своє місце).

3. Сортування за допомогою **обміну** (обмінні – виконується обмін між собою двох деяких (найчастіше сусідніх) елементів масивів, якщо відповідні елементи розташовані у вихідному масиві невірно; процес повторюється або певну кількість разів, або доки при черговому проході елементи в масиві не будуть переставлятися).

Вчитель може акцентувати увагу учнів на тому, що це розподіл сортування за принципом дії (by insertion; by selection; by exchange). Більшість алгоритмів сортування базуються на тому факті, що треба переставляти два елементи таким чином, щоб після перестановки вони були правильно розташовані один відносно одного.

Вчитель може запропонувати поетапно розглянути сортування методом включення на такому масиві.

12	-8	0	30	5			100
----	----	---	----	---	--	--	-----

Розбиваємо його на дві частини. До першої входить єдиний впорядкований елемент $\{12\}$, а до другої – всі останні $\{-8\ 0\ 30\ 5\ 100\}$. Напишемо тепер процес впорядкування по етапах:

I етап: елемент, що впорядковується, -8.

$-8 < 12$, тому виконується обмін, тобто після першого кроку масив має наступний вигляд:

-8	12	0	30	5		100
----	----	---	----	---	--	-----

На цьому цикл припиняє свою роботу, тому що досягнуто початок масиву ($i=1$).

II етап: елемент, що впорядковується = 0.

1) $0 < 12$, отже виконується обмін, тобто після першого кроку масив має вигляд:

-8	0	12	30	5		100
----	---	----	----	---	--	-----

2) $0 > -8$, тому обмін не виконується, здійснюється вихід із циклу, масив залишається без змін.

III етап: елемент, що впорядковується = 30.

1) $30 > 12$, вхід до циклу не відбувається, масив залишається без змін.

IV етап: елемент, що впорядковується = 5.

1) $5 < 30$, виконується обмін, після перестановки масив має вигляд:

-8	0	12	5	30	100
----	---	----	---	----	-----

2) $5 < 12$, здійснюється обмін, масив набуває вигляду:

-8	0	5	12	30	100
----	---	---	----	----	-----

3) $5 > 0$, цикл припиняє свою роботу, масив залишається без змін.

V етап: елемент, що впорядковується = 100.

1) $100 > 30$, вхід до циклу не відбувається, тому що умова одразу хибна і масив залишається без змін.

Після ознайомлення і розгляду прикладу сортування методом включення вчителю варто застосувати методичний прийом під назвою «Мозковий штурм». Доречним буде запитання про недоліки методу вставки. Передбачувана відповідь: для великих масивів такий ланцюжок порівнянь, де кожний наступний елемент алгоритм порівнює його з кожним більшим за значенням від нього елементом у відсортованій частині, може бути достатньо громіздким, що передбачає значне збільшення часу виконання програми. Коли учні дійдуть такого висновку, вчитель повідомляє про те, що для прискорення процесу зазвичай використовують так званий метод половинного поділу. Тобто на кожному етапі елемент, який необхідно вставити, порівнюється з елементом, що міститься в середині відсортованої частини. Якщо його значення більше від даного елемента, то наступним аналізується елемент із середнім індексом в меншій частині, якщо менше — то в більшій і так далі, поки не залишиться один елемент. Після цього, всі елементи значення яких являються більшими від значення елемента, який необхідно вставити, зсуваються в циклі, і на це місце робиться вставка.

Принцип сортування за допомогою **прямого вибору** такий, як і в методі вставки: масив поділяється на відсортовану та невідсортовану частини. Основна ідея методу полягає в тому, що на кожному кроці шукається елемент,

найменший з тих, що залишилися невідсортованими, а після цього він додається до впорядкованої частини масиву останнім (бо він там є найбільшим). Оскільки блок-схема чітко показує систематичну послідовність всіх етапів виконання поставленого завдання, а також всі групи, які залучені в цей процес, то пропонуємо вчителю не відмовлятися від графічного відображення методу сортування за допомогою прямого вибору, який представлено на рисунку 7.

На першому кроці серед усіх елементів масиву знаходимо мінімальний і міняємо його місцями з першим елементом. Після цього серед групи, що починається з другого елемента, знову ж таки знаходимо мінімальний елемент і міняємо його місцями вже з другим елементом. Продовжуючи даний процес далі, на останньому етапі визначаємо мінімальний елемент серед двох останніх (за номером індексу), після чого, знайдений елемент міняється місцями з передостаннім елементом вихідного масиву.

```

uses crt; const =6;
var
  i,j,k,z: integer;
  x: integer; a:array[1..n]of
integer;
begin ClrScr; randomize;
for i:= 1 to n do begin
  a[i]:=random(256); write ('
',a[i]);
  end;
  writeln;
  writeln;
  for i:=1 to n-1 do begin
    k:=i;
    x:=a[i];
    for j:= i+1 to n do
      begin
        if a[j] < x then
          begin
            k:=j;
            x:=a[k];
          end;
        end;
    a[k]:=a[i];
    a[i]:=x;
    for z:=1 to
      n do write('
',a[z]);
    writeln;
    writeln;
    for i:=1 to n do write('
',a[i]);
    readkey;
  end.

```

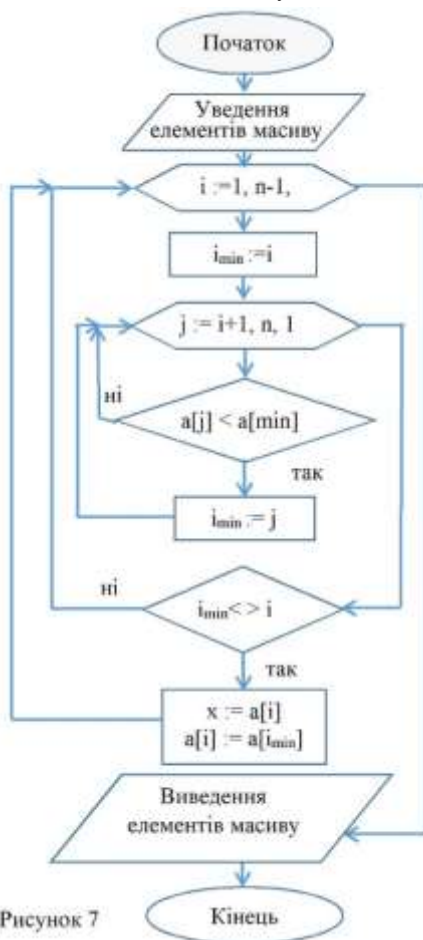


Рисунок 7

Далі запишемо програму сортування даного масиву. Для демонстрації на екран виводяться вихідний масив, масив після кожного етапу вибору і обміну та кінцевий масив.

Сортування за допомогою **обміну** базується на процесі порівняння і при необхідності обміну місцями двох сусідніх елементів масиву. Найбільш відомим обмінним сортуванням є метод *«бульбашки»*. Будемо сортувати одновимірний масив за зростанням.

Порівняємо перший елемент масиву з другим, якщо перший виявиться більше другого, то поміняємо їх місцями. Потім порівняємо другий з третім, якщо другий більше третього, то також поміняємо їх, далі порівнюємо третій і четвертий, і якщо третій більшого четвертого, міняємо їх місцями. Так чинимо, поки не порівняємо $(n-1)$ -ий і n -ий елементи. Учителю обов'язково треба зазначити, що після першого проходу по всьому масиву максимальний елемент переміщується в крайнє праве положення.

Номер елемента \ Вигляд масиву	1	2	3	4	5	6
Вихідний масив	45	0	15	30	5	-5
Масив після першого перегляду	0	15	30	5	-5	45
Масив після другого перегляду	0	15	5	-5	30	45
Масив після третього перегляду	0	5	-5	15	30	45
Масив після четвертого перегляду	0	-5	5	15	30	45
Масив після п'ятого перегляду	-5	0	5	15	30	45

Рисунок 8

І тут варто задати питання: чи є сенс на наступному етапі перевіряти весь масив? Учні повинні дійти висновку, що на практиці при першому проході перевіряють елементи з номерами від 1 до n (останнього), на другому від 1 до $(n-1)$ і т.д. Можна запропонувати дітям переглянути наступну таблицю (рис. 8), яка ілюструє впорядкування масиву методом *«бульбашки»*.

Зразок програми:

```
uses crt;
const n=6;
var i,j,k,d:integer;
tmp:integer;
a : array [1..n] of integer; begin
ClrScr; randomize;
for i:= 1 to n do begin
a[i]:=random(256);
write (' ',a[i]);
end;
writeln;
writeln;
d:=n-1;
```

```

for j:=1 to n-1 do begin
for i:=1 to d do
if ( a[i]>a[i+1]) then begin
    tmp:=a[i]; a[i]:=a[i+1]; a[i+1]:=tmp;
end;
d:=d-1;
for k:=1 to n do
write(' ',a[k]);
    writeln;
end;
writeln;
for i:=1 to n do
write(' ',a[i]);
readkey;
end.

```

На рисунку 9 наведена блок-схема даного методу, а поруч – текст програми, призначеної для упорядкування масиву по зростанню методом «бульбашки».

Ще одна операція, яка виконується з масивом – це **пошук даних** (знаходження елемента з заданими властивостями). Задане значення критерію пошуку називається ключем пошуку. Це може бути умова рівності елемента заданій величині, найменший елемент масиву тощо. Для організації пошуку в таблиці елементів із заданими властивостями необхідно організувати циклічний перегляд всіх елементів, кожний з яких командою розгалуження порівняти із заданим критерієм або перевірити на деяку властивість. Якщо масив одновимірний, цикл для організації перегляду всіх елементів буде один, якщо ж масив двовимірний – циклів буде два. Найпростішим, але не завжди оптимальним методом пошуку є прямий лінійний пошук. Цей метод використовується тоді, коли немає ніякої додаткової інформації щодо групи елементів, серед яких провадиться пошук. Метод полягає в послідовному перегляді всіх елементів і перевірці їх на відповідність ключу пошуку. Умовою закінчення пошуку може бути або факт знаходження даного елемента, або той факт, що дану сукупність елементів перевірено повністю і не знайдено елементів, що відповідають критерію пошуку.

Розглянемо задачу пошуку максимального елемента (Max) і його номера (Nmax) в масиві X, що складається з n елементів.

Алгоритм розв'язання цієї задачі наступний. Припустимо, що перший елемент масиву є максимальним, і запишемо його в змінну Max, а в Nmax – його номер (число 1). Потім всі елементи, починаючи з другого, порівнюємо в циклі з максимальним. Якщо поточний елемент масиву виявляється більшим максимальний, то запишемо його в змінну Max, а в змінну Nmax – точне значення індексу i.

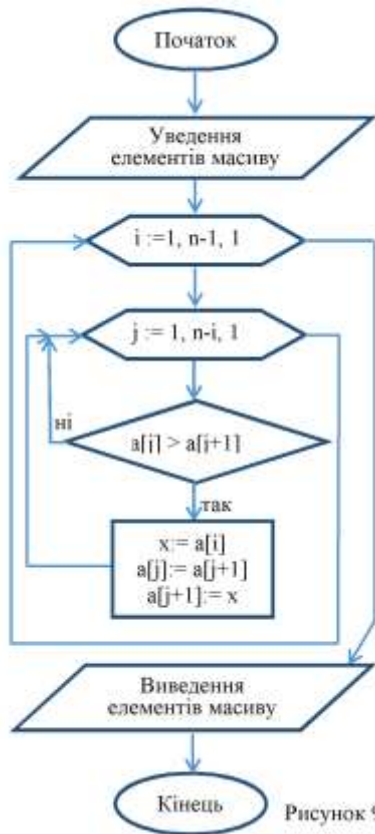


Рисунок 9

Процес визначення максимального елемента в масиві відображений у фрагменті програми:

```
Max:=X[1]; Nmax:=1;  
for i:=2 to N do if X[i]>Max then begin Max:=X[i]; Nmax:=i; end;  
write('Max=',Max:1:3,  
max=',Nmax);
```

Приклад план-конспекту урока

Тема: «Одновимірні масиви. Опис масиву та дії над елементами масиву»

Мета:

Навчальна – сформулювати уявлення про масиви як про засіб зберігання інформації; визначити ефективність використання масивів при обробці великої кількості даних; навчитися описувати одновимірний масив, будувати алгоритм роботи з масивами.

Розвиваюча – розвиток пізнавального інтересу та творчої активності учнів; розвивати у школярів вміння висловлювати власні думки, моделювати ситуацію, планувати свою діяльність.

Виховна – елементи патріотичного виховання, вміння співпрацювати, толерантність, ініціативність, вміння досягати поставленої мети.

Задачі уроку:

Навчальна – дати визначення масиву, як структурованого типу даних, розповісти про способи опису і заповнення масиву, про дії над елементами масиву.

Розвиваюча – розвиток алгоритмічного мислення, пам'яті, уважності.

Виховна – розвиток пізнавального інтересу, логічного мислення.

Тип уроку: урок засвоєння нових знань.

Педагогічні технології: технологія співробітництва, проблемно-пошуковий метод, інформаційно-комп'ютерні технології.

Обладнання: комп'ютер, проектор, звичайна дошка, інтерактивна дошка, презентація «Одномірні масиви», створена в MS PowerPoint.

Хід уроку.

I. Організаційний момент. Привітання учнів. Перевірка відсутніх на уроці. Підготовка учнів до уроку.

II. Створення проблемної ситуації. Постановка задачі.

Вчитель. Сьогодні урок розпочинаємо з розв'язування задачі.

Задача: Ввести з клавіатури n довільних чисел, деякі з них можуть повторюватися. Порахуйте кількість чисел, що дорівнюють найменшому числу масиву.

Складемо алгоритм розв'язку задачі.

Передбачувані відповіді учнів.

1. Ввести n (кількість чисел, що вводяться).

2. Знайти найменше число.

- Ввести число і запам'ятати його в змінну min .
- У циклі **поки** не переглянемо n чисел роботи:
- Ввести число і запам'ятати його в змінну x
- Порівнювати x з min . Якщо $x < \text{min}$, то записуємо в змінну min значення x .

3. Порівняти введені раніше числа зі значенням min .

Вчитель. А як виконати порівняння, якщо ми не запам'ятовували числа, які вводили?

III. Повідомлення теми уроку. Формулювання цілей уроку.

Вчитель. Як вирішити цю проблему? Або записувати всі числа, що вводяться на аркуш (при невеликих значеннях n), або потрібно десь зберігати ці числа.

Отже діти ми підійшли до того, що для вирішення завдання вам потрібні нові знання. Запишіть тему нашого уроку: «Одновимірні масиви. Опис масиву та дії над елементами масиву».

Ви знаєте тему уроку. Пропоную сформулювати цілі уроку (вчитель допомагає учням і доповнює їх відповіді).

IV. Вивчення нового матеріалу.

З поняттям масиву ви знайомі з географії і не раз чули вислови: гірський масив, житловий масив. Першим житловим масивом Києва є Першотравневий масив, розташований вздовж Чоколівського бульвару. Будівництво масиву розпочато 1956 року на місці городів та полів. У цей масив входять такі вулиці:

1. Євеванська, Адама Міцкевича (один із найвидатніших польських поетів). 2. Пітерська. 3. Іскрівська. 4. Уманська. 5. Юлівса Фучика (чеський письменник і журналіст, страчений фашистами в берлінській в'язниці в 1943 році). 6. Петра Ніщинського (псевдонім *Петро Байда* — український композитор і поет-перекладач), 7. Курська.

Цю інформацію можна розмістити так:

1.	2.	3.	4.	5.	6.	7.	8.
Євеванська	Міцкевича	Пітерська	Іскрівська	Уманська	Фучика	Ніщинського	Курська

Маємо, по-перше лінійну таблицю, а по-друге — це однотипні занумеровані дані, що мають спільне ім'я («вулиці», можна це ім'я уточнити). Кожному номеру в таблиці відповідає певне значення. Складаючи програми, ми використовували прості типи даних, наприклад, `var a: real; b: integer;` Тут можна поставити запитання: «Які ще прості типи даних вам відомі?»

Але щоб скласти програму, наприклад, в якій потрібно обчислити середній бал успішності учнів за рік(семестр) потрібно ввести дуже багато змінних для обробки та запам'ятовування великого набору даних. У такому випадку для опису змінних простий тип даних не підходить, а використовують так званий структурований тип даних, як таблиця, а інакше — двовимірний масив. Це дає нам можливість зекономити електронний простір, а також зробити програму компактною.

Масив — це однотипні занумеровані дані, що мають спільне ім'я. Приклад одновимірного масиву — це таблиця температур за тиждень (за місяць). Приклад двовимірного масиву — це таблиця результатів спортивних змагань кількох команд. (Наведіть свої власні приклади). Одновимірний масив інакше ще називають лінійною таблицею або вектором.

Масив відноситься до структурованих типів даних (упорядкована сукупність даних). Номери елементів масиву інакше називаються індексами, а самі елементи масиву — змінними з індексами (індексованими змінними).

При зверненні до елемента масиву потрібно вказати ім'я масиву та індекс того елемента з яким ви хочете виконати дію. Індекс елемента масиву вказується в квадратних дужках після імені масиву.

Наприклад, масив `a` заповнено таким чином:

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
-5	8	3	-2	0	4	3	-2	1	3

Ім'я масиву, індекс, значення.

Зверніть увагу, що дані в масивах зберігаються лише до кінця роботи програми. Для їх довготривалого зберігання програма повинна записати дані в файл.

Характеристики масиву:

- **тип** — всі елементи масиву одного типу;

- **розмірність** — кількість індексів елементів (одновимірний, двовимірний, ..., багатовимірний);
- **розмір** (довжина або діапазон зміни індексу) — загальна кількість елементів у масиві.

Одновимірний масив — це приклад масиву, в якому елементи нумеруються одним індексом.

Способи задання (опису) масиву.

Найпростіший спосіб опису масиву — це оголосити змінну в розділі опису змінних var з використанням зарезервованого слова array (масив). Опис має такий вигляд:

Var

<ім'я масиву>: array [нижня межа; верхня межа] of <тип елементів масиву>;

Наприклад:

Const n=100;

var a: array[1..n] of real; (100 елементів — дійсні числа)

b: array[0..50] of char; (51 елемент — символи)

c: array[-3..4] of boolean; (8 елементів — логічні значення)

x, y: array[1..20] of integer; (два масиви x і y містять по 20 елементів — цілі числа)

Опис масиву потрібно компілятору для виділення пам'яті під його елементи.

Для опису масивів часто використовується попередній опис типу в розділі опису типів даних. Такий опис може вимагатися, наприклад, при використанні імені масиву в якості параметра процедури або функції.

Загальний вигляд такий:

Type

<ім'я типу>: array [нижня межа; верхня межа] of <тип елементів масиву >;

Var

<ім'я масиву>: <ім'я типу елементів масиву >;

Наприклад:

Type z: array [1..20] of integer;

Var x, y: z

Мозковий штурм. (Слайд 9).

Питання до мозкового штурму:

1. Чи обов'язково при виконанні програми заповнювати всі комірки масиву даними? Чому?

2. Якщо комірка не заповнена, то яке значення в ній знаходиться?

3. Чи може реальна кількість елементів в масиві бути меншою, ніж зазначено при описі? Чому?

4. А чи може бути більшою? Чому

Заслуховуються відповіді учнів. Обговорюємо і робимо висновки.

Висновок. При виконанні програми зовсім не обов'язково заповнювати всі комірки даними (в цьому випадку значення комірки дорівнюватиме нулю), оскільки кількість елементів, яким надано значення, в масиві може бути меншою, ніж зазначено при його описі, але більше елементів в масиві бути не може.

Дії над елементами масиву:

1. Знаходження суми елементів масиву.

Дано масив X, що складається з n елементів. Знайти суму елементів цього масиву. Змінній S присвоюється значення нуль, потім послідовно до змінної S додаються елементи масиву X.

2. Знаходження добутку елементів масиву.

Дано масив X, що складається з n елементів. Знайти добуток елементів цього масиву. Розв'язання зводиться до того, що значення змінної P, в яку попередньо була записана одиниця, послідовно множиться на значення i-го елемента масиву.

Способи заповнення одновимірного масиву(лінійної таблиці)

Варіант 1. Заповнення масиву за формулою (може бути іншою) for i:=1 to n do for j:=1 to m do a[i]:=5+2*i;
Варіант 2. Заповнення масиву з клавіатури for i:=1 to n do for j:=1 to m do begin write('vvesti a[',i,',',j,']='); readln(a[I,j]);end;
Варіант 3. Заповнення масиву випадковими числами for i:=1 to n do For j:=1 to m do a[I,j]:=random(100)+1;
Варіант 4. Сталими числами const a:array[1..3,1..4] of integer=(3,5,7,9,4,11,13,1,5,2,1,7);

Може бути запропонована й задача знаходження добутку елементів масиву, що мають непарний індекс.

3. Пошук максимального елемента масиву та його номеру.

V. Робота за комп'ютером.

Залача. (Та, за допомогою якої створюємо проблемну ситуацію).

Умова: Дано одновимірний масив цілих чисел $A[i]$, де $i = 1.2. \dots n$. Визначити, скільки разів мінімальний елемент зустрічається у даному масиві та порядковий номер першого найменшого елемента.

Алгоритм розв'язання.

1. Описати масив. 2. Заповнити масив. 3. Знайти мінімальний елемент масиву. 4. Підрахувати кількість елементів, що дорівнюють мінімальному.

Для знаходження мінімального елемента масиву спочатку необхідно пройти по всіх елементах масиву і знайти серед них максимальний, запам'ятавши його номер. Для цього користуються стандартним алгоритмом:

1) береться перший елемент масиву і його значення присвоюється змінній *min*, тобто він вважається за зразком найменшого елемента;

2) по черзі з масиву вибираються всі останні елементи і, якщо серед них знайдеться менший за вибраний зразок, то змінній *min* присвоюється нове значення, яке тепер буде новим зразком. У іншій змінній, наприклад, N_{min} запам'ятовується номер цього найменшого елемента (початкове значення цієї змінної було 1, тому що спочатку ми вважали найменшим 1-ий елемент).

Після закінчення перегляду всього масиву змінна *min* буде містити шуканий мінімум, а змінна N_{min} — його номер. Щоб запам'ятати номер першого мінімального елемента, необхідно шукати в матриці елемент, що точно менший зразка. Якщо ж ми будемо шукати елемент, що не більший за зразок, то в змінній N_{min} залишиться номер останнього найменшого елемента (подумайте чому).

Після знаходження мінімуму другим проходом можна вже підрахувати кількість таких елементів в масиві. Для цього кожен елемент порівнюється із зразком, що знаходиться в змінній *min*, та до лічильника *count* додається одиниця в випадку повторення (спів падання) цих значень.

Програма, що реалізує описаний алгоритм.

Program Example_321_1_2;

Uses crt;

Const n = 30;

Var A:array[1..n] of integer; {A - масив даних чисел}
i:byte; {i - змінна циклу}

count, N_{min} :byte; {count - кількість мінімальних елементів в масиві, N_{max} — номер першого найменшого елемента}
 min :integer; { min – мінімальний елемент масиву}

Begin

Clrscr;

Randomize; {Заповнення масиву випадковими числами та виведення його на екран для контролю за роботою програми}

For i:=1 to n do

Begin

A[i]:=random(150) - random(80);

Write(A[i]:5);

end: {Надання змінним початкових значень}

$min:=A[1]$;

$N_{min}:=1$;

count:=0; {Прохід по масиву для пошуку мінімуму та його номеру}

for i:=1 to n do

begin

if A[i]> min then begin $min :=A[i]$; $N_{min}:=i$; end;

end;

{Другий прохід по масиву для підрахунку кількості мінімальних елементів}

for i:=1 to n do

begin

if A[i]= min then count:=count+1;

end;

Writeln('Мінімум = ', min);

Writeln('Номер першого мінімуму = ', N_{min});

Writeln('Кількість мінімумів = ',count);

Readkey; {Затримка зображення на екрані}

End.

VI. Підсумок уроку.

Рефлексія.

1. Переваги масивів? (Індекси елементів масиву забезпечують доступ не до одного, а до послідовності елементів. Обробка масивів проводиться при зміні індексів елементів. Відразу можна зберігати і обробляти велику кількість однотипних даних.)

2. Як задається опис масиву?

3. Яким чином задається звернення до елемента масиву?

4. Чому при описі масиву краще вживати константи, а не вказувати розміри масиву в явному вигляді?

5. Чи досягли ми мети уроку?

2.8. Бази даних (16 год.)

Основні терміни та визначення:

Атрибути – спеціальні об'єкти, за допомогою яких можна задати правила опису властивостей.

База даних – сукупність взаємозв'язаних даних, використовуваних багатьма користувачами.

Екземпляр об'єкта – одиничний об'єкт.

Інкапсуляція – виокремлення властивостей і методів, властивих об'єкту як його атрибути.

Клас – множина об'єктів реального світу (або їх моделей), які мають однакові властивості й залежність їх від зовнішніх впливів.

Об'єкт – модель реального об'єкта, в якій узагальнено властивості, спільні для деякої множини об'єктів реального світу, або абстрактних об'єктів.

Поліморфізм – здатність об’єкта належати більш ніж до одного класу або започатковувати новий клас.

Успадковування – правила утворення об’єкта дочірнього класу з можливістю додавання або ревізирования властивостей і методів.

Розділ «**Бази даних**» (БД) було вирішено відновити у програмі масової школи, що вмотивовано не лише великим поширенням використанням систем керування базами даних (СКБД) у сучасному суспільстві, а й тим, що інформатика в старшій школі стає обов’язково-вибірковим предметом і деякі учні взагалі не зможуть ознайомитися з цим об’єктом вивчення. Невеликий час, який можна надати для вивчення БД і СКБД, визначає особливості методики навчання. Суттєві ускладнення виникають внаслідок того, що для появлення ефективності СКБД і засвоєння основних прийомів роботи з ними бажано застосовувати досить великі набори даних, а створення учнями БД (у т.ч. багатотабличних, що дуже бажано) і наповнення їх достатньою кількістю даних за наданий час майже нереальне. Тому слід зорієнтуватися на вже готові, наповнені даними БД. Математичний апарат, на якому засновано теорію СКБД, не розглядається, але деякі специфічні терміни (відношення, об’єкт, клас, запис) треба ввести й пояснити.

Бажано максимально можливо вилучити з навчального процесу рутинні операції – введення великої кількості тексту, даних, а зосередити роботу на видах діяльності, які забезпечують можливості для компетентнісного підходу. З цією метою слід використати вже готові БД з наповненням, доступним для розуміння учнями, зі структурою, яку можна відобразити не більше, ніж п’ятьма – сімома блоками. Для того, щоб можна було продемонструвати роботу СКБД, потрібно, щоб у таблицях була достатня кількість записів (не менше десяти), а структура БД забезпечувала ілюстративним матеріалом зв’язування таблиць, пошук за умовою тощо.

Безумовно, потрібно ввести поняття типів БД, відношення, запису. Тут потрібно використати горизонтальні зв’язки між розділами – типізація змінних у мові програмування, типізація вмісту комірки у ЕТ, поняття об’єкта. Уводячи поняття запису як кортежу даних, можна провести паралель з масивом, наголошуючи на відмінностях.

Дуже важливо й провести паралель між рядком комірок у ЕТ і записом у реляційній БД, наголошуючи на строгості вимог однотипності даних у полі запису і нестрогості цієї вимоги до комірок стовпчика.

Поняття «відношення» у реляційних базах даних (РБД) розглядається як відображення зв’язків між об’єктами та їх властивостями в реальному світі, їх модель. Для формалізації дій над об’єктами РБД розроблено спеціальний математичний апарат, який ми не розглядатимемо, але деякі поняття й терміни використовуватимемо.

Модель однієї й тієї ж предметної області можна спроектувати різними способами. Наприклад, можна створити одну таблицю з великою кількістю стовпців, або навпаки, розподілити властивості об’єктів у більшій кількості невеликих таблиць. Бажано проілюструвати цю властивість. Приклад може бути таким: нехай автомобіль, що належить приватній особі, характеризуватимемо значеннями таких властивостей: виробник, модель, рік випуску, державний номерний знак, власник (прізвище, ім’я, по батькові), тоді відношення напишемо спочатку так: «**Приватний автомобіль**(виробник, модель, рік випуску, державний номерний знак, власник(прізвище, ім’я, по батькові))».

Уважно проаналізувавши зроблений запис, побачимо, що приватного власника автомобіля не завжди можна ідентифікувати однозначно, адже

може бути кілька людей з однаковими прізвищем, ім'ям, по батькові. Однозначно громадянина України (як і будь-якої розвинутої держави) можна ідентифікувати, вказавши його ідентифікаційний код або серію і номер паспорта. Отже, є сенс відношення **Приватні автомобілі** описати таким чином: «**Автомобіль** (виробник, модель, рік випуску, державний номерний знак, ідентифікаційний код власника)», додавши до нього відношення «**Власник** (ідентифікаційний код, прізвище, ім'я, по батькові)». Замінивши одне відношення двома (які слід подати не менше ніж двома таблицями), ми створили умови для точного описання екземпляра об'єкта і отримання можливості однозначно ідентифікувати кожний екземпляр об'єктів відношення. Після пояснення треба зазначити, що операція, яку ми щойно виконали, має назву «декомпозиція відношення» і використовується майже завжди при створенні РБД.

Теорія баз даних – важливий розділ сучасної інформатики, їх широке використання в різних галузях людської діяльності робить актуальною підготовку користувачів баз даних.

Тому мета вивчення баз даних як сукупності засобів для зберігання структурованої інформації, полягає в узагальненні й систематизації цих уявлень, формуванні відповідних теоретичних знань, з'ясуванні загальних принципів опрацювання структурованої інформації й оволодінні навичками опрацювання баз даних за допомогою конкретної системи управління базами даних. Водночас розділ предметної галузі, що використовується для навчання і дані з якої зберігаються в базі даних, повинен бути відомим і зрозумілим учням, і його вивчення не повинне вимагати залучення додаткових теоретичних відомостей з інших предметних галузей. Але разом з тим вивчення відповідного матеріалу дозволяє здійснювати міжпредметні зв'язки на уроках інформатики, актуалізуючи знання учнів з географії, біології, хімії, історії тощо.

Можна виявити такі способи діяльності користувача під час розв'язування типових інформаційних задач, інваріантних відносно різноманіття типів інформаційних систем і засобів подання даних, які полягають в послідовному застосуванні методів:

- аналізу інформаційних потреб користувачів і явищ предметної галузі, які моделюються в базі даних;
- синтезу процедур опрацювання даних у режимах пошуку, оновлення, захисту, перетворення даних;
- машинної реалізації одержаних процедур на комп'ютері;
 - інтерпретації одержаних результатів.

Аналізуючи різні аспекти діяльності користувача при роботі з базою даних, легко встановити, що всі вони пов'язані з розв'язуванням інформаційних задач трьох основних типів:

- одержування інформації на основі даних, які вже зберігаються в базі;
 - створення нової бази даних і підтримка одержаної моделі предметної галузі в певному стані;
 - оновлення раніше створеної бази даних (тобто додавання нових і вилучення застарілих даних).

Саме такі задачі доцільно обирати за навчальні і використовувати в навчальному процесі. Поряд з традиційними задачами щодо опрацювання інформації: пошук, додавання, вилучення, зміна даних – учні можуть розв'язувати задачі подання даних у табличній формі, з елементами статистичного опрацювання. Це дозволить застосовувати, з одного боку, знання із курсу математики, географії, економіки та інших предметів під час розв'язування конкретних прикладних задач, з іншого, – одержати досить чіткі уявлення

про переваги використання комп'ютерів і телекомунікаційних мереж для опрацювання великих масивів інформації.

Вивчення баз даних слід почати з обґрунтування актуальності даного додатку комп'ютерної техніки. Задачу можна сформулювати так: є великий обсяг даних про будь-яку реальну систему об'єктів або подій, наприклад, про книжки в бібліотеці, про працівників підприємства, про товари на складі, про дорожньо-транспортні події за тривалий період часу і т. п. Необхідно організувати зберігання цієї інформації так, щоб її було зручно переглядати, поповнювати, змінювати, шукати потрібні відомості, робити будь-які вибірки, здійснювати сортування у будь-якому порядку. Такою роботою людям доводилося займатися і задовго до появи комп'ютерів. Основним засобом зберігання даних був папір. Дані зберігалися у вигляді списків у товстих журналах, теках, на картонних картках. Більшості учнів вони добре знайомі: на кожній картці записані відомості про окрему книжку. В алфавітному каталозі картки систематизували за прізвищами авторів в алфавітному порядку, у предметному каталозі — з тематики книжок. Подібні систематизовані картотеки використовуються у відділах кадрів підприємств. Вони зручні тим, що легко можна витягнути потрібну картку, замінити її, додати нові картки, зберігаючи встановлений порядок. Проте, якщо така картотека містить тисячі карток, то, як би не була удосконалена її організація, обробка даних у ній — справа тривала і трудомістка.

Інший приклад — архіви різних документів. Наприклад, існують історичні архіви, архіви судових справ, архіви патентів на винаходи і багато інших. Деколи такі архіви займають цілі будівлі. Пошук у них потрібних документів вимагає значних зусиль. Крім того, існують кіноархіви, фотоархіви, архіви звукових записів.

Комп'ютерні інформаційні системи дозволяють зберігати великі обсяги даних, здійснювати в них швидкий пошук, вносити зміни, виконувати всілякі маніпуляції з даними (групувати, сортувати та ін.). Слід навести приклади таких інформаційних систем. Це, наприклад, система продажу залізничних і авіаційних квитків. Інший знайомий учням приклад: під час теле-репортажів з великих міжнародних змагань, олімпійських ігор на екран миттєво виводиться досьє будь-якого спортсмена, про якого говорить коментатор — це працює комп'ютерна інформаційна система.

Основою інформаційної системи є *база даних* — організована сукупність даних на магнітних дисках. Під час вивчення теми треба пригадати з учнями таке питання — класифікація баз даних. Бази даних класифікуються за різними ознаками. За характером інформації БД діляться на *фактографічні* і *документальні*. Якщо проводити аналогію з описаними вище прикладами інформаційних сховищ, то фактографічні БД — це картотеки, а документальні — це архіви. У фактографічних БД зберігається: коротка інформація в чітко певному форматі, У документальних БД — всілякі документи. Причому це можуть бути не тільки текстові документи, але і графіка, відео і звук (мультимедія).

Класифікація за *способом зберігання даних* ділить БД на *централізовані* і *розподілені*. Уся інформація в централізованій БД зберігається на одному комп'ютері. Це може бути автономний ПК або сервер мережі, до якого мають доступ користувачі-клієнти. Розподілені БД використовуються в локальних і глобальних комп'ютерних мережах. У такому разі різні частини бази зберігаються на різних комп'ютерах. Третя ознака класифікації баз даних — за *структурою організації даних*. Бази даних називаються відповідно *реляційними* (табличні БД), *ієрархічними* і *мережевими БД*. Теоретично дове-

дено, що будь-яка система даних може бути відображена за допомогою таблиць. Найпростіша реляційна БД містить одну таблицю, складніша може складатися з безлічі взаємопов'язаних таблиць.

У термінології реляційних баз даних *рядки таблиці називаються записами, стовпці — полями*. Сама назва «реляційна БД» походить від англійського слова «relation», що перекладається як «відношення». Тут термін «відношення» розуміється як взаємозв'язок між полями таблиці. У реляційному підході таблиця називається відношенням. Пояснення даного матеріалу слід проводити на конкретних прикладах. Слід підкреслити, що в базах даних кожна таблиця повинна мати своє ім'я. Учителеві зручно працювати, якщо ці таблиці винести на плакати.

Учителю важливо розуміти, що тема «Бази даних» містить в собі низку вузлових питань, що мають фундаментальне значення для курсу інформатики в цілому. У цій темі учні зустрічаються з поняттям величини. Зазначене поняття широко використовуватиметься в електронних таблицях, в алгоритмах і програмах. *Величина — це окремий інформаційний об'єкт, який має власне ім'я та займає місце в пам'яті комп'ютера*.

Необхідно звернути увагу учнів на таку обставину: символічне поле може складатися з цифр. З цієї причини іноді виникає плутанина з символічним і числовим типами. Щоб цього не відбувалося, потрібно дотримуватися такого правила: *якщо поле позначає порядковий номер або цифровий код, то йому потрібно призначити символічний тип*. Якщо ж поле позначає кількість чогось або якусь розмірну величину, то це поле повинне мати *числовий тип*. Наприклад, полями символічного типу повинні бути: інвентарний номер книжки в бібліотеці, номер будинку, номер телефону. Числовий тип мають, наприклад, такі поля: вік людини, маса об'єкта, відстань, кількість учнів у класі.

У даному розділі учні зустрічаються з логічним типом даних, з логічними величинами. Перше поняття про логічну величину можна дати так: це відповідь на альтернативне запитання. Наприклад: Чи «є дана книжка в бібліотеці?» або чи «Поступив абітурієнт в університет?», або «На вулиці йде дощ?» і т. п. Відповідями на такі запитання можуть бути тільки "так" або "ні". Синонімами є "істина" і "хибність". Якщо поле таблиці прийматиме тільки такі значення, то йому призначається логічний тип.

Необхідно закріпити поняття "головний ключ", "ім'я поля", "тип поля" на серії завдань.

Завдання першого типу: дано ім'я таблиці і перелік нулів, вимагається вказати головний ключ і визначити типи всіх полів.

Завдання другого типу: визначена предметна галузь бази даних; вимагається дати назву таблиці, визначити імена полів і їхні типи, призначити головний ключ. Звичайно, ця задача має відношення до галузі проектування БД і вже з цієї причини складна.

Після ознайомлення з основними поняттями, що відносяться до організації інформації в реляційних БД, слід перейти до вивчення програмного забезпечення, призначеного для роботи з базами даних. Таке програмне забезпечення називається СУБД — система управління базами даних.

Для персональних комп'ютерів існує ціла низка СУБД реляційного типу. Історично однією з перших таких систем була dBASE, використовувани в цій системі формати представлення даних, мова обробки даних стали стандартом для низки наступних СУБД. До складу пакета Microsoft Office входить реляційна СУБД Access, яка все частіше використовується в школі.

Є два способи роботи користувача з базою даних: робота за допомогою прикладних програм, наперед складених програмістом у середовищі СУБД, і робота шляхом безпосередньої взаємодії із СУБД.

Під час вивчення даної теми вчителю варто дати опис середовища якої-небудь конкретної СУБД. На відміну від інтерфейсу текстових або графічних редакторів, інтерфейси різних СУБД менш уніфіковані, тому тут важко давати узагальнений опис. Адаптацію до конкретної системи повинен виконати *вчитель*.

Вивчення конкретної СУБД слід проводити за стандартною методичною схемою: «середовище – режими роботи – система команд – дані». Наприклад, для СУБД Access розрізняються такі основні режими роботи: режим роботи з таблицею: "Таблиця"; режим роботи із запитами "Запити"; режим роботи із звітами: "Звіт"; режим роботи з формами "Форма".

Під час ознайомлення учнів із кожним з об'єктів СУБД (таблиця, форма, звіт, запит, макрос, модуль) доцільно дотримуватися такої методики.

1. За допомогою невеликих конкретних завдань і запитань на прикладі вже створеної бази даних продемонструвати основні засоби для роботи з об'єктом, пояснити вказівки, які передбачені для роботи з ним.

2. Навчити учнів працювати з кожним з об'єктів у двох режимах: у режимі роботи з конкретним об'єктом (режим таблиць, режим форм, режим звітів тощо) і в режимі конструктора відповідних об'єктів; з'ясувати різницю двох режимів і призначення режиму конструктора; сформувати вміння учнів виконувати основні вказівки в кожному з режимів, вільно переходити від одного режиму до іншого; сформувати уявлення про те, що будь-який об'єкт за допомогою системи управління базою даних можна змінити тільки в режимі конструктора, а працювати (виконувати певну систему вказівок під час опрацювання даних) з об'єктом – у відповідному режимі об'єкта.

3. Продемонструвати різні способи створення кожного з об'єктів. Вказати різницю між такими способами і сформулювати правила використання кожного окремого способу створення кожного з об'єктів.

4. Під час ознайомлення з кожним з об'єктів використовувати індуктивний метод на частково-пошуковій основі й метод демонстраційних прикладів з підказками різного типу – від заповнених діалогових вікон до карток-підказок, точних і узагальнених алгоритмів виконання відповідних дій.

У СУБД MS Access передбачено використання таких об'єктів: файли, таблиці, форми, звіти, запити, модулі, макроси. Ознайомлювати учнів з їх призначенням доцільно в такому порядку: файли – таблиці – запити – форми – звіти. Макроси й модулі можна вивчати у разі поглибленого вивчення інформатики (через брак часу за навчальною програмою).

Вивчення середовища баз даних можна поділити на дві частини: спочатку слід навчити учнів працювати з готовою базою даних для розуміння основних понять, властивостей об'єктів і операцій над ними. Далі можна приступати до вивчення можливостей використання конкретної системи управління базами даних (СУБД) та створення власних баз даних. Після вивчення теми можна запропонувати творчу або проектну роботу на проектування баз даних, що є достатньо складним завданням, але разом з тим і творчим, й саме тому цікавим для більшості учнів.

Демонструючи можливості використання, наприклад, СУБД MS Access, доцільно звернути увагу на такі питання:

1. Можливість виконання різними способами основних операцій:
 - пошук у базі даних записів, що задовольняють задану умову;
 - поновлення в базі даних значень деяких полів;
 - створення звіту за результатами проведених операцій;

- додавання до бази даних одного чи кількох записів;
- вилучення з бази даних одного чи кількох записів.

2. Можливість працювати з різними предметними галузями, які найчастіше використовують переваги СУБД. Наприклад, розклад потягів і літаків, телефонна книжка мешканців міста, адресна книжка платників податків у районі, бібліотечний каталог видань, інформація про наявність товарів у магазині, інформація про співробітників підприємства, інформація про стан здоров'я хворих у лікарні тощо.

3. Можливість опрацювання в різних предметних галузях різних за типом даних: текстові, числові, графічні, звукові тощо.

Пристаюючи до вивчення баз даних, необхідно чітко усвідомити, що база даних — це основна складова сучасної інформаційної системи (ІС). На наш погляд, доцільно скористатися таким визначенням ІС.

Інформаційна система — це система, у якій можуть накопичуватися величезні обсяги даних, і яка забезпечує автоматичний пошук, обробку і видачу необхідної інформації.

Слід особливо зауважити, що інформаційні системи нині застосовуються в різноманітних сферах людської діяльності, наприклад, бухгалтерський облік і аудит, бібліотечна справа, банківські і фінансові установи, системи продажу й резервування залізничних квитків, податкова служба, системи управління послугами і товарами. Структура і функції ІС значною мірою визначаються тією предметною галуззю, у якій вона застосовується.

Головна функція інформаційної системи, незалежно від предметної галузі, залишається незмінною і полягає в підвищенні ефективності інтелектуальної праці людини. Вона звільняється від виконання рутинних операцій. Забезпечується висока швидкість доступу до даних і отримання їх із джерел, розташованих на великих відстанях.

У процесі розгляду питання про класифікацію інформаційних систем слід звернути увагу учнів на те, що нині існує значна кількість класифікаційних ознак, однак основними є призначення і спосіб доступу до даних. Слід також враховувати, що системи обробки транзакцій сприймаються учнями важко, тому поясненню їх сутності слід приділити більше уваги. Необхідно також навести конкретні приклади застосування систем обробки транзакцій у банківських установах, системах резервування і продажу авіаквитків тощо. Вивчення поняття бази даних бажано розпочати з ознайомлення поняття моделі організації даних. Модель даних визначає, у який спосіб здійснюється об'єднання даних у структури різної складності. На нашу думку, доцільно дотримуватися такого визначення терміну бази даних.

База даних — це організована сукупність даних певної предметної галузі, які відображають стан об'єкта і зв'язки між ними. Предметною галуззю може бути, наприклад, школа, поліклініка, фірма, будівельний супермаркет тощо.

У процесі розгляду реляційної моделі даних першочергову увагу слід приділити таким поняттям. Рядок таблиці (або запис) називають кортежем відношення, а ім'я стовпця (поля) — атрибутом. Усю сукупність імен атрибутів відношення називають його схемою. Зазвичай, схема зображується так: вказується великими літерами ім'я таблиці, а в дужках — перелік імен стовпців. Кількість атрибутів відношення (стовпців) називається його степенем, а кількість кортежів — кардинальністю або потужністю. Сукупність всіх значень стовпців називають його доменом.

Потрібно роз'яснити, що таблиця може бути віднесена до реляційної моделі бази даних, якщо вона відповідає таким вимогам:

- таблиці не можуть міститися однакові рядки;

- усі стовпці таблиці повинні мати різні назви;
- у кожному стовпці можуть міститися тільки однакові типи даних;
- у кожній клітинці таблиці повинні міститися тільки прості дані (не можна в одній клітинці вказувати, наприклад, дату народження і місце народження).

Таблиця повинна містити стовпець або декілька стовпців, які однозначно ідентифікують будь-який рядок. Такий стовпець (або сукупність стовпців) називають **ключем** відношення. Якщо ключ містить один стовпець, його називають простим, інакше — складеним. Таблиця може містити декілька ключів, але використовується лише один з них, який називають **первинним**.

Ключовим поняттям будь-якої бази даних є зв'язок між таблицями, що здійснюється за допомогою ключових полів. Процес зв'язування таблиць бажано розглянути на прикладі. Можна скористатися прикладами, наведеними у підручнику. Учні повинні розуміти, що зв'язок між таблицями встановлюється шляхом додавання ключового поля однієї таблиці до другої таблиці, у якій це поле вже не є ключовим.

Таблиця, ключ якої додається до другої, називається **головною**, а таблиця, у яку додається нове поле — **підлеглою**. Ключові поля головної таблиці, додані до підлеглої, називають **зовнішнім ключем** підлеглої таблиці.

Необхідно також пояснити, що інколи зовнішній ключ водночас може бути і первинним ключем тієї самої підлеглої таблиці. Крім того, імена полів зовнішнього ключа не обов'язково мають бути однаковими з іменами відповідних полів первинного ключа. Однак значення первинного і зовнішнього ключів повинні бути однаковими.

Усі наведені вище поняття й терміни обов'язково розглядаються на прикладі конкретної бази даних. Учителю бажано розробити власні приклади на основі тих, що наведені у підручнику.

Питання нормалізації баз даних є для учнів одними з найскладніших. Тому вивчення його слід починати з пояснення того, що одну й ту ж базу даних можна створити з різною кількістю таблиць, наприклад, з двох, трьох або більше. Особливо багато нюансів виникає в процесі визначення вмісту полів і їх розподілу по таблицях. Наприклад, адреса народження в одному полі з назвою АДРЕСА може містити такі дані: місто, вулицю, будинок, номер квартири. Ці самі дані можна розмістити в чотирьох окремих полях з назвами: МІСТО, ВУЛИЦЯ, БУДИНОК, НОМЕР КВАРТИРИ.

Нормалізацію рекомендується виконувати після внесення до бази даних усіх елементів даних і створення попередньої її структури. Правила нормалізації потрібно виконувати послідовно. На кожному етапі перевіряється відповідність бази даних попередній нормальній формі. Для більшості баз даних достатньо перевірити перші три форми. Кожна «старша» нормальна форма вважається кращою попередніх, а при переході до наступної нормальної форми властивості попередніх нормальних форм зберігаються.

На початку розгляду сутності індексування таблиць можна нагадати, що сучасні бази даних можуть мати сотні пов'язаних таблиць, кожна з яких містить мільйони записів. У цих таблицях часто доводиться знаходити рядки за значенням ключового поля або іншого, а також за значеннями сукупності кількох полів. Пошук таких рядків можна виконати різними методами, наприклад, методом прямого перебору всіх рядків. Однак цей метод потребує багато часу на його реалізацію. Значно швидшим є бінарний метод пошуку, але він вимагає впорядкування рядків за значенням певного поля. Процес упорядкування даних у таблиці є тривалим. Він вимагає багато машинного

часу, а збереження додаткових (упорядкованих) таблиць пов'язане зі значними витратами пам'яті. Для розв'язання цієї проблеми використовуються **індексні таблиці** або просто **індекси**, процес створення яких називається **індексуванням**. Індекс найчастіше містить один стовпець, у якому записується вказівник на певний рядок таблиці з даними. Порядок розміщення вказівників в індексній таблиці визначається значенням певного поля таблиці даних. У процесі індексування не здійснюється упорядкування реально існуючих таблиць і не створюються нові таблиці даних, а створюються індекси, у яких вказівники визначають, у якому порядку повинні розміщуватися рядки основної таблиці за значенням певного її поля.

Після вивчення сутності індексування можна перейти до розгляду функцій і класифікації систем керування базами даних. Але слід враховувати, що саме на цьому етапі вони сприймаються учнями досить абстрактно. Тому їх необхідно розглядати лише з точки зору користувача і навести найголовніші з них.

На вивчення локальних інформаційних систем не доцільно використовувати багато часу. Важливо усвідомити, що функціонування локальної ІС може підтримуватися за одним з трьох варіантів:

- на ПК встановлюється повнофункціональна СКБД;
- на ПК встановлюється ядро СКБД і застосування користувача;
- на ПК встановлюється тільки застосування користувача у вигляді виконуваного файлу (файлу типу exe).

У процесі розгляду інформаційних систем необхідно враховувати, що нині поширеною є архітектура типу клієнт-сервер, тому їй необхідно приділити першочергову увагу. Учні повинні розуміти, що в інформаційній системі цього типу функції СКБД поділені на дві частини. На комп'ютері сервері вона має назву **сервер бази даних**, а на комп'ютері-клієнті — **клієнт**. Причому розподіл функцій між цими частинами програм у різних ІС може значно відрізнятись.

У загальному випадку клієнт — це програма, яка виконує функції зв'язку користувача з інформаційною системою. Вона перетворює команди користувача в команди запитів, які надходять до серверної частини. На сервері ці запити обробляються, на основі чого відбираються необхідні дані й передаються клієнту. Отримані дані клієнт перетворює в інформацію, доступну і зрозумілу користувачеві. Клієнтом може бути спеціально розроблене застосування для виконання конкретного завдання. Крім того, клієнтом можуть бути вже існуючі програми, які підтримують інтерфейс із сервером, наприклад, текстовий процесор, табличний процесор, система керування базами даних. Типовим прикладом такої СКБД є MS Access.

Сервер бази даних — це потужна програма, яка виконує функції керування і захисту даних у ЦБД. У більшості випадків запити до цієї програми надходять мовою SQL, тому, зазвичай, її називають SQL-сервером. Як SQL-сервер використовуються, наприклад, такі потужні системи, як MS SQL Server, Novell NetWare SQL та інші.

Отже, сервер бази даних передає на комп'ютер-клієнт не файли, а тільки необхідні для нього дані, що значно зменшує навантаження на комп'ютерну мережу. Такі дані формуються сервером БД на основі запитів клієнтів. Клієнт обробляє отримані дані й видає користувачеві інформацію у зручному для нього вигляді.

Що стосується розподілених і паралельних баз даних, то слід нагадати, що у перших комп'ютерних мережах використовувалися централізовані бази даних, тобто бази даних, які зберігалися як єдине сховище даних на одному потужному комп'ютері. Усі користувачі цієї мережі мали доступ до

єдиного сховища даних. У принципі на комп'ютерах-клієнтах могли зберігатися і власні (індивідуальні, автономні) бази даних, але вони не були пов'язані з централізованою базою даних.

Розподілені бази даних зберігаються частинами на комп'ютерах мережі (вузлах мережі), які логічно взаємопов'язані між собою. Є різні способи розподілу даних. Один із них передбачає наявність частини БД на кожному комп'ютері мережі. Отже, на комп'ютерах зберігаються окремі частини БД, але ці частини мають між собою зв'язки і в сукупності вони є логічно єдиною базою даних. Для керування такими БД використовуються розподілені системи керування базами даних (РСКБД).

Паралельні бази даних використовуються в багатопроцесорних системах. У процесі паралельної обробки даних одна задача поділяється на кілька менших, які одночасно виконуються на кількох вузлах комп'ютерної мережі, що прискорює час її виконання.

Розподіл БД у системах паралельної обробки даних залежить від архітектури багатопроцесорних систем, яка визначає спосіб взаємодії між процесором, оперативною пам'яттю і дисковим простором. Однак, незалежно від архітектури багатопроцесорних систем дані в них зберігаються на кількох магнітних дисках. Розподіл даних за багатьма дисками здійснюється для того, щоб кілька процесорів отримали одночасний доступ до даних, розміщених на магнітних дисках. Водночас на різних дисках можуть розміщуватися як окремі таблиці БД, так і окремі їх кортежі.

Вивчати основні можливості СКБД Access 2007 необхідно у формі практичного заняття. Головним елементом інтерфейсу користувача слід розглядати стрічку, на якій розміщено всі інструментарії для роботи з обраним об'єктом. Доступ до них бажано здійснювати через пункт меню. Будь-яка робота з базою даних розпочинається з головного вікна.

Необхідно звернути увагу на те, що в нижній частині вікна кнопки **Office** є кнопка **Параметри Access**. Після натиснення на цю кнопку відкривається вікно, у якому розміщені команди для налаштування основних параметрів Access. Але на початковому етапі бажано не користуватися цими командами. Необхідні параметри встановлені заздалегідь і діють за замовчуванням. Після повного ознайомлення із системою пізніше можна повернутися до цього питання. Якщо відкривалося вказане вікно, потрібно закрити його шляхом натиснення на кнопку **Скасувати**. Закриття вікна здійснюється шляхом натиснення кнопки миші поза межами цього вікна.

Завершити розгляд основних можливостей системи Access 2007 необхідно створенням порожньої бази даних шляхом самостійного розроблення структури БД і її об'єктів. У процесі її створення слід розглянути вкладки системи: **Основне**, **Створити**, **Зовнішні дані**, **Знаряддя бази даних**.

Рекомендації щодо вивчення поняття таблиці. Таблицю слід розглядати як головний об'єкт будь-якої бази даних, у якій власне й зберігаються дані. У процесі створення й модифікації структури таблиць необхідно навчити учнів виконувати такі основні дії:

- розробляти структури таблиць;
- визначати властивості полів;
- уводити в комп'ютер структури таблиць;
- установлювати зв'язки між таблицями;
- здійснювати модифікацію таблиць.

Усі операції над таблицями слід розглядати на прикладі двох нескладних таблиць. Для зручності в роботі доцільно властивості полів оформити у вигляді окремих таблиць. Порядком введення структури таблиць можна розглядати на прикладі однієї таблиці. Рекомендується створювати структуру

таблиці в режимі конструктора таблиць. Під час збереження структури таблиці на даному етапі можна не визначати ключове поле. Тому після отримання попередження **Первинний ключ не визначено** слід натиснути кнопку **Ні**. У результаті таблиця буде збережена і в області переходів з'явиться її ім'я. Радимо після цього закрити таблицю.

Учні повинні усвідомити, що модифікувати структуру таблиці можна як безпосередньо після її введення, так і після наповнення даними. Усі операції з модифікації таблиць учні виконують під керівництвом учителя. Порядок встановлення простого ключа бажано розглянути на прикладі раніше створеної таблиці. Звернути увагу учнів на існуючі варіанти переходу в режим Конструктора, а також на позначення ключового поля в таблиці.

Під час вивчення порядку індексування роз'яснити, що система Access підтримує прості індекси (по одному полю) і складні індекси (по декількох полях). Індексувати можна всі поля, крім полів типу **МЕМО**, **OLE** і **Гіперпосилання**. Індексування ключового поля система здійснює автоматично. Порядок створення індексів розглядати на прикладі конкретного поля, дотримуючись правил, наведених у підручнику.

Розглядаючи питання зв'язування таблиць, учні повинні чітко усвідомити, що одночасно працювати з даними різних таблиць можна тільки в тому випадку, якщо ці таблиці мають між собою зв'язки. Зв'язок між таблицями встановлюється за допомогою ключового поля головної й зовнішнього поля підлеглої таблиці. Система Access 2007 автоматично встановлює зв'язки між таблицями за умови правильного визначення ключового поля в головній таблиці, а в підлеглої таблиці є його ім'я з однаковим типом даних.

Роботу з даними в таблицях доцільно розглядати в такій послідовності:

- введення даних;
- перегляд даних (навігація по таблиці);
- пошук і заміна даних;
- сортування даних;
- фільтрація даних.

Уведення даних у таблиці доцільно починати з безпосереднього введення даних за допомогою клавіатури в комірки відкритої таблиці. Розглядати цей спосіб можна на прикладі введення даних у таблицю ПРАЦІВНИКИ бази даних *vik_1*. Потрібно звернути увагу на те, що у випадку неправильного встановлення ключових полів, неправильного індексування або спроби увести недозволений тип даних, їх введення буде заблоковане. Перегляд даних у таблиці можна починати з використання панелі навігації. Навігація по таблиці можна здійснити і звичайними засобами, тобто за допомогою клавіш переміщення курсору або за допомогою миші.

У процесі вивчення порядку сортування даних особливу увагу слід звернути на те, що сортувати рядки таблиці можна по будь-яких полях, крім полів типу **МЕМО**, **OLE** і **Гіперссилка**. Сортувати можна за різними ознаками, наприклад, за алфавітом, за датою, у порядку зростання або зменшення чисел, причому сортувати можна як по одному, так і по декількох полях. За замовчуванням під час відкриття таблиці рядки впорядковуються за значенням ключового поля.

Сутність і порядок виконання фільтрації доцільно розглянути на конкретних прикладах. Наприклад, якщо виконати фільтрацію в таблиці БУДИНКИ за значенням 1980 поля *Рік будови*, то будуть відібрані й виведені на екран перший і шостий рядки. Усі три способи фільтрації Access 2007 необхідно також розглядати на прикладах.

Операції з таблицями учні сприймають легко. Можна дати можливість учням оволодіти цими операціями самостійно, а потім один з учнів повторює порядок їх виконання для всього класу.

Вивчення **запитів** бажано розпочинати з роз'яснення, що запит — це об'єкт Access 2007, головне призначення якого — отримання з однієї або кількох таблиць бази необхідних даних у зручній формі. Він забезпечує зручну форму подання даних, а головне — запит можна зберігати, тобто користуватися ним багаторазово. За допомогою запиту можна створити нову таблицю, модифікувати записи в таблицях, змінювати й видаляти дані, виконувати обчислювальні операції з даними. Можна навести, наприклад, такі приклади: з таблиці ПРАЦІВНИКИ можна отримати дані про кількість працівників, стаж роботи яких більше 15 років. З таблиць ПРАЦІВНИКИ й БУДИНКИ можна отримати дані про працівників, які обслуговують будинки з кількістю квартир більше 90.

Доцільно пояснити, що залежно від функцій, які виконують запити, розрізняють запити на вибірку (вибирають дані з таблиць), запити на зміну (модифікують структуру й дані в таблицях), запити до сервера (здійснюють вибір даних із сервера) та інші. Найчастіше використовуються перші два види запитів, особливо запит на вибірку. Тому, надалі бажано розглядати тільки запити на вибірку і запити на зміну.

Основним інструментом створення запитів необхідно обрати **Конструктор запиту**. Спочатку слід розглянути запити на вибірку. Запити за умовою доцільно розглянути на такому прикладі таблиці БУДИНКИ: нехай із цієї таблиці необхідно отримати дані про будинки, які побудовані після 1960 року, причому тільки по трьох полях: *Шифр, Рік будови, Квартир*.

Переименування запитів можна вивчати за такою методикою. Для надання запиту нового імені необхідно відкрити базу даних, у якій міститься даний запит (для нашого прикладу база *vik_1*), натиснути праву кнопку миші в області переходів на імені *zanim1*, у контекстному меню, що відкрилося, вибрати пункт **Переименувати**. У результаті ім'я *zanim1* буде виділено прямокутником іншого кольору. Потім уведуться в поле цього імені нове ім'я, наприклад ім'я *zanim2*, і натискається клавіша **Enter**. Для редагування запит викликається на екран у режимі таблиці, для чого необхідно двічі натиснути клавішу миші на його імені. Після цього можна перейти в режим **Конструктора** (для цього натискається кнопка під командою **Вигляд**) і в меню, що відкриється, виконується команда **Конструктор**. У результаті на екрані буде висвітлено бланк запиту з ім'ям *zanim2*. Після цього можна перейти до розгляду основних можливостей з редагування запитів і сортування записів.

Важливу увагу слід приділити запитам з полями, що обчислюються. Система Access 2007 дозволяє створювати запити з полями, значення яких обчислюються зі значень інших полів. Наприклад, можна створити для таблиці ПРАЦІВНИКИ запит із полем *Кількість років*, значення якого обчислюється на основі значення поля *Рік народження*, тобто, знаючи рік народження, можна обчислити його вік. Ще один приклад: нехай працівники отримують, крім окладу, щомісячну премію, яка залежить від стажу, а саме — за кожний рік стажу нараховується один відсоток премії від окладу. Тоді можна створити запит із полем *Зарплата*, значення якого обчислюється на основі даних двох полів: $\text{Зарплата} = \text{Оклад} + (\text{Стаж} / 100) * \text{Оклад}$.

Для створення поля, що обчислюється, необхідно ввести вираз, який обчислює значення поля, у рядок **Поле** вільного стовпця бланку запиту.

Починаючи вивчення запитів із параметрами, слід пояснити, що вони створюються, зазвичай, за певною умовою. Наприклад, запит, який вибирає

з таблиці ПРАЦІВНИКИ прізвища робітників зі стажем більше 10 років, або запит, який вибирає з цієї таблиці робітників зі стажем більше 10, але менше 20 років. Як бачимо, наведені приклади запитів відрізняються незначно. Але це різні запити, які зберігаються у файлі бази даних, і можуть бути виконані в будь-який час. Виникає потреба у створенні запиту, у якому можна змінювати умову під час його виконання. Ці запити називають запитами з параметрами. Такі запити видають на екран повідомлення про необхідність введення певної умови з метою отримання необхідних результатів.

Послідовність і технологію створення запиту з параметрами доцільно розглядати на прикладі таблиць ПРАЦІВНИКИ й БУДИНКИ. Наприклад, можна створити запит з полями *Адреса* й *Рік будови* таблиці БУДИНКИ та з полями *Прізвище* й *Рік народження* таблиці ПРАЦІВНИКИ, причому умова з відбору записів буде виконуватися по полю *Рік будови*.

Що стосується запитів на створення й модифікацію таблиць, то їх доцільно розглядати в такій послідовності:

- запит на створення нової таблиці на основі даних, які зберігаються в існуючих таблицях;

- запит на додавання даних до існуючої таблиці;
- запит на оновлення даних в окремих полях таблиці;
- запит на видалення непотрібних записів із таблиць.

При цьому учні повинні розуміти, що запит на створення таблиці вибирає необхідні дані з однієї або кількох існуючих таблиць бази даних і розміщує ці дані в нову таблицю. Нова таблиця може бути створена в поточній БД або в іншій, ім'я якої вказується в процесі конструювання запиту. Такі запити створюються в основному в тих випадках, коли потрібно скопіювати або записати дані в архів.

Запити на додавання даних додають необхідну сукупність записів з однієї або декількох таблиць до однієї або кількох інших таблиць. Цей тип запиту не змінює дані в окремих полях, можна додавати тільки повні рядки даних. Зазвичай, початкові й кінцеві таблиці знаходяться в одній і тій же базі даних, хоча це не обов'язково.

Можна дотримуватися такої послідовності створення запиту на додавання даних:

- створюється запит на вибірку вже описаним способом;
- запит на вибірку перетворюється на запит на додавання;
- вибираються кінцеві поля в запиті на додавання;
- виконується запит на додавання записів.

Технологію створення такого запиту можна розглянути на прикладі раніше створеної таблиці НОВА. Нехай із таблиць ПРАЦІВНИКИ й БУДИНКИ потрібно додати до таблиці НОВА запис, у якому кількість квартир дорівнює 60, тобто такий рядок: 12/43 60 Майборода І.С. 2020.

Запитом з функціями також необхідно приділити важливу увагу. Обов'язково потрібно використовувати такі функції: Sum (підрахунок суми), Count (підрахунок кількості елементів), Avg (обчислення середнього значення), Min (пошук найменшого значення), Max (пошук найбільшого значення). Слід враховувати, що технологія створення запитів із функціями несуттєво відрізняється від технології створення звичайних запитів на вибірку. Вивчення запитів із функціями необхідно розглядати на конкретних прикладах.

Рекомендації щодо вивчення розділу «Форми».

Вивчення форм необхідно починати зі створення найпростіших форм: спочатку за допомогою інструменту **Форми**, а потім — **Розділити форму**. На вкладці **Створити** зліва в групі **Форми** розміщені дві кнопки **Форма** й

Розділити форму. Цим кнопкам відповідають два однойменних способи створення форм. Спосіб **Форма** можна розглянути на прикладі таблиці ПРАЦІВНИКИ. Необхідно звернути увагу учнів, що з таблиці ПРАЦІВНИКИ до форми введено тільки перший запис цієї таблиці, а під формою розміщені всі записи таблиці БУДИНКИ, які пов'язані з першим записом таблиці ПРАЦІВНИКИ. До цієї форми можна вносити зміни. Наприклад, можна змінити розмір поля відповідно до даних, які будуть внесені в таблицю. Але, як це робиться, буде розглянуто в наступних розділах.

Спосіб **Розділити форму** можна розглянути на прикладі таблиці БУДИНКИ. Форми, створені за цим способом, одночасно відображають дані в режимі форми й у режимі таблиці. Причому, як і за попереднім способом у формі відображається тільки один запис. Цей спосіб також дозволяє додавати, змінювати й вилучати дані у формі й у таблиці. У таблиці й формі можна працювати по черзі.

Інші способи створення форм доцільно розглянути лише у режимі огляду, а основну увагу слід приділити способу **Конструктор форм**. Цей спосіб має значну кількість елементів керування, за допомогою яких здійснюється доступ і робота з даними, які містяться в таблицях бази даних. Такими елементами є поле для введення, відображення й корегування даних, які зберігаються в комірках таблиці бази, прапорці, кнопки, перемикачі, списки, надписи, рамки об'єктів для відображення графіки й об'єктів OLE.

Необхідно звернути увагу на те, що форма, створена за допомогою **Конструктора форм**, найчастіше містить такі розділи: заголовок форми, верхній колонтитул, область даних, нижній колонтитул, примітка форми.

Порядок і технологію створення форми, наприклад, з ім'ям *форма3*, можна розглянути для таблиці БУДИНКИ з полями *Шифр*, *Адреса*, *Квартир* і *Прізвище*, причому перед першими трьома полями бажано ввести текст *Будівлі*, а перед прізвищем — *Обслуговують*.

Учні повинні усвідомити, що модифікація форми передбачає виконання таких дій:

- настроювання параметрів відображення форми;
- зміна властивостей форми;
- додавання елементів керування;
- зміна властивостей елементів керування;
- видалення елементів керування.

Настроювання параметрів відображення форми, порядок і технологію зміни її властивостей, додавання елемента керування до існуючої форми, зміну властивостей елементів керування, видалення елементів керування доцільно розглядати на прикладі *форми 3*.

Рекомендації щодо вивчення розділу «Звіти».

Звіт необхідно розглядати як об'єкт бази даних, призначений для відображення даних у зручній формі, близької до повноцінних документів. Звіт — це фактично інструмент для виведення на екран і на друкарський пристрій необхідних даних у вигляді, які задовольняють потреби користувача.

Слід звернути увагу учнів на те, що технологія створення звітів близька до технології створення форм. Для їх розробки використовуються ті команди й інструменти, якими доводилося користуватися в процесі створення форм. Але якщо форми забезпечують введення і модифікацію даних у базі даних, то звіти — зручний їх перегляд і друк. Тому звіти бажано вивчати у формі самостійної роботи.

Спочатку необхідно оволодіти методикою **автоматичного створення звітів** — самого простого і швидкого способу створення звіту. Звіт за цим способом формується одразу без визначення додаткової інформації. Такий

звіт можна одразу переглянути, відправити по e-mail або роздрукувати. Технологію створення звіту можна розглянути на прикладі таблиці ПРАЦІВНИКИ.

Під час вивчення методики створення звіту в режимі конструктора необхідно пояснити, що основними розділами звіту в режимі конструктора є такі:

- **Заголовок звіту** — розміщується на початку звіту перед верхнім колонтитулом.;
- **Верхній колонтитул звіту** — розміщується у верхній частині сторінки;
- **Заголовок сторінки** — друкується перед кожною новою групою записів звіту;
- **Подобиці** — є основним змістом звіту;
- **Нижній колонтитул сторінки** — використовується для нумерації сторінок й інформації по них. Друкується в нижній частині сторінки;
- **Нижній колонтитул звіту** — містить підсумкову інформацію про звіт. Друкується в кінці звіту.

Основні елементи керування, які можна використовувати в процесі створення звіту, розміщені в групі **Елементи керування** вкладки **Конструктор (Назва, Номер сторінки, Надпис, Дата та час та інші)**.

Порядок і технологію створення звіту в режимі конструктора можна розглядати для бази даних `vik_1` на прикладі таблиць ПРІЗВИЩА і БУДИНКИ. Наприклад, до звіту з таблиці ПРАЦІВНИКИ можна використати поля *Прізвище* і *Телефон*, а з таблиці БУДИНКИ — поля *Шифр*, *Адреса* і *Рік будови*.

У процесі практичної роботи з розробленими звітами доцільно використовувати такі операції:

- переглядати звіт перед його друкуванням;
- друкувати звіт;
- перетворювати звіт у формат MS Word;
- створювати знімок звіту для його відправлення електронною поштою.

Завершити вивчення розділу бажано виконанням групового проекту з використанням СУБД, наприклад бази даних учнів класу з можливістю пошуку номера телефона, адреси телекомунікаційної програми, дня народження, уподобань, спортивних досягнень тощо.

ЛІТЕРАТУРА

1. Програма курсу «Інформатика, 8-9 класи загальноосвітніх навчальних закладів з поглибленим вивченням інформатики» http://old.mon.gov.ua/ua/activity/education/56/692/educational_programs/1384763942/
2. Дорошенко Ю.О. Технологічне навчання інформатики : Навчально-методичний посібник / Ю. О. Дорошенко, Т. В. Тихонова, Г. С. Луньова. – Х. : Вид-во «Ранок», 2011.- 304с. .
3. Інформатика 5–11 класи: навчальні програми, методичні рекомендації щодо організації навчально-виховного процесу в 2016-2017 навчальному році з коментарем провідних фахівців. – Харків : Видавництво «Ранок», 2016. – 176 с.
4. Інформатика 2–11 класи: методичні рекомендації щодо організації навчально-виховного процесу в 2015-2016 навчальному році з коментарем провідних фахівців. – Харків : Видавництво «Ранок», 2015. – 96с.
5. Математика. Інформатика : методичні рекомендації МОН України щодо організації навчального процесу в 2017/2018 навчальному році; оновлені на компетентнісній основі навчальні програми 5-9 класів; методичні коментарі провідних науковців щодо впровадження ідей Нової української школи. К. : УОВЦ «Оріон», 2017. – 96с.

НАВЧАЛЬНЕ ВИДАННЯ

Лапінський Віталій Васильович
Семко Лариса Петрівна
Семененко Ірина Миколаївна

ІНФОРМАТИКА 9

методичний посібник

За авторською редакцією

Верстка – Мирончик Ю.П.
Обкладинка – Лук'яненко Л.

Підписано до друку 03.10.2018 р. Формат 60x90 1/16
Гарнітура Петербург. Друк. офсетний. Папір офсетний.
Ум.друк. 6,0 арк.
Наклад 300 пр.

Віддруковано у ТОВ «КОНВІ ПРІНТ».
03680, м. Київ, вул. Антона Цедіка, 12
тел. +38 044 332-84-73.

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців,
виготовлювачів і розповсюджувачів
видавничої продукції серія
ДК № 6115, від 29.03.2018 р.