

Г.А. ЗАБАРОЎСКІ А.Я. ПУПЦАЎ

ІНФАРМАТЫКА

11



Г. А. Забароўскі А. Я. Пупцаў

ІНФАРМАТЫКА

Вучэбны дапаможнік для 11 класа
агульнаадукацыйных устаноў
з беларускай мовай навучання

*Дапушчана
Міністэрствам адукацыі
Рэспублікі Беларусь*

Мінск
«Народная асвета»
2010
Правообладатель Народная асвета

УДК 004(075.3=161.3)

ББК 32.81я721

3-41

Аўтары:

Г. А. Забароўскі (раздзелы 1, 2), А. Я. Пупцаў (раздзелы 1, 3, 4)

Пераклад з рускай мовы *Н. М. Алганавай*

Рэцэнзенты:

кафедра інфарматыкі і камп'ютарнага мадэліравання
Гродзенскага дзяржаўнага ўніверсітета імя Янкі Купалы
(канд. пед. навук, дацэнт *Н. П. Макарава*);
старшы выкладчык кафедры прыродазнаўчых дысцыплін
і інфармацыйных тэхналогій Мінскага абласнога інстытута
развіцця адукцыі *А. А. Буслайскі*

Забароўскі, Г. А.

3-41 Інфарматыка : вучэб. дапам. для 11-га кл. агульнаадукац. устаноў з беларус. мовай навучання / Г. А. Забароўскі, А. Я. Пупцаў; пер. з рус. мовы Н. М. Алганавай. — Мінск : Нар. асвета, 2010. — 150 с. : іл.

ISBN 978-985-03-1371-3.

УДК 004(075.3=161.3)

ББК 32.81я721

ISBN 978-985-03-1371-3

© Забароўскі Г. А., Пупцаў А. Я., 2010

© Алганава Н. М., пераклад на беларускую мову, 2010

© Афармленне. УП «Народная асвета»,
2010

Правообладатель Народная асвета

Ад аўтараў

Паважаныя школьнікі!

У адзінаццатым класе Вы працягнече паглыбляць свае веды ў вобласці інфармацыйных тэхналогій, асноў алгарытмізацыі і праграмавання.

У першым раздзеле вучэбнага дапаможніка Вы пазнаёміцесь з інструментамі і метадамі вэб-канструявання, вывучыце асновы мовы разметкі гіпертэкставых дакументаў HTML, даведаецца пра асаблівасці падрыхтоўкі графікі для сеткі Інтэрнэт, навучыцесь ствараць вэб-сайты.

У другім раздзеле будзе працягнута вывучэнне асноў алгарытмізацыі і праграмавання. Атрыманыя раней веды Вы навучыцесь выкарыстоўваць для решэння практычных задач з розных предметных абласцей.

Трэці раздзел прысвечаны апрацоўцы інфармацыі ў сістэмах кіравання базамі даных.

Чацвёрты, апошні, раздзел заканчвае знаёмства з інфармацыйнымі сістэмамі і тэхналогіямі.

Пытанні пад знакам  прызначаны для замацавання вывучацца матэрыялу. Дадатковы матэрыял для цікавых вылучаны знакам .

Пасля параграфаў і пунктаў у дапаможніку пропануюцца практыкаванні, якія дазволяюць Вам замацаваць свае веды і практычныя ўменні работы на камп'ютары.

Жадаем поспехаў у вывучэнні інфарматыкі і інфармацыйных тэхналогій. Добрая веды дапамогуць Вам выбраць професію і прыносіць карысць Радзіме.

АСНОВЫ ВЭБ-КАНСТРУЯВАННЯ

§ 1. Уяўленне аб вэб-канструяванні

1.1. Інструменты і метады распрацоўкі вэб-сайтаў

Большасць інфармацыйных рэсурсаў у сетцы Інтэрнэт прадстаўлена ў выглядзе вэб-сторонак, якія аб'ядноўваюцца ў вэб-сайты. Дзякуючы развіццю тэлекамунікацыйных тэхналогій размешчаная на вэб-сторонках інфармацыя даступная мноству людзей. У адрозненне ад інфармацыі на папяровых носьбітах, напрыклад кніг, яна можа аператыўна змяняцца і практычна імгненна дастаўляцца ў любое месца па запыце карыстальніка.

Для стварэння вэб-сторонак выкарыстоўваецца мова разметкі гіпертэкставых дакументаў HTML (*HyperText Markup Language*).

У якасці інструментаў вэб-канструявання могуць выкарыстоўвацца офісныя праграмы (напрыклад, Word, PowerPoint, што ўваходзяць у склад Microsoft Office), якія не з'яўляюцца спецыяльнымі сродкамі распрацоўкі вэб-сторонак. Лепшыя вынікі можна атрымаць, выкарыстаўшы спецыяльныя праграмы — вэб-редактары, напрыклад Microsoft FrontPage.

Вылучаюць дзве асноўныя групы метадаў і адпаведных інструментаў распрацоўкі вэб-сайтаў: *візуальныя і ручныя* (праграмныя).

Візуальныя метады дазваляюць выконваць усе работы па стварэнні вэб-сторонак з высокай ступенню аўтаматызацыі і не патрабуюць ведання мовы разметкі HTML. Яны памяншаюць працаёмкасць і час распрацоўкі сайта. Сутнасць візуальных метадаў адлюстравана ў прынцыпе WYSIWYG (ад англ. *What you see is what you get* — Што бачыш, тое і атрымаеш).

Распрацавана шмат спецыяльных інструментаў — рэдактараў візуальнага канструявання. Найбольш вядомымі з'яўляюцца Microsoft FrontPage, Adobe (Macromedia) Dreamweaver, NanoWebEditor і інш. З дапамогай такіх рэдактараў вэб-сторонкі ствараюцца (рысуюцца) у інтэрактыўным рэжыме, пры гэтым аўтаматычна генерыруецца адпаведны HTML-код, які ўяўляе сабой набор камандаў мовы разметкі HTML.

Адзначым, што ўзгаданыя рэдактары дазваляюць выконваць усе работы па стварэнні сайта без непасрэднага падключэння да сеткі Інтэрнэт, або, як кажуць, у рэжыме *афлайн* (*offline*). Затым створаны вэб-сайт публікуецца ў сетцы Інтэрнэт, г. зн. размяшчаецца на вэб-серверы.



У апошні час для канструявання і супраджэння вэб-сайтаў выкарыстоўваюцца сістэмы кіравання іх змесцівам (кантэнтам) — CMS (*Content Management System*), якія прапануюцца спецыяльнымі платнымі або бясплатнымі службамі. Сістэмы CMS уяўляюць сабой канструктары. Яны дазваляюць ствараць сайт у рэжыме непасрэднага падключэння да сеткі, або **анлайн** (*online*), і супраджаць яго далей. Пры стварэнні структуры сайта і распрацоўцы навігацыі па ім у сістэме CMS таксама не патрабуецца ведаць мову разметкі HTML.

Сярод бясплатных сістэм CMS найбольш папулярныя Joomla! (<http://joomla.ru/>) і Drupal (<http://www.drupal.ru>).

Вядома, цяжка стварыць добры сайт, не ведаючы асноў мовы разметкі HTML. Для работы з HTML-кодам могуць выкарыстоўвацца спецыяльныя інструменты распрацоўкі, якія дазваляюць набіраць каманды HTML **уручную**, напрыклад рэдактары HotDog, Adobe HomeSite і інш. Гэтыя інструменты робяць больш простымі ўвод і рэдагаванне кода. Аднак, набіраць асноўныя каманды (**тэгі**) мовы HTML можна нават у найпрасцейшым тэкставым рэдактары **Блокнот**, а праглядаць вынікі работы можна з дапамогай браўзера.

Важную ролю ў выбары інструментаў і метадаў вэб-канструявання адыгрывае статычнасць або дынамічнасць ствараемых старонак, а таксама наяўнасць інтэрактыўных элементаў. *Статычныя* старонкі адлюстроўваюцца браўзерам карыстальніка ў тым выглядзе, у якім былі створаны і размешчаны на вэб-серверы. *Дынамічныя* старонкі генерыруюцца па запыце карыстальніка — інфармацыя на іх загружаема сервернымі праграмамі з баз даных. Такія базы даных звычайна змяшчаюць інфармацыю, якая патрабуе пастаяннага абнаўлення. Напрыклад, інтэрнэт-магазін атрымлівае звесткі аб новых таварах і цэнах, што змяніліся.

Інтэрактыўныя (г. зн. кіруемыя карыстальнікам) элементы вэб-старонак выкарыстоўваюцца для ўвядзення пароля, выбару тавару, ацэньвання яго якасці, уводу адказаў пры тэсціраванні або галасаванні, для фарміравання запыту для пошуку інфармацыі ў аддаленай базе даных і да т. п.



Акрамя мовы разметкі гіпертэкставых дакументаў HTML, пры стварэнні вэб-сайтаў выкарыстоўваюць і спецыяльныя мовы вэб-праграмавання. Шырокое прымененне атрымала мова сцэнарыяў Java Script. Напісаныя на ёй канструкцыі, або скрыпты, устаўляюцца непасрэдна на вэб-старонкі і інтэрпретуюцца браўзерам. Яны выкарыстоўваюцца для стварэння асобых, як правіла, інтэрактыўных элементаў вэб-старонак, напрыклад дынамічных меню, гадзіннікаў, календароў, форм запытаў, лічыльнікаў наведвання старонак, сістэм галасавання і да т. п. Для сервернага праграмавання найбольш

часта выкарыстоўваюць мовы PHP (ад англ. *Hypertext Preprocessor* — прэпрацесар гіпертэксту), PERL (ад англ. *Practical Extraction and Report Language* — практычна мова для здабывання даных і стварэння справа-здач).



1. У чым розніца паміж візуальнymi і ручнымі метадамі вэб-канструявання?
2. Якія інструменты могуць выкарыстоўвацца пры стварэнні вэб-сайтаў?
3. Якія старонкі называюць статычнымі? Дынамічнымі?

1.2. Праектаванне сайта

Вылучаюць наступныя асноўныя этапы распрацоўкі вэб-сайтаў:

- вызначэнне тэматыкі сайта, яго мэт і задач;
- проектаванне структуры сайта, стварэнне раздзелаў і сувязей паміж старонкамі;
- распрацоўка дызайну сайта, г. зн. стылю афармлення старонак;
- падрыхтоўка матэрыялаў (тэкстаў і графічных аб'ектаў) для размяшчэння на вэб-старонках;
- канструяванне старонак сайта (стварэнне HTML-кода);
- размяшчэнне ў сетцы (публікацыя) і тэсціраванне сайта.

Разгледзім на прыкладзе, як спраектаваць вэб-сайт кінатэатра.

Вызначым асноўную мэту сайта: прывабіць гледачоў, і задачы: інфармаванне аб рэпертуары кінатэатра, рэклама фільмаў.

Распрацоўку праекта пачнём са стварэння інфармацыйнай мадэлі сайта. Няхай для прастаты наш першы сайт будзе складацца з чатырох вэб-старонак. Структуру гэтага сайта для нагляднасці пакажам у выглядзе двухуздоўневай схемы (рыс. 1.1).



Рыс. 1.1

Правообладатель Народная асвета

На першым (верхнім) узроўні схемы змесцім першую (галоўную) вэб-сторонку. На ёй будуць размешчаны: агульная інфармацыя аб кінатэатры (напрыклад, фатаграфія, адрес) і тэкставыя гіперспасылкі.

На другім узроўні схемы змесцім вэб-сторонкі, прысвечаныя фільмам пэўных жанраў, напрыклад «Драмы», «Камедыі», «Дэтэктывы». Яны маюць падобную структуру і будуць адрознівацца толькі зместам (рыс. 1.2).

Структура сайта ў далейшым можа быць ускладнена дадаваннем старонак наступных узроўняў, напрыклад старонак, прысвечаных акцёрам, рэжысёрам, сцэнарыстам, кампазітарам і да т. п.

Цяпер спраектуем дызайн сайта. Асноўнымі структурнымі элементамі афармлення вэб-сторонак з'яўляюцца тэкставыя блокі (асноўны тэкст, загалоўкі, спісі, тэкставыя гіперспасылкі) і графічныя аб'екты: відaryсы (рысункі, фатаграфіі, анімацыі), «шпалеры» (фонавыя рysункі), відaryсы-гіперспасылкі, раздзяляльныя лініі. Для размяшчэння элементаў у вызначаных месцах старонак будзем выкарыстоўваць табліцы.

Спалучэнне ўласцівасцей элементаў вэб-сторонкі вызначае стыль яе афармлення. **Стыль тэксту** задаецца сукупнасцю параметраў фармату: шрыфт, памер, напісанне; водступы, выраўноўванне, міжсімвальны і міжрадковы інтэрвалы і інш. **Стыль графічных элементаў** задаецца сукупнасцю параметраў формы: памер, колер, фактура матэрыялу, а таксама разнастайных эффектаў: цень, бляск, празрыстасць і да т. п.

Для стылёва граматнага афармлення дакументаў зручна выкарыстоўваць готовыя шаблоны — тэмы. Напомнім, што **тэмай** называюць спецыяльна распрацаваны набор элементаў афармлення і колеравых схем дакумента. З дапамогай тэмы можна задаць стыль афармлення як асобнай старонкі, так і ўсяго сайта. Тэмы дазваляюць задаваць стылі асноўнага тэксту, загалоўкаў, гіперспасылак, спіскаў; колер фону; «шпалеры»; колер і таўшчыню меж табліцы і да т. п. Тэмы пропануюцца ва ўсіх офісных праграмах, аднак найбольшую карысць прыносіць іх выкарыстанне пры стварэнні презентацый і вэб-сайтаў.

Усе старонкі аднаго ўзроўню будзем афармляць у адзінным стылі.

Правообладатель Народная асвета

Назва фільма	
Фрагмент фільма (фатаграфія)	Жанр Рэжысёр Акцёры Студыя, год
Кароткі змест	

Рыс. 1.2

Непасрэднае канструяванне старонак сайта пачнём з падрыхтоўкі ўсіх яго элементаў (тэкстаў, рыsunкаў). Затым гэтыя элементы і (або) спасылкі на іх будзем устаўляць у гатовыя шаблоны.

Файлы ўсіх старонак вэб-сайта будзем захоўваць у адной папцы, напрыклад KINO. Гэта зробіць больш простым размяшчэнне сайта на вэб-серверы. Пры вялікай колькасці старонак або відарысаў іх трэба захоўваць ва ўкладзеных папках, напрыклад KINO\FOTO\. Файлу галоўнай вэб-старонкі звычайна даюць імя index або main з пашырэннем .htm.

Толькі пасля размяшчэння файлаў у папках можна ствараць гіперспасылкі, з дапамогай якіх выконваецца навігация па сайце — пераходы са старонкі на старонку. У нашым прыкладзе зручна спачатку распрацаваць старонкі другога ўзроўню, а затым аформіць галоўную старонку і стварыць гіперспасылкі для выкліку старонак ніжэйшага ўзроўню. Затым трэба праверыць работу спасылак у рэжыме афлайн. Толькі пасля ўважлівай праверкі і выпраўлення памылак створаны вэб-сайт можна апублікаваць, г. зн. размясціць на вэб-серверы.



1. Якія этапы распрацоўкі вэб-сайта можна вылучыць?
2. Што разумеюць пад стылем афармлення вэб-старонкі?

Практыкаванне

Распрацуйце праект вэб-сайта па адной з тэм: Мая сям'я. Мая школа. Mae сябры. Mae любімая справы. Мая Радзіма — Беларусь. Прырода роднага краю. Цікавыя прафесіі.

§ 2. Выкарыстанне офісных дадаткаў для стварэння вэб-старонак

2.1. Стварэнне вэб-старонак у рэдактары MS Word

Любы дакумент MS Office можна захаваць у выглядзе вэб-старонкі. Для гэтага дастаткова выканаць каманду **Файл → Сохранить как веб-страницу**. Аднак далёка не ўсякі падрыхтаваны для друку на паперы дакумент будзе добра выглядаць у браўзера. Найлепшыя вынікі можна атрымаць, калі спецыяльна рыхтаваць дакумент для размяшчэння ў сетцы Інтэрнэт.

Выкарыстанне тэкставага рэдактара MS Word з'яўляецца адным з самых простых спосабаў распрацоўкі вэб-старонак. Разгледзім на прыкладзе, як гэта робіцца. Пачнём са стварэння старонак ніжняга ўзроўню, а затым аформім галоўную старонку і дададзем гіперспасылкі.

Правообладатель Народная асвета

The screenshot shows a web browser window with the title bar "Анастасія Слуцкая". The address bar displays "file:///D:/!Work/KINO/drama.htm". The main content area features a painting of a woman in a crown and ornate robes, identified as the protagonist of the movie. To the right of the image, there is a box containing film details:

- Жанр:** гістарычна драма
- Рэжысёр:** Юрый Елхай
- Акцёры:** Святлана Зелянкоўская, Генадзь Давыдзька, Анатоль Кот, Фуркат Файзіеў, Сяргей Глушко...
- "Беларусьфільм", 2003 г.**

Below the image and details, there is a descriptive text block:

Пачатак XVI стагоддзя. Багатыя землі ў цэнтры Еўропы на перасячэнні гандлёвых шляхоў становіцца жаданай здабычай для суседзяў. Полчышчы татар нападаюць з поўдня. Хуткімі набегамі яны дасягаюць самых аддаленых беларускіх гарадоў. Падаюць крэпасці, апустошваюцца землі. Бястрашная княгіня Анастасія Слуцкая ўстае на абарону беларускай зямлі ад незлічоных полчышчаў, што нясуць смерць.

Рыс. 1.3

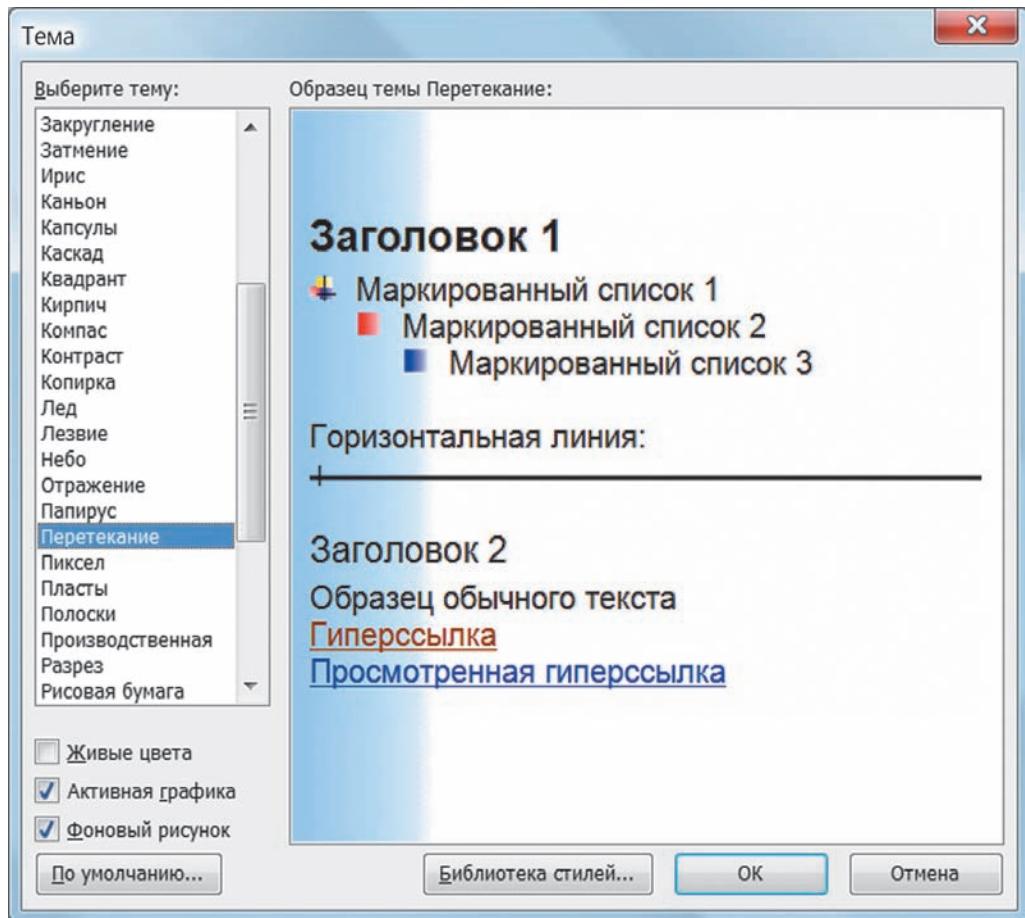
Прыклад 1. Стварыць вэб-сторонку фільма «Анастасія Слуцкая» (рыс. 1.3).

Перш за ўсё з дапамогай каманды **Файл** → **Создать** выберам пункт **Веб-страница** (гэты пункт можна выбраць і ў вобласці задач **Создание документа**). Затым, выканавшы каманду **Формат** → **Тема**, выберам для аформлення старонкі тэму **Перетекание** (рыс. 1.4).

Для размяшчэння на старонцы фатаграфіі і тэкстаў выкарыстаем табліцу з двух радкоў і двух слупкоў. Аб'яднаем ячэйкі ніжняга радка. Уставім падрыхтаваныя загадзя тэксты і відарысы з файлаў. Аформім іх у адпаведнасці з рисункам 1.3.

Створаны дакумент захаваем як вэб-сторонку ў папцы KINO пад іменем drama.htm. Выберам тып захаванага файла **Веб-страница (*.htm; *.html)**. Пры гэтым усе выкарыстаныя ў дакумэнце відарысы будуць змешчаны ў асобную папку з іменем вэб-сторонкі і пашырэннем .files (у нашым прыкладзе — drama.files).

Правообладатель Народная асвета



Рыс. 1.4

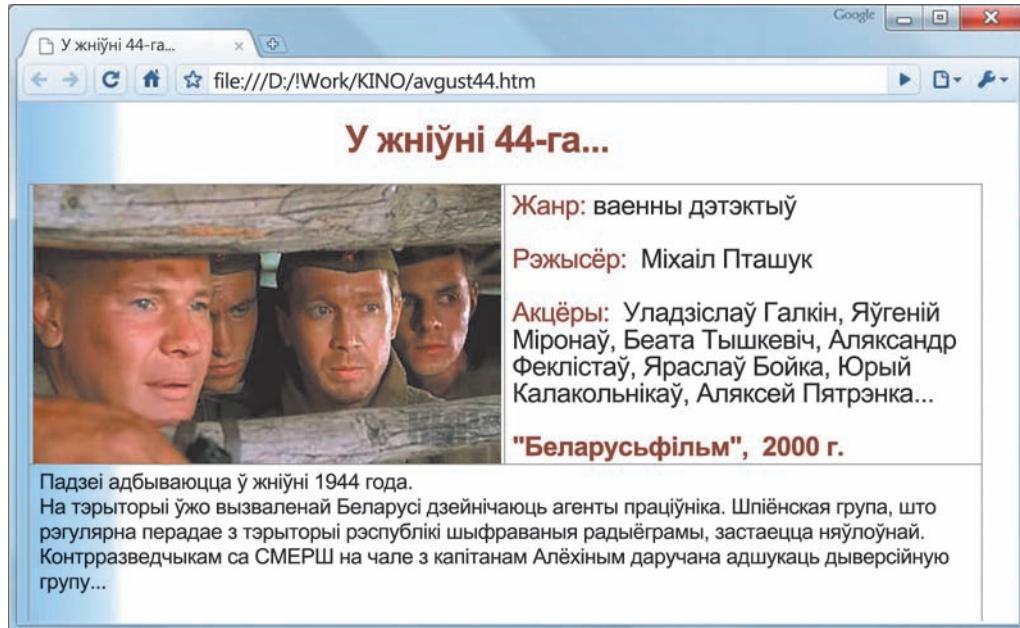
Аналагічным чынам ствараюца старонкі іншых фільмаў і жанраў, напрыклад «Камедыі» і «Дэтэктывы». Паколькі яны маюць падобную структуру і адрозніваюца толькі зместам, то новыя старонкі можна атрымліваць рэдагаваннем ужо створаных, змяняючи відарысы і тэксты і захоўваючи іх пад новымі імёнамі.

Створым вэб-старонку фільма «У жніўні 44-га...» (рыс. 1.5) на аснове ўжо створанай намі старонкі «Анастасія Слуцкая». Захаваем яе пад іменем august44.htm.

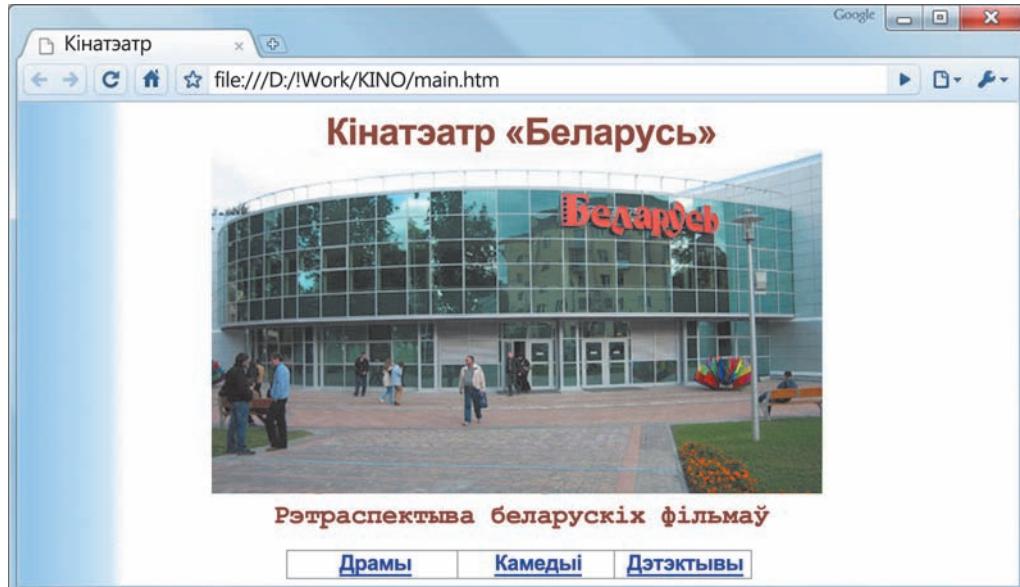
Прыклад 2. Стварыць галоўную старонку сайта кінатэатра «Беларусь» (рыс. 1.6), якая змяшчае гіперспасылкі на старонкі фільмаў.

Для галоўной старонкі сайта выкарыстаєм тэму **Перетекание**.

Правообладатель Народная асвета



Рыс. 1.5



Рыс. 1.6

Правообладатель Народная асвета

Важны этап распрацоўкі вэб-сторонкі кінатэатра — стварэнне гіперспасылак, якія адкрываюць старонкі фільмаў. Для размяшчэння гіперспасылак створым табліцу. Затым для стварэння тэкстовых гіперспасылак выканаем наступныя дзеянні:

- Вылучым тэкст гіперспасылкі, напрыклад слова «Драмы».
- З дапамогай каманды **Вставка** → **Гиперссылка** або кнопкі  на панэлі інструментаў выберам пункт **Связать с файлом, веб-страницей**, вылучым імя файла drama.htm і пацвердзім выбар націсканнем кнопкі ОК.

Гіперспасылкі ў выглядзе відарысаў ствараюцца аналагічна.

Скончышы стварэнне галоўнай старонкі, не забудзем захаваць яе пад іменем main.htm у той жа папцы, у якой раней захоўвалі старонкі фільмаў.

Праглядзім створаныя старонкі ў браўзеры. Праверым работу спасылак, перайшоўши па іх на старонкі фільмаў і вярнуўшыся на галоўную старонку з дапамогай кнопкі **Назад** браўзера.



Павысіць прывабнасць вэб-сторонкі можна, змясціўши на ёй дынамічныя і інтэрактыўныя элементы (анімацыі, аўдыё- і відэафрагменты, формы апытанняў). Пры гэтым трэба памятаць, што анімацыі часта адцягваюць увагу карыстальнікаў ад тэкставай інфармацыі, а вялікія памеры аўдыёвідэафайліў істотна запавольваюць загрузку старонак.



Прыклад 3. Размясціць на галоўнай старонцы сайта кінатэатра «Беларусь» бягучы радок.

З дапамогай каманды **Вид** → **Панели инструментов** актыўізуем панэль **Веб-компоненты** (рыс. 1.7). Націснем на гэтай панэлі кнопкі .

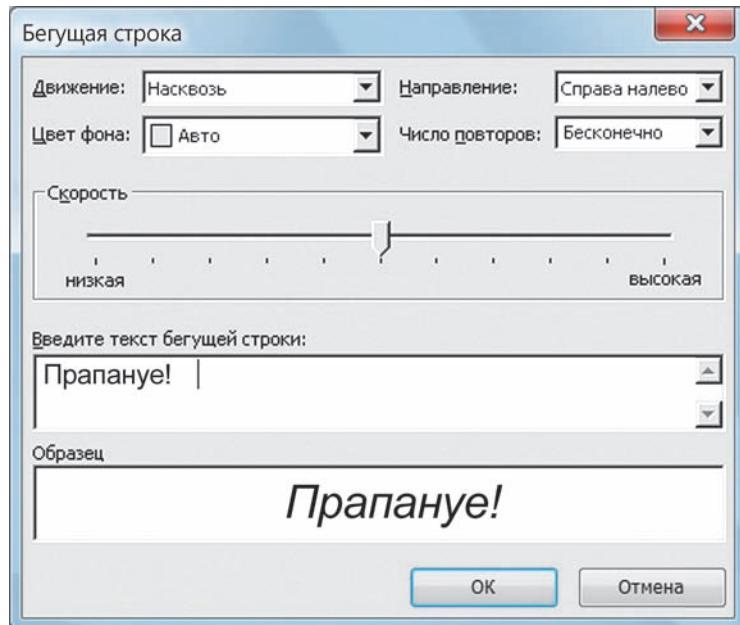


Рис. 1.7

У акне **Бегущая строка** (рыс. 1.8), што адкрыеца, набяром патрэбны тэкст, напрыклад «**Пропануе!**». Зробім неабходныя ўстаноўкі: выберам напрамак руху **Справа налево** і націснем кнопкі ОК.

Захаваем змененую старонку.

Рэдактар MS Word дазваляе падключыць да вэб-сторонкі відэа- і гукавыя файлы. Для гэтага неабходна націсканнем значка **Звук**  або **Фільм**  на панэлі **Веб-компоненты** адкрыць адпаведнае дыялогавае акно, націснуць кнопкі **Обзор**, знайсці патрэбны відэа- або гукавы файл, устанавіць параметры



Рыс. 1.8

прайгравання, напрыклад колькасць паўтораў, і пацвердзіць выбар націсканнем кнопкі ОК.



1. Як можна захаваць дакумент MS Word у выглядзе вэб-сторонкі?
2. З дапамогай якіх дзеянняў можна стварыць гіперспасылку?

Практыкаванне

Стварыце фрагмент сайта па адной з тэм: Мая Радзіма — Беларусь. Мая школа. Мае сябры. Мае любімая заняткі. Мая сям'я (прыклад галоўнай старонкі паказаны на рисунку). У якасці гіперспасылак выкарыстайце карцінкі з калекцыі кліпаў.



Правообладатель Народная асвета

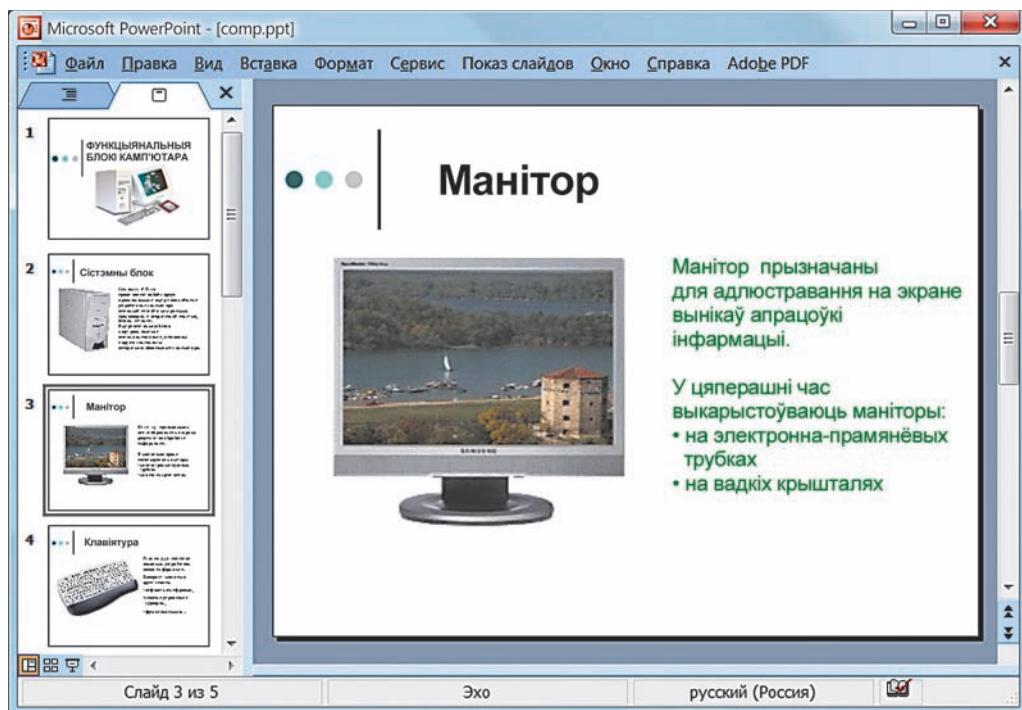
2.2. Захаванне презентацыі PowerPoint у выглядзе вэб-старонак

Презентацыя PowerPoint, як правіла, змяшчае некалькі слайдаў, кожны з якіх можа быць захаваны як асобная вэб-старонка. Усю презентацыю можна захаваць як сайт, структура якога будзе дакладна адпавядаць структуры презентацыі.

Прыклад 1. Стварыць фрагмент сайта на аснове презентацыі «Функцыянальныя блокі камп’ютара».

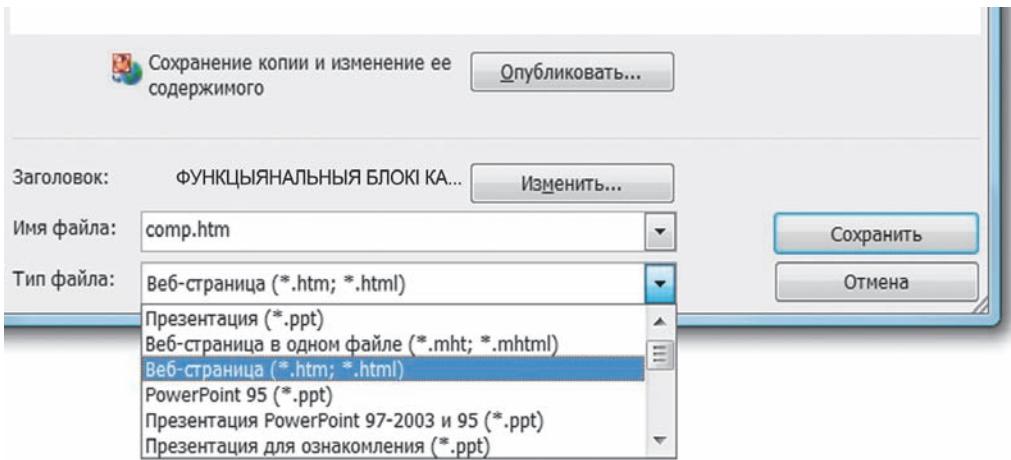
Адкрыем у рэдактары PowerPoint презентацыю «Функцыянальныя блокі камп’ютара» (файл comp.ppt). Прагледзім яе структуру. Презентацыя складаецца з пяці слайдаў, якія змяшчаюць тэксты і відарысы (рыс. 1.9).

Для захавання презентацыі ў выглядзе вэб-старонак у меню **Файл** выберам пункт **Сохранить как веб-страницу**. У акне **Сохранение документа** выберам тып файла для захавання **Веб-страница (*.htm, *.html)**, увядзём імя файла або па-



Рыс. 1.9

Правообладатель Народная асвета



Рыс. 1.10

кінем старое (у нашым прыкладзе — comp.htm) і націнем кнопкі **Сохранить** (рыс. 1.10).

Пры выкарыстанні прапанаваных па ўмаўчанні настроек усе слайды презентацыі захаваюцца ў папцы comp.files у выглядзе асобных старонак, якія можна будзе перагортваць у правай частцы акна браўзера з дапамогай гіперспасылак. Гэтыя гіперспасылкі размешчаны ў левай частцы захаванай старонкі comp.htm (па ўмаўчанні белага колеру на чорным фоне).

Зменім гэтыя настройкі. Для гэтага націскнем кнопкі **Опубликовать** выклічам акно **Публикация веб-страницы** (рыс. 1.11) і настроім параметры вэбдокумента. У гэтым акне можна выбраць аб'екты для публікацыі (**Полная презентация** або **Слайды**), падтрымку браўзераў, а таксама змяніць загаловак вэб-старонкі. Па ўмаўчанні ў якасці аб'екта для публікацыі прапануецца **Полная презентация**. У гэтым выпадку ў выглядзе вэб-старонак будуць захаваны ўсе слайды.

Колеравыя схемы, эфекты анімацыі і іншыя параметры настройваюцца ў акне **Параметры веб-документа** (рыс. 1.12), якое выклікаецца націскнем кнопкі **Веб-параметры**.

На ўкладцы **Общие** выберам колеравую схему **Цвета презентации (выделение)**. У гэтым выпадку фон вэб-старонак будзе адпавядаць фону слайдаў презентациі.

Уключаны па ўмаўчанні флагок **Добавить панель смены слайдов** можна здымаць толькі ў тым выпадку, калі на слайдах загадзя змешчаны гіперспасылкі, якія забяспечваюць пераходы паміж слайдамі.

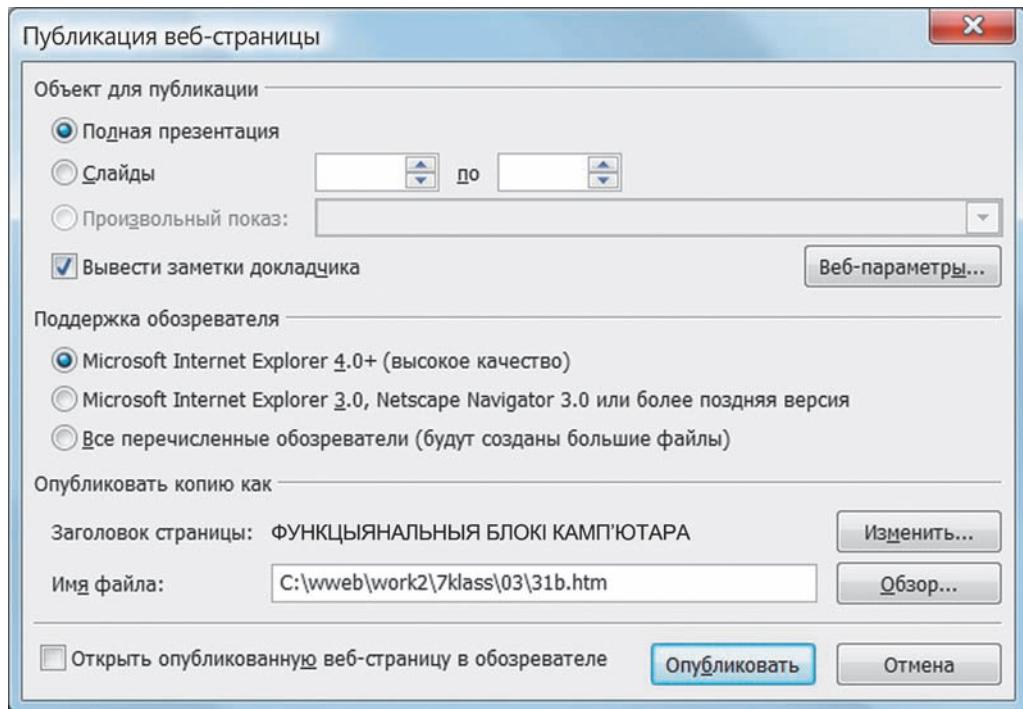


Рис. 1.11

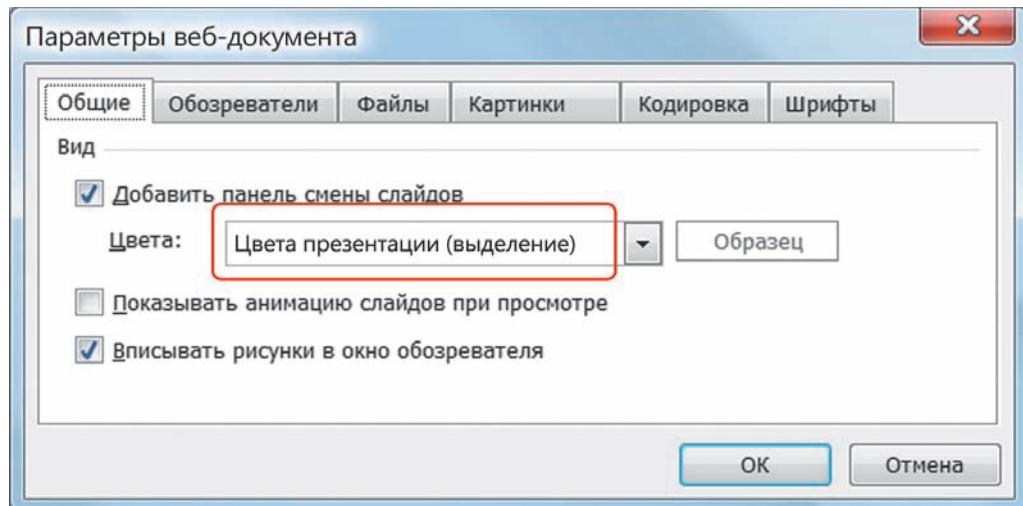
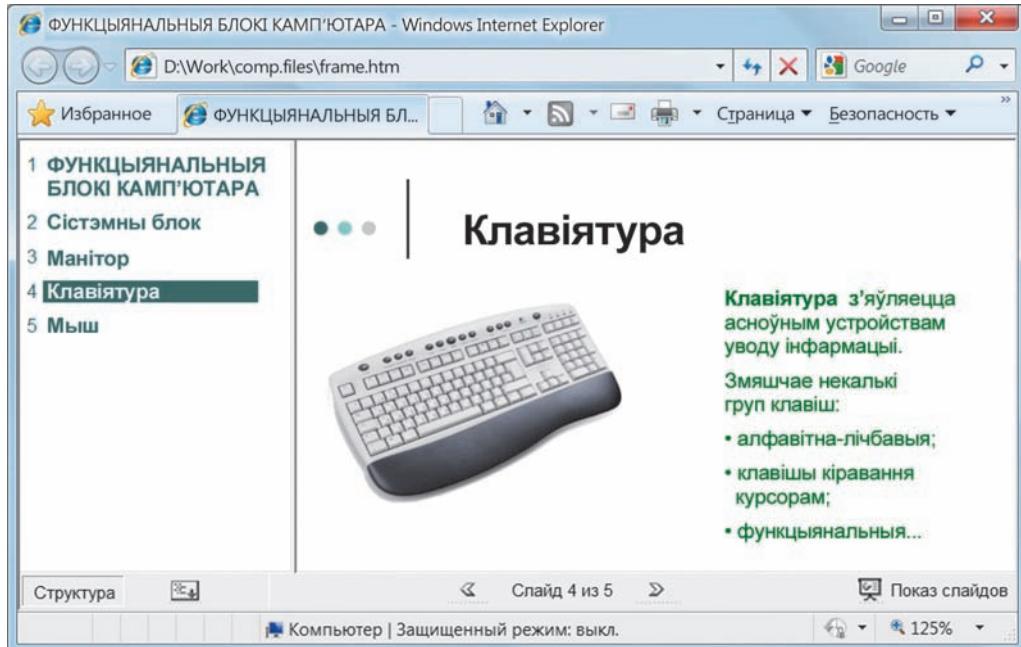


Рис. 1.12

Правообладатель Народная асвета



Рыс. 1.13

Захаваем дакумент. Прагледзім захаваныя вэб-старонкі. Выгляд адной з іх паказаны на рымсунку 1.13.

На заканчэнне адзначым, што захаванне дакумента ў выглядзе вэб-старонак з'яўляецца самым простым спосабам стварэння вэб-сайтаў і шырока выкарыстоўваецца ў адукцыі для размяшчэння ў сетцы разнастайных матэрыялаў вучэбнага прызначэння: сачыненняў, рэфератаў, дакладаў і презентаций. У прафесіянальным вэб-канструяванні гэтыя інструменты і методы практычна не выкарыстоўваюцца з-за неаптымальнасці HTML-кода атрыманых старонак, што абцяжарвае іх рэдагаванне і прыводзіць да вельмі вялікіх памераў файлаў і павольнай загрузкі старонак.

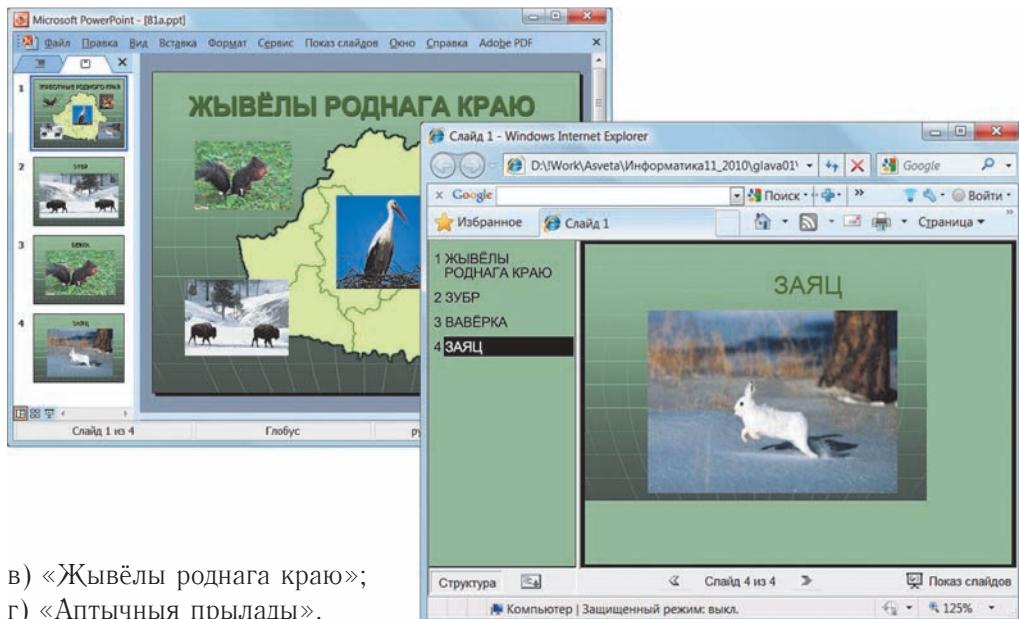
- ?**
1. З дапамогай якіх дзеянняў можна захаваць презентацыю PowerPoint у выглядзе вэб-документа?
 2. Якія параметры вэб-старонак можна настройваць?

Практыкаванне

Стварыце фрагмент сайта на аснове гатовай презентацыі:

- «Электрычны ток»;
- «Прасторавыя фігуры»;

Правообладатель Народная асвета



- в) «Жывёлы роднага краю»;
г) «Аптычныя прылады».

§ 3. Асновы мовы разметкі HTML

3.1. Стварэнне HTML-документа ў рэдактары Блокнот

Вэб-старонка ўяўляе сабой тэксты выявы документ, у якім расстаўлены каманды мовы HTML. Яны інтэрпрэтуюцца браўзерам. Напрыклад, такія каманды могуць паказваць, як павінен адлюстроўвацца змест старонкі на экране.

HTML-документ можна ствараць у найпрактычнейшых тэкстовых рэдактарах, напрыклад у рэдактары **Блокнот**, а затым захоўваць у файле з пашырэннем .htm або .html.

Разметка HTML-документа заключаецца ў расстаноўцы **тэгаў** — змешчаных у вуглавыя дужкі камандаў мовы HTML. Іх можна набіраць вялікім або малымі лацінскімі літарамі.

Кароткае апісанне асноўных тэгаў дадзена ў Дадатку 1.

Большасць тэгаў парныя. Адкрываючы тэг апісвае каманду і пачынае яе дзеянне. Закрываючы тэг, які паўтарае адкрываючы, але пачынаецца косай рысай / (слэшам), гэта дзеянне спыняе. Напрыклад, тэг **** задае тлустасе напісанне. Размечаны тэкст «Другое **слова******вылучана тлустым шрыфтам» браўзерам будзе адлюстраваны так:

Другое **слова** вылучана тлустым шрыфтом.
Правообладатель Народная асвета

Разгледзім структуру найпрасцейшага HTML-документа.

```
<html>
    <head>
        <title>Загаловак акна</title>
    </head>
    <body>
        Змест дакумента
    </body>
</html>
```

HTML-документ пачынаецца адкрываючым тэгам `<html>`, а заканчваецца — закрываючым `</html>`. Унутры, як у кантэйнеры, размешчаны два блокі.

У блоку `<head>...</head>` (галава) размяшчаецца службовая інфармацыя, якая не адлюстроўваецца на старонцы. Напрыклад, тэкст, змешчаны паміж тэгамі `<title>...</title>`, адлюстроўваецца не на старонцы, а ў загалоўку акна браўзера.

Уся інфармацыя, што адлюстроўваецца браўзерам (тэксты, рэсурсы, відэофрагменты, анімацыі), размяшчаецца ў блоку `<body>...</body>` (цела).

У HTML-дакуменце можна таксама змяшчаць каментарыі, якія не адлюстроўваюцца браўзерам. Яны запісваюцца ў вуглавых дужках з клічнікам `<!Каментары>`.

Прыклад 1. У рэдактары **Блокнот** стварыць HTML-дакумент, які змяшчае тэкст:

Аўтарская старонка

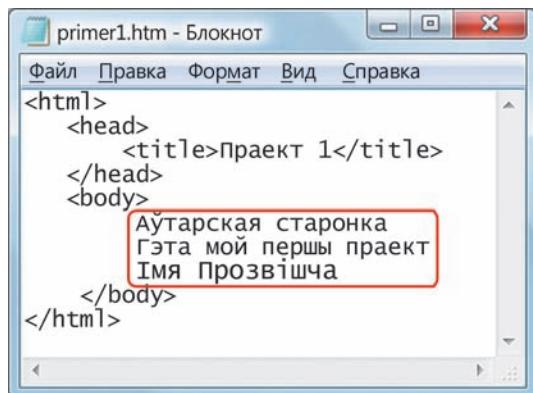
Гэта мой першы праект

Імя Прозвішча

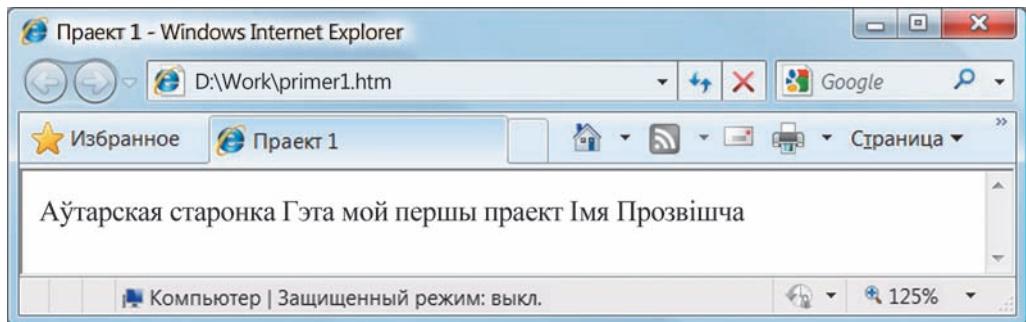
Захаваць створаны дакумент пад іменем `primer1.htm`. Прагледзець яго ў браўзерах.

Адкрыем рэдактар **Блокнот**. Набяром або скапіруем гатовы шаблон HTML-документа з файла `шаблон.htm`. Напоўнім яго зместам, г.зн. увядзём зададзены тэкст, які показана на рэсунку 1.14.

Захаваем дакумент у фармаце HTML. Паколькі створаны ў рэдактары **Блокнот** дакументы па



Рыс. 1.14



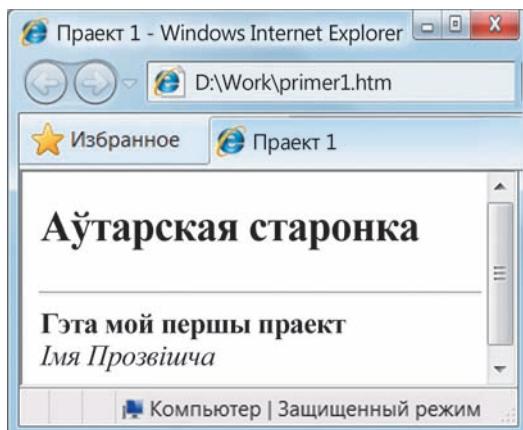
Рыс. 1.15

ўмаўчанні захоўваюца з пашырэннем .txt, выкарыстаем меню **Файл → Сохранить как**, у дыялогавым акне, што адкрыеца, выберам варыянт **Все файлы**, увядзём імя і пашырэнне: primer1.htm.

Адкрыем створаны дакумент у браўзеры (рыс. 1.15). Звернем увагу, што тэкст адлюстроўваеца ў адзін радок, хоць набраны ў трох радкі з водступамі — пераходы на новыя радкі, а таксама ўсе прабелы больш за адзін браўзерам ігнаруюцца.

Такім чынам, выгляд вэб-старонкі ў браўзеры задаецца тэгамі HTML, але таксама залежыць ад тыпу браўзера. У прыведзеных прыкладах выкарыстоўваецца браўзер MS Explorer.

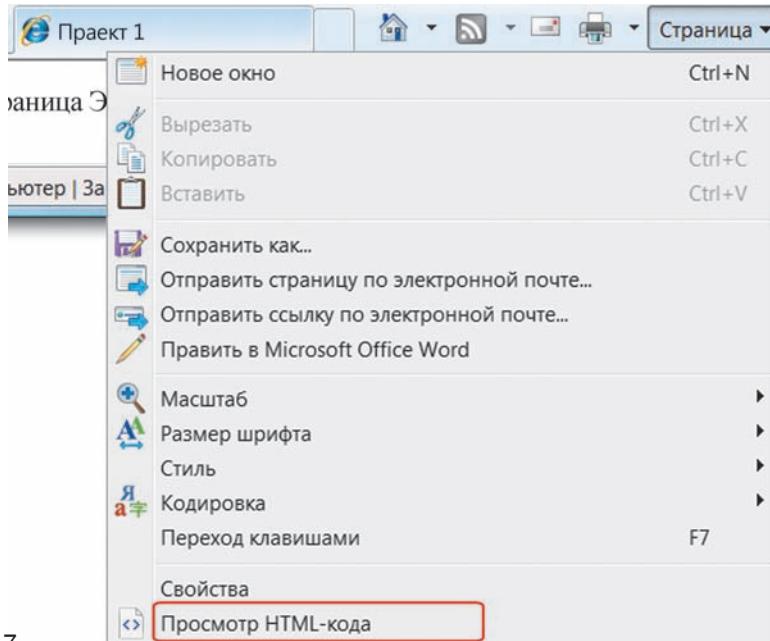
Для адлюстравання тэксту ў выглядзе асобных абзацаў выкарыстоўваюць тэг `<p>`. Пры праглядзе ў браўзеры абзацы аддзяляюцца адзін ад аднаго пустым радком. Для прымусовага пераходу на новы радок без стварэння абзаца выкарыстоўваюць няпарны тэг `
`. Часта паміж абзацамі змяшчаюць раздзяляльную лінію, якая задаецца няпарным тэгам `<hr>`.



Рыс. 1.16

Правообладатель Народная асвета

Прыклад 2. Аформіць створаную ў прыкладзе 1 старонку ў адпаведнасці з рэйсункам 1.16.



Рыс. 1.17

Адрэдагуем HTML-дакумент у рэдактары **Блокнот**. Яго можна адкрыць непасрэдна з браўзера з дапамогай каманды **Страница** (або **Вид**) → **Просмотр HTML-кода** (рыс. 1.17).

Расставім тэгі:

```
<h2>Аўтарская старонка</h2><hr>           <!Загаловак уздоўжно 2>
<b>Гэта мой першы праект</b> <br>      <!Паўтлусты шрыфт>
<i>Імя Прозвішча</i>                      <!Курсіў>
```

Захаваем дакумент у файле з іменем `avtor.htm`. Для прагляду абноўленай старонкі націснем кнопку **Обновить** або клавішу `F5`.

Адкрываючыя тэгі мовы HTML могуць утрымліваць **атрыбуты**, якімі задаюцца параметры разметкі дакумента. Кожны атрыбут мае назыву (імя) і пэўнае значэнне, якое запісваецца ў двукосці пасля знака «роўна». Тэг можа мець некалькі атрыбутаў, якія пералічваюць праз прабелы, напрыклад:

```
<тэг атрыбут1="значэнне" атрыбут2="значэнне"...>.
```

Парарадак размяшчэння атрыбутаў у тэгу можа быць адвольным. Калі значэнне атрыбута змяшчае толькі літары англійскага алфавіта, лічбы або злучок і не мае прабелаў, двукосце можна не ставіць.

Выгляд усёй вэб-сторонкі задаецца атрыбутамі тэга `<body>`, у той час як асобныя яе элементы, напрыклад загалоўкі, раздзелы, абзацы, табліцы, рэсурсы, могуць мець сваё ўласнае афармленне.

Колер фону страницы задаецца атрыбутам `bgcolor`, а колер тэксту — атрыбутам `text`. Значэннем гэтых атрыбутаў з'яўляецца колер, які задаецца сваёй называй на англійскай мове, напрыклад `red` (чырвоны), або яго шаснаццацковым кодам, напрыклад `#FF0000` (Дадатак 2). Па ўмаўчанні колер страницы белы, а колер тэксту — чорны.

Сіні колер страницы можна задаць так: `<body bgcolor="blue">` або так: `<body bgcolor="#0000FF">`.

Заўважым, што колеры, якія адлюстроўваюцца рознымі браўзерамі, а таксама выводзяцца на друк, могуць крыху адрознівацца ад прыведзеных у Дадатку 2.

Для выраўноўвання тэксту ўсёй страницы, асобнага абзака, раздзела або загалоўка выкарыстоўваецца атрыбут `align`, які можа прымаць наступныя значэнні: `center` — выраўноўванне па цэнтры, `left` — па левым краі, `right` — па правым краі. Выраўноўванне тэксту па шырыні выкарыстоўваецца не рэкамендуецца, паколькі пры памяншэнні акна браўзера паміж словамі могуць утварацца вялікія прамежкі.

Для задання шрыфту, колеру і памеру сімвалаў тэксту выкарыстоўваецца тэг ``. Шрыфт задаецца атрыбутам `face`, значэннем якога з'яўляецца назва шрыфту, напрыклад `Arial`. Колер сімвалаў задаецца атрыбутам `color`. Калі шрыфт не зададзены, то па ўмаўчанні выкарыстоўваецца шрыфт `Times` чорнага колеру.

Памер сімвалаў задаецца атрыбутам `size` і можа выражаться ва ўмоўных адзінках, якія могуць прымаць значэнні ад 1 да 6. Па ўмаўчанні прынята значэнне памеру 3.

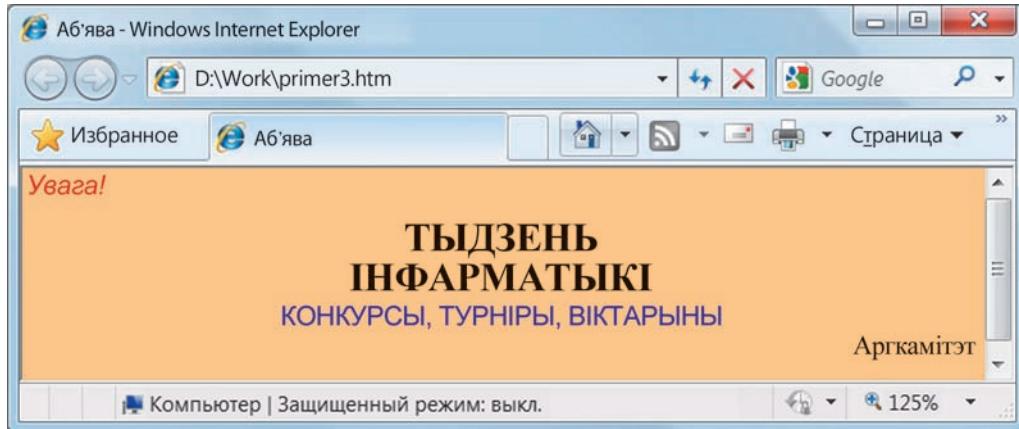
Памер сімвалаў можа таксама выражаться ў адносных адзінках: лічках адносна базавага памеру, напрыклад `size="+n"` або `size="-n"`. Так, для базавага памеру 3 атрыбут `` задае памер 5, а атрыбут `` — памер 1.

Прыклад 3. Стварыць вэб-сторонку ў адпаведнасці з рэсурсам 1.18.

Адкрыем у рэдактары **Блокнот** файл `rgtme3.txt` з тэкстамі аб'явы і набяром HTML-код:

```
<html>
  <head>
    <title>Аб'ява</title>
  </head>
```

Правообладатель Народная асвета



Рыс. 1.18

```

<body bgcolor="gold">
    <font face="Arial" size="+2" color="red"><i>Увага! </i>
    </font>
    <h1 align="center">ТЫДЗЕНЬ<br>ІНФАРМАТЫКІ</h1>
    <font face="Arial" size="+2" color="blue">
        КОНКУРСЫ, ТУРНІРЫ, ВІКТАРЫНЫ</font>
    <p align="right">Аргкамітэт</p>
</body>
</html>

```

Заўважым, што дапускаецца камбінаванне і ўкладзенасць тэгаў. Так, для адлюстравання слова «Увага!» шрыфтам Arial і курсіўным напісаннем у тэг `<font...>` укладзены тэг `<i>`.

Захаваем дакумент у файле з іменем Primer3.htm. Прагледзім яго ў браўзеры.



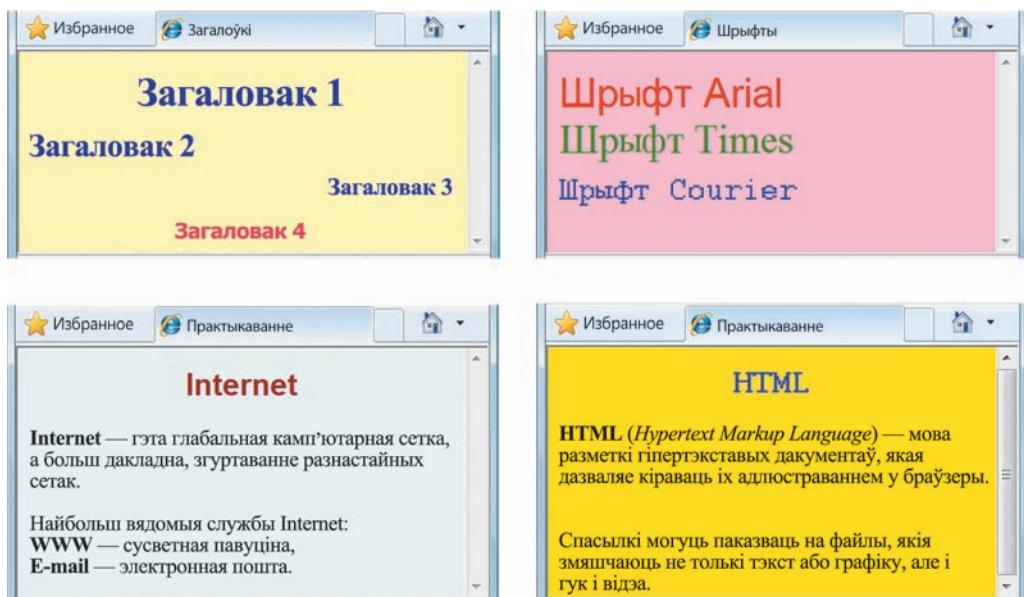
Фрагмент тэкstu можа адлюстроўвацца ў выглядзе бягучага радка з дапамогай тэга `<marquee>`. Напрамак руху задаецца атрибутам `direction`. Напрыклад, рух слова «Увага!» злева направа задаецца так:

```
<marquee direction="right">Увага!</marquee>
```

1. Якім тэгам задаецца абзац? Пераход на новы радок?
2. Якімі тэгамі задаецца тлустае і курсіўнае напісанне сімвалаў?
3. Як задаць фон і колер тэксту ўсёй вэб-старонкі?
4. Якімі тэгамі і атрибутамі задаецца шрыфт, памер і колер сімвалаў?

Практикаванне

Адкрайце прапанаваны настаўнікам тэкставы дакумент у рэдактары **Блокнот**. Аформіце і захавайце яго ў выглядзе вэб-старонкі, паказанай на адным з рэсункаў. Для задання колеру фону выкарыстайце Дадатак 2.



3.2. Відарысы на вэб-старонках

Важную ролю ў афармленні вэб-старонак адыгрываюць графічныя аб'екты: фатаграфіі, рысункі, фонавыя відарысы («шпалеры»), раздзяляльныя лініі. Яны робяць старонкі больш прывабнымі, а ў многіх выпадках з'яўляюцца і асноўнымі крыніцамі інфармацыі.

Прызначаныя для вэб-старонак відарысы можна ствараць і рэдагаваць у розных графічных рэдактарах. Яны могуць быць атрыманы з дапамогай лічбавага фотаапарата, сканера або ўзяты з сеткі Інтэрнэт. Найбольш часта выкарыстоўваюцца графічныя форматы GIF, JPG і PNG, файлы якіх маюць невялікія памеры, што дазваляе паскорыць працэс загрузкі. У фармаце **JPG** захоўваюць паўтонавыя відарысы, напрыклад фатаграфіі. Формат **GIF** падтрымлівае анімацыю і празрысты колер.

Важна падкрэсліць, што ўсе відарысы, якія мы бачым на вэб-старонцы, захоўваюцца ў асобных файлах, а ў HTML-кодзе змешчаны толькі спасылкі на іх.

Для адлюстравання рысункаў прызначаны няпарны тэг ``. Яго неад'емным атрыбутам з'яўляецца імя графічнага файла ``. Вэб-старонкі і файлы з відарысамі могуць захоўвацца ў розных папках, тады ў атрыбуце `src` тэга `` неабходна пазначаць шлях. Указанне шляху з'яўляецца абавязковым для спасылак на ўсе аб'екты, напрыклад на іншыя старонкі, аўдыё- і відэафрагменты.

Тэг `` можа ўтрымліваць і іншыя атрыбуты, якія вызначаюць спосаб адлюстравання рысунка.

Вышыню і шырыню адлюстраванага на экране рысунка ў пікселях можна задаць значэннямі атрыбутаў `height` і `width` адпаведна, а таўшчыню рамкі вакол відарыса — значэннем атрыбута `border`. Па ўмаўчанні рамкі няма, г.зн. `border="0"`. Задаўшы памеры, можна павялічыць або паменшыць адлюстраваны на экране відарыс (пры гэтым зыходны рысунак і файл, у якім ён захоўваецца, застаюцца нязменнымі). Калі памеры не зададзены, то пры загрузцы рысунка можа змяняцца размешчэнне тэксту і іншых аб'ектаў на экране. Каб пазбегнуць гэтага, карысна заўсёды задаваць названыя атрыбуты, нават калі відарыс не ма-штабуецца.

Значэнні атрыбута `align` задаюць выраўноўванне тэксту адносна рамкі рысунка (`top` — па верхнім краі; `middle` — па сярэдзіне; `bottom` — па ніжнім краі) або спосаб абцякання рысунка тэкстам (`left` — рысунак злева ад тэксту; `right` — рысунак справа ад тэксту).

З дапамогай атрыбута `alt` можна задаць тэкст паведамлення, якое будзе выводзіцца замест рысунка, калі ён не знайдзены або калі адключаны яго паказ у браўзеры. Акрамя таго, гэты тэкст з'яўляецца ў выглядзе падказкі пры навядзенні курсора мыши на рысунак.

У якасці фону старонкі можна выкарыстоўваць відарыс з файла. Фонавы рысунак («шпалеры») задаецца з дапамогай атрыбута `background` у тэгу `<body>`. Каб фонавы рысунак, які паўтараецца, запаўняў усю старонку без стыкаў, яго верхняя і ніжняя, левая і правая стороны не павінны адрознівацца. Задаць фонавы рысунак і адначасова зафарбаваць яго некаторым колерам нельга.

Прыклад 1. Змясціць на вэб-старонцы відарысы з файлаў у адпаведнасці з рысункам 1.19.

У рэдактары **Блокнот** адкрыем файл `graf.txt` з шаблонам старонкі.

У тэгу `<body>` пазначым рысунак фону `kletka1.gif`. Расставім тэгі фарматавання тэксту. Уставім тэгі для адлюстравання рысунка ваўка з празрыстым фонам з файла `volk.gif` (без атрыбутаў); анімацыі з файла `tv.gif` (мяжа таўшчынёй 2 пікселі); дзвюю фатаграфій бусла з файла `aist.jpg` розных памераў.

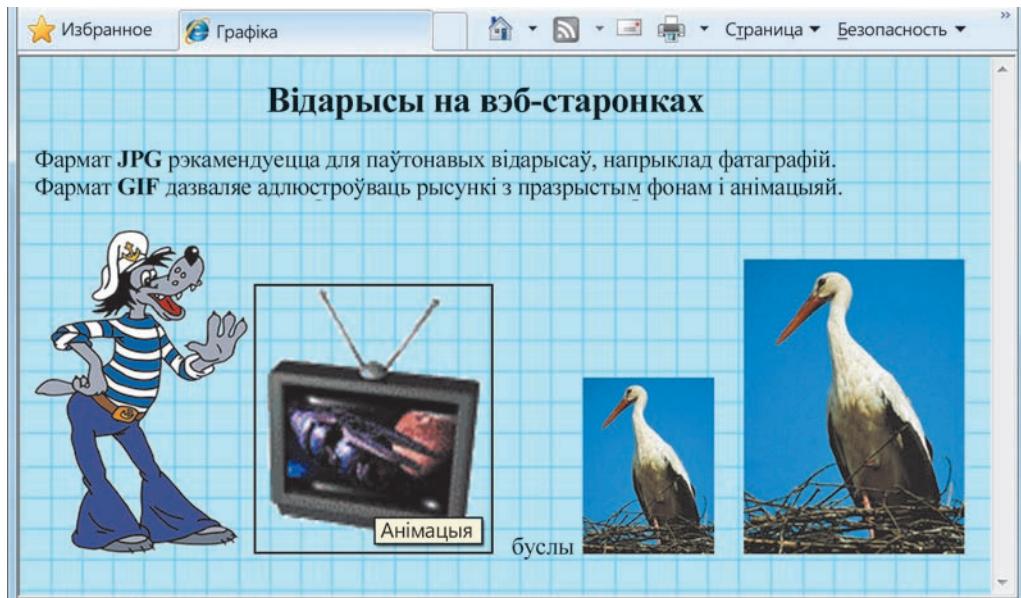


Рис. 1.19

HTML-код вэб-старонкі можа мець наступны выгляд:

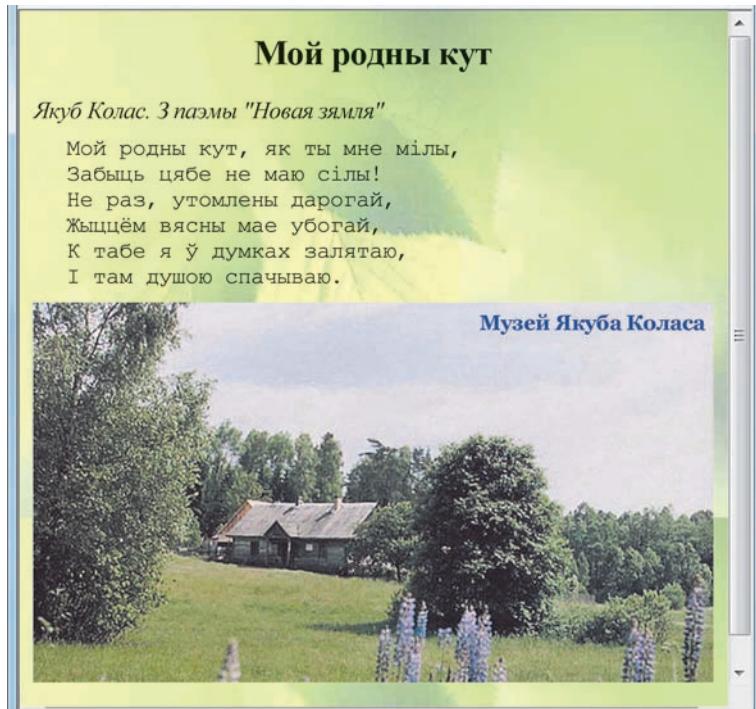
```
<html>
<head><title>Графіка</title></head>
<body background="kletka1.gif">
    <h2 align="center">Відарысы на вэб-старонках</h2>
    Фармат <b>JPG</b> рэкамендуеца для паўтонавых відарысаў, напрыклад фатаграфій.<br>
    Фармат <b>GIF</b> дазвале адлюстроўваць рысункі з празрыстым фонам і анімацыяй.<p>
        
        
        буслы 
        
    </body>
</html>
```

Захаваем HTML-дакумент у файле з іменем graf.htm. Прагледзім яго ў браўзеры. Пры навядзенні курсора на вобласць рысунка tv.gif з'яўляеца падказка «Анімацыя».

Правообладатель Народная асвета

Каб адлюстроўваць тэкст на экране ўтым жа выглядзе, што і ў рэдактары **Блокнот**, выкарыстоўваюць тэг `<pre>`. Гэта дазваляе спрасціц фарматаванне, напрыклад не ўстаўляць тэгі `
` у канцы кожнага радка.

Прыклад 2. Стварыць вэб-сторонку «Мой родны кут» (рыс. 1.20).



Рыс. 1.20

У рэдактары **Блокнот** адкрыем файл `kut.txt` з тэкстам урыўка з паэмы Якуба Коласа «Новая зямля». Дададзім тэгі пачатку і заканчэння вэб-сторонкі. Расставім тэгі фарматавання. Змесцім тэкст верша паміж тэгамі `<pre>...</pre>`.

Пад тэкстам уставім фатографію музея Якуба Коласа з файла `muzkolas.jpg`. У якасці фону выкарыстаем відарыс лісткоў з файла `bg02.jpg`.

HTML-код гэтай вэб-сторонкі можа мець наступны выгляд:

```
<html>
  <head><title>Мой родны кут</title>
  </head>
  <body background="bg02.jpg">
    <h2 align="center">Мой родны кут</h2>
    <i>Якуб Колас. З паэмы "Новая зямля"</i>
    Правообладатель Народная асвета
```

```
<pre>Мой родны кут, як ты мне мілы,  
Забыць цябе не маю сілы!  
Не раз, утомлены дарогай,  
Жыццём вясны мае убогай,  
К табе я ў думках залятаю,  
І там душою спачываю.</pre>  
  
</body>  
</html>
```

Захаваем HTML-документ у файле з іменем kuf.htm.

Ад удалага выбару колеру фону або фонавага рисунка, шрыфту і колеру тэксту, а таксама іншых элементаў афармлення старонкі залежыць успрыманне яе зместу. У Інтэрнэце свабодна распаўсяджаецца вялізная колькасць «шпалер» і гатовых шаблонаў вэб-старонак рознага прызначэння (напрыклад, на сайце «Азбука дизайнера»). Стварэнне ўласных вэб-старонак на іх аснове зводзіцца ў асноўным да ўводу тэксту і спасылак на патрэбныя відарысы.



Вельмі папулярныя, напрыклад, шаблоны віншавальных паштавак і рэкламных старонак. Такі шаблон лёгка распрацаваць самастойна.



Прыклад 3. Стварыць вэб-старонку — шаблон віншавальнай паштоўкі (рыс. 1.21).



Рыс. 1.21

Правообладатель Народная асвета

HTML-код вэб-старонкі можа мець наступны выгляд:

```
<html>
  <head><title>Паштоўка</title>
  </head>
  <body background="bg003b.jpg" text=maroon>
    <font face="courier">
      <h1 align="center">Віншую<br>з днём нараджэння!<br>
      </h1>
    </body>
</html>
```

На аснове гэтага шаблону можна ствараць розныя паштоўкі, змяняючы тэкст віншавання, «шпалеры» і рыsunак.



1. Які тэг адлюстроўвае рыsunак на вэб-старонцы?
2. Як задаюцца памеры відарысаў?
3. Які тэг задае фонавы рыsunак на вэб-старонцы?

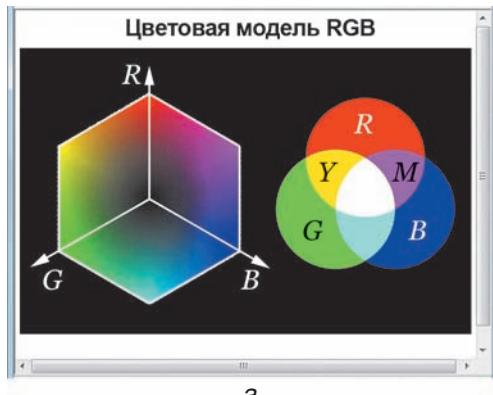
Практыкаванні

1. Змясціце на адной старонцы некалькі відарысаў:
а) розных памераў (напрыклад, 150 × 120, 100 × 80);



- б) у рамках рознай таўшчыні (напрыклад, 0, 1, 3, 8).

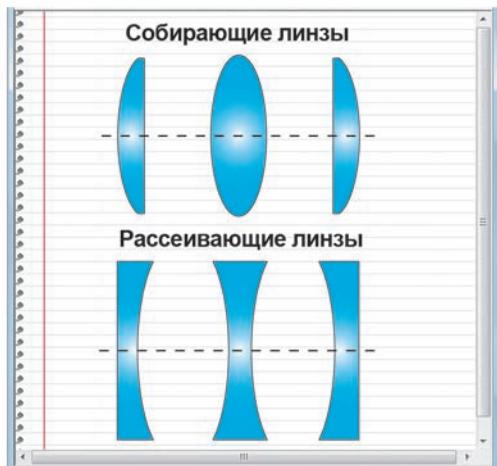


*а**б*

Спадчына
Янка Купала

Ад прадзедаў спакон вякоў
Нам засталася спадчына;
Паміж сваіх і чужакоў
Яна мне ласкай матчынай.

Аб ёй мне баюць казкі-сны
Вясennія праталіны,
І лесу шелест верасны,
І ў полі дуб апалены.

в*г*

2. Адкрыйце прапанаваны настаўнікам тэкставы дакумент у рэдактары **Блокнот**. Аформіце і захавайце яго ў выглядзе вэб-старонкі, паказанай на адным з рисункаў.

3.3. Стварэнне гіперспасылак

Пераходы са старонкі на старонку вэб-сайта выконваюцца з дапамогай гіперспасылак. За арганізацыю спасылак на мове HTML адказвае тэг `<a>` з абавязковым атрибутам `href`, значэннем якога з'яўляецца адрес (URL) рэсурса. Напрыклад, HTML-код тэкставай гіперспасылкі на старонку «Графіка», захаваную ў файле `graf.htm` у той жа папцы, што і зыходная старонка, мае выгляд:

Правообладатель Народная асвета

`Графіка`,
г. зн. адрасам рэсурса з'яўляеца імя
файла.

Звычайна блізкія па тэматыцы страниці сайта захоўваюць у адпаведных папках. Няхай, напрыклад, на дыску D: у папцы kino захоўваюць галоўную страницу main.htm, ва ўкладзенай папцы drama — страницу фільма drama2.htm, а ў папцы multy — страницы karlson.htm і ну погоди.htm (рыс. 1.22). У гэтым выпадку пры стварэнні гіперспасылак неабходна пазначаць шляхі да файлаў.

Магчымы два варыянты:

- запісаць *поўны шлях* ад кораня дыска да выклікаемай страниці;
- пазначыць *адносны адрас* (шлях ад зыходнай страниці да выклікаемай). Напрыклад, спасылка са страницы main.htm на страницу karlson.htm будзе такой:

```
<a href="multy/karlson.htm">Карлсон</a>.
```

Спасылка са страницы drama2.htm (папка drama) на страницу karlson.htm (папка multy) будзе мець выгляд:

```
<a href="../multy/karlson.htm">Карлсон</a>.
```

Сімвалы `..` (дзве кропкі і слэш) абазначаюць вяртанне ў папку kino з укладзенай папкай drama.

Пры пераносе папкі kino з усім ўкладаннямі ў іншае месца, у тым ліку на іншы дыск або камп'ютар, адносныя адресы не змяняюцца!

Гіперспасылкай можа быць не толькі тэкст, але і рысунак. Для стварэння такой гіперспасылкі паміж парай тэгаў `...` трэба ўставіць тэг, які паказвае на файл відарыса, напрыклад:

```
<a href="graf.htm"></a>.
```

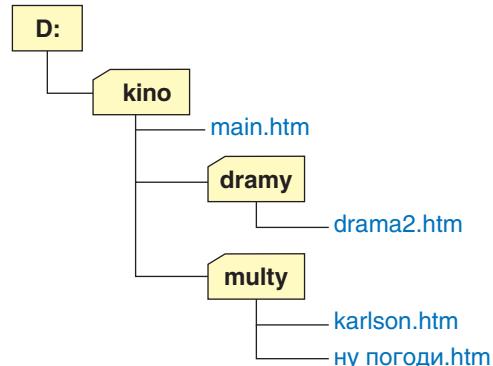
Магчыма сумеснае выкарыстанне тэксту і графікі ў адной спасылцы:

```
<a href="graf.htm">Графіка</a>.
```

Па ўмаўчанні выкліканая гіперспасылкай страница адкрываеца ў тым жа акне, што і зыходная, замяняючы яе. Назад можна вярнуцца з дапамогай адпаведнай кнопкі на панэлі інструментатаў браўзера. Для прагляду выкліканай страницы ў новым акне неабходна пазначыць атрибут `target="_blank"`, напрыклад:

```
<a href="graf.htm" target="_blank">Графіка</a>.
```

Правообладатель Народная асвета



Рыс. 1.22

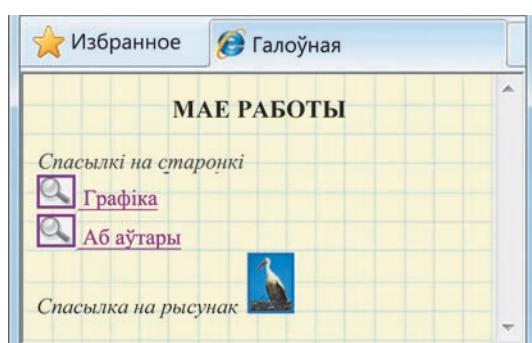
Гіперспасылка можа паказваць не толькі на вэб-старонку, але і на любы змешчаны ў сетцы па вядомым адресе інфармацыйны рэсурс: рысунак, тэкст, презэнтацыю, аўдыёвідэафрагмент. Вядома, на камп'ютары карыстальніка павінна быць усталявана адпаведнае праграмнае забеспечэнне для яго адлюстравання.

Няхай, напрыклад, рэферат па астрономії mesiac.doc змешчаны ў папцы astro на сایце www.referaty.by. Гіперспасылка для яго прагляду або капіравання можа быць такой:

```
<a href="www.referaty.by/astro/mesiac.doc">Месяц</a>.
```

Яшчэ раз падкрэслім, што расстаўляць гіперспасылкі трэба толькі пасля стварэння рэсурсаў, на якія яны паказваюць, і размяшчэння іх файлаў у зададзеных папках.

Прыклад 1. Стварыць галоўную старонку вэб-сайта «Мае работы» (рыс. 1.23). Змясціць на ёй гіперспасылкі на старонкі «Графіка», «Аб аўтары» (адкрываюцца ў tym жа акне), а таксама на рысунак (адкрываецца ў новым акне).



Рыс. 1.23

Няхай фрагмент сайта будзе змешчаны ў папцы PRO. Скаліруем у гэту папку створаныя загадзя файлы вэб-старонак «Графіка» graf.htm і «Аб аўтары» avtor.htm разам з усімі выкарыстанымі рысункамі.

Аформім галоўную старонку ў адпаведнасці з рысункам 1.23. Расставім гіперспасылкі. HTML-код гэтай вэб-старонкі можа мець наступны выгляд:

```
<html>
  <head><title>Галоўная</title></head>
  <body background="kletka2.gif">
    <h3 align="center">МАЕ РАБОТЫ</h3>
    <i>Спасылкі на старонкі</i><br>
    <a href="graf.htm">Графіка</a><br>
    <a href="avtor.htm">Аб аўтары</a><br>
    <i>Спасылка на рысунак</i>
    <a href="aist.jpg" target="_blank">
      </a>
  </body>
</html>
```

Захаваем HTML-дакумент у файле з іменем index.htm.

Праверым работу гіперспасылак. Старонкі «Графіка» і «Аб аўтары» адкрывающа ў бягучым акне пасля петрычкі мышшу па тэксце спасылкі або па карцінцы з сімвалічным відарысам лупы (файл icon.gif). Рысунак (файл aist.jpg) адкрываеца ў новым акне пасля петрычкі мышшу па яго паменшаным відарысе (з таго ж файла aist.jpg).

Заўважым, што для паскарэння загрузкі старонкі з вялікай колькасцю графічных спасылак трэба выкарыстоўваць мініяцюрныя відарысы з асобных файлаў.

У разгледжаным прыкладзе ўсе файлы захоўвающа ў адной папцы. У выпадку вялікай колькасці старонак або відарысаў іх трэба змяшчаць ва ўкладзеных папках і пазначаць шляхі да іх.



Гіперспасылкі могуць паказваць не толькі на асобныя дакументы, але і на зададзеная месцы ў дакуменце. Спасылкі ўнутры доўгіх старонак павышаюць зручнасць іх прагляду. Яны стварающа ў два этапы:

1) спачатку ў месцы, куды будзе ажыццяўляцца пераход па спасылцы, устаўляецца закладка, напрыклад:

```
<a name="Закладка1">...</a>;
```

2) затым ствараеца спасылка на гэту закладку:

```
<a href="#Закладка1"> тэкст гіперспасылкі </a>.
```

Для спасылкі на адрес электронной пошты яго трэба пазначыць у якасці URL, напрыклад

```
<a href="mailto:sova@les.by">ліст саве</a>.
```

Як ужо адзначалася, у блоку `<head>` могуць змяшчацца тэгі, якія ўтрымліваюць службовую інфармацыю для браўзера і пошукавых сістэм. Яны называюцца **мета-тэгамі**. Большасць мета-тэгаў з'яўляюцца неабязважковымі. Аднак некаторыя з іх вельмі карысныя.

Так, браўзер будзе адлюстроўваць тэкст старонкі на нацыянальнай мове, калі ў мета-тэгу `charset` пазначыць табліцу кадзіроўкі. Напрыклад, для кірыліцы мета-тэг будзе такі:

```
<meta http-equiv="content-type" content="text/html;
charset=Windows-1251">
```

Для захоўвання інфармацыі аб аўтары выкарыстоўваюць мета-тэг

```
<meta name="author" content="Алесь Палескі">.
```

Правообладатель Народная асвета



Па словах, пералічаных праз коску або прабел у мета-тэгу "keywords" (ключавыя слова), пошукавыя сістэмы індэксуюць вэб-старонкі, г. зн. змяшчаюць інфармацыю аб іх у сваю базу даных, каб потым аператывна выдаваць па запытах карыстальнікаў. Даўжыня радка ключавых слоў не можа перавышаць 800 сімвалаў. Словы ў ім не павінны паўтарацца, напрыклад:

```
<meta name="keywords" content="Адукацыя Мультымедыа  
Multimedia Графіка Відэа">.
```



1. З дапамогай якога тэга ствараецца гіперспасылка?
2. Як у гіперспасылках выкарыстоўваюцца відарысы?

Практыкаванне

Стварыце фрагмент вэб-сайта з 3-4 вэб-старонак па адной з тэм: Дом, у якім я жыву. Мой клас. Мая школа. Любімая вершы (песні, кнігі, фільмы). Любімая паэты (пісьменнікі, мастакі, артысты). Любімая справы (прыклад галоўнай старонкі паказаны на рисунку).

ЛЮБІМЫЯ СПРАВЫ



Спорт



Камп'ютар



Рыбалка

§ 4. Падрыхтоўка відарысаў для Інтэрнэта

Пры стварэнні вэб-сайта трэба клапаціца аб tym, каб яго старонкі загружаліся як мага хутчэй. Гэта асабліва актуальна, калі на старонках сайта неабходна змясціць шмат відарысаў. Таму такое вялікае значэнне мае граматны выбар фармату іх захоўвання.

Як Вы ведаецце, нарысаваныя ў графічным рэдактары Paint карцінкі па ўмаўчанні захоўваюцца ў фармаце BMP (*Bit Map Picture* — бітавая карта відарыса). У гэтым фармаце відарысы захоўваюцца ў файлах у незапакаваным выглядзе. Інфармацыйны аб'ём відарыса ў бітах вызначаюць множаннем яго шырыні **W** (width) і вышыні **H** (height) у пікселях на глыбіню колеру **C** у бітах на 1 піксель (**W · H · C**). Так, файл з невялікай карцінкай памерам 100×100 пікселяў пры глыбіні колеру 24 біт/піксель будзе мець аб'ём каля 29 Кбайт. Аб'ём файла з відарысам, памер якога 800×600 пікселяў, складзе каля 1,4 Мбайт, а час яго загрузкі пры скорасці мадэма 56 кбіт/с будзе большы за 200 с, што зусім непрымальнна.

У большасці графічных фарматуў відарысы захоўваюцца ў файлах у запакаваным выглядзе. Існуюць розныя спосабы памяншэння памеру файла відарыса з мінімальнымі стратамі якасці. Гэты працэс называюць **аптымізацыяй**.

Пры вялікай разнастайнасці графічных фарматуў для размяшчэння на сайтах у асноўным выкарыстоўваюцца відарысы фарматуў GIF, JPG і PNG.

Выбар фармату захоўвання відарыса залежыць ад уласцівасцей відарыса. Так, фармат **GIF** выкарыстоўваюць для відарысаў з выразнымі лініямі, аднароднымі заляўкамі, тэкстам, напрыклад чарцяжоў, карт. Такія відарысы называюць *штырхавымі*. Гэты фармат падтрымлівае палітру адценняў, якая складаецца больш чым з 256 колераў, і дазваляе задаваць празрысты фон. Аптымізацыя заключаецца ў памяншэнні колькасці выбранных колераў. Важней вартасцю фармату GIF з'яўляецца магчымасць анімацыі. Таму гэты фармат шырока выкарыстоўваецца для размяшчэння на вэб-старонках простых графічных элементаў: сімвалаў, формул, лагатыпаў, рэкламных ролікаў.

Фармат **JPG** дазваляе перадаваць багатую палітру колераў і лепш за ўсё падыходзіць для відарысаў з плаўным пераходам тонаў без выразных ліній і контураў, напрыклад партрэтай, пейзажаў. Такія відарысы называюць *паўтонавымі*. Аптымізацыя заключаецца ў замене абласцей з невялікім градыентам колеру на аднатонныя. Ступень сціскання харектарызуецца паказчыкам якасці ад 0 да 100. Часта па ўмаўчанні яго прымаюць роўным 65, што адпавядае сцісканню зыходнага незапакаванага відарыса ў 6—10 разоў. Захаванне відарысаў у фармаце JPG прадугледжана практычна ва ўсіх сучасных лічбавых фотапаратах.

Пры падрыхтоўцы відарысаў для Інтэрнэта выпрацаваны пэўныя рэкамендацыі. Памеры захаваных у файлах відарысаў павінны адпавядаць памерам відарысаў, што адлюстроўваюцца на вэб-старонках, нягледзечы на тое, што іх шырыню і вышыню можна задаваць у тэгу ``. Напрыклад, для манітораў $17'$ з адразнівам 1280×768 пікселяў шырыня відарыса на старонцы часцей за ёсё не перавышае $600—800$ пікселяў, а вышыня — $400—600$. Калі відарыс абцякаеца тэкстам, то яго памер павінен знаходзіцца ў межах $200—400$ пікселяў.

Працэс аптымізацыі відарыса зводзіцца да дзвюх асноўных апераций:

- прывядзенне відарыса да патрэбнага памеру;
- сцісканне відарыса с захаваннем аптымальнай якасці.

Прыклад 1. Фатаграфія памерам 2560×1920 пікселяў захавана ў файле аб'ёмам $2,4$ Мбайт. Яе шырыню і вышыню паменшылі ў 4 разы. Вызначыць аб'ём файла і памеры відарыса.

Памер відарыса стане роўным 640×480 пікселяў. Пры гэтым аб'ём файла паменшыцца ў $4 \cdot 4 = 16$ разоў і складзе $0,15$ Мбайт. У 16 разоў паменшыцца і час загрузкі гэтага відарыса!

Задзейнім, што змяніць памеры відарысаў і захоўваць іх у фарматах GIF або JPG можна практычна ў любым растральным графічным рэдактары, у тым ліку ў Paint. Для сціскання відарысаў патрабуецца больш складаны рэдактар.

Разгледзім аптымізацыю відарысаў у графічным рэдактары Paint.Net, які вольна распаўсюджваецца ў Інтэрнэце (<http://paintnet.ru>).

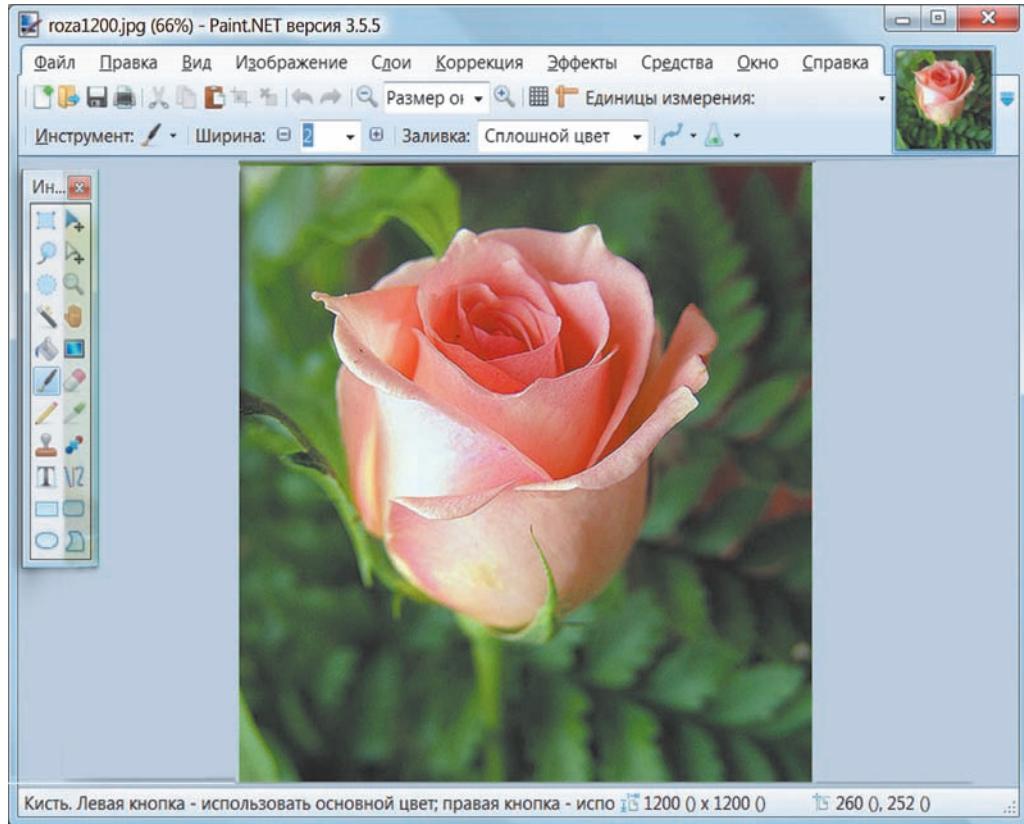
Прыклад 2. Фатаграфія памерам 1200×1200 пікселяў захавана ў файле goza1200.jpg аб'ёмам 745 Кбайт. Аптымізаваць відарыс так, каб на вэб-старонцы яго памер склаў 300×300 пікселяў.

Адкрыем у рэдактары Paint.Net відарыс, захаваны ў файле goza1200.jpg (рыс. 1.24).

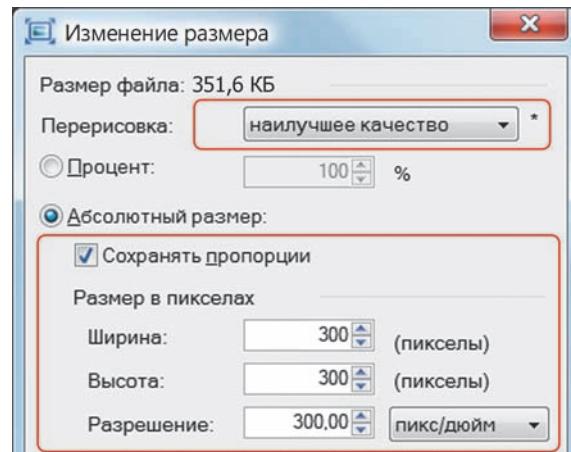
Перш за ёсё паменшым памер відарыса. Для гэтага з меню **Изображение** выклічам акно **Изменение размера** і ўстановім значэнне **Ширина:** 300 пікселяў (рыс. 1.25). Пры ўключаным флагку **Сохранять пропорции** ў такой жа пропорцыі зменіцца і вышыня.

Захаваем паменшаны відарыс у фармаце JPG. Для гэтага з дапамогай меню **Файл** выклічам дыялогавае акно **Сохранить как**, выберам тып файла JPEG і набяром новае імя файла, напрыклад goza300.jpg. Пасля націскання кнопкі **Сохранить** з'явіцца дыялогавае акно настройкі якасці відарыса **Параметры сохранения**. Па ўмаўчанні паказчык **Качество** нашага відарыса роўны 100 . Аб'ём файла складае прыкладна 83 Кбайт.

Працягнем аптымізацыю шляхам сціскання відарыса с захаваннем прымальнай якасці. З дапамогай бягунка будзем памяншаць паказчык **Качество** да з'яўлення на аптымізаваным відарысе бачных скажэнняў. Так, пры паказчыку 50



Рыс. 1.24



Рыс. 1.25

Правообладатель Народная асвета

бачных зміненняў яшчэ няма, а аб'ём файла пры гэтым складае каля 19 Кбайт (рыс. 1.26). Спынімся на гэтым значэнні, націснуўшы кнопкі OK. У выніку якасць відарыса пагорышлася нязначна, а аб'ём файла з аптымізаваным відарысам стаў амаль у 40 разоў меншы за аб'ём зыходнага.

 Значна больш шырокія магчымасці аптымізацыі відарысаў прадстаўляе графічны рэдактар Adobe Photoshop.

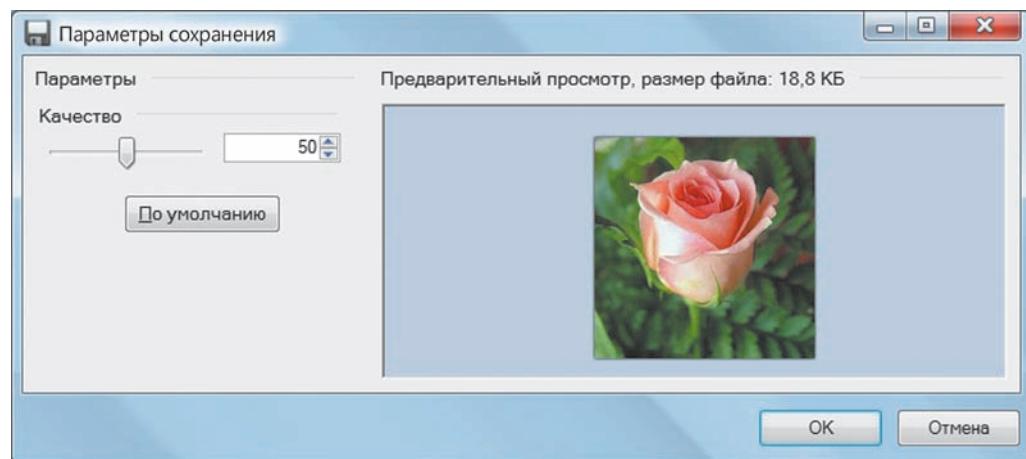


Рис. 1.26

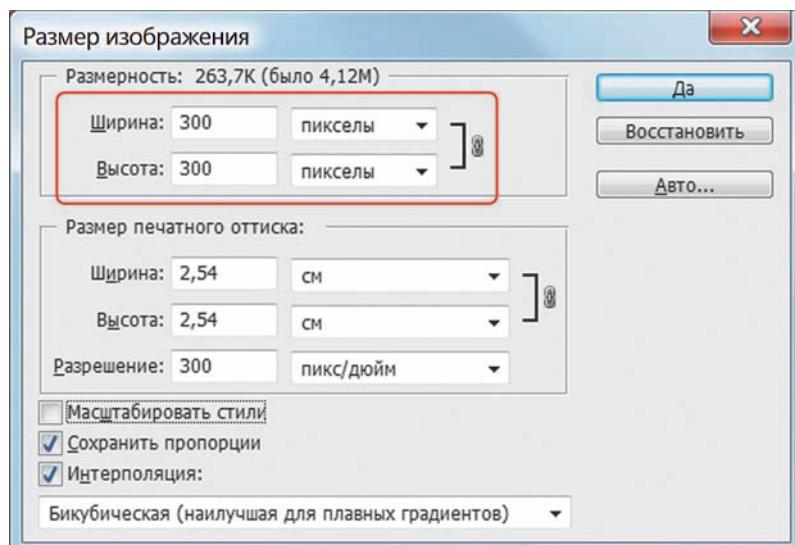


Рис. 1.27

Правообладатель Народная асвета

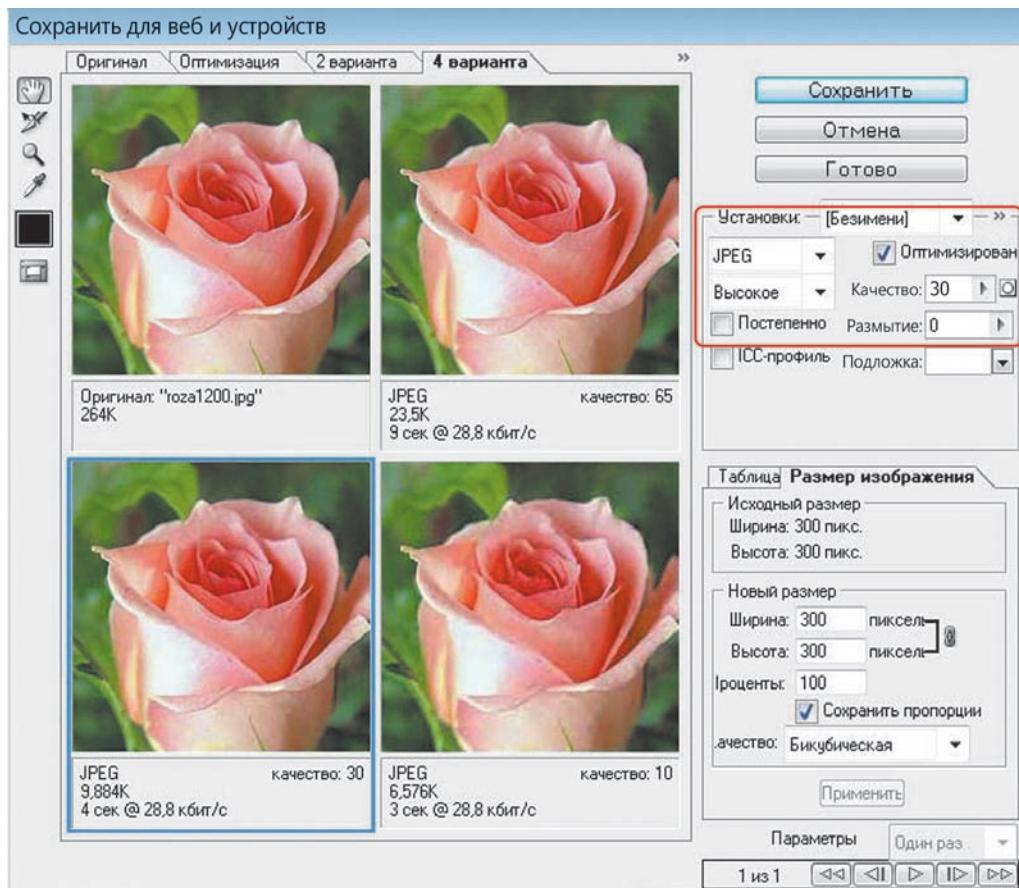


Прыклад 3. Аптымізація відарыс, які захоўваецца ў файле roza1200.jpg так, каб на вэб-сторонцы яго памер склаў 300×300 пікселяў.

Адкрыем у рэдактары Adobe Photoshop відарыс з файла roza1200.jpg.

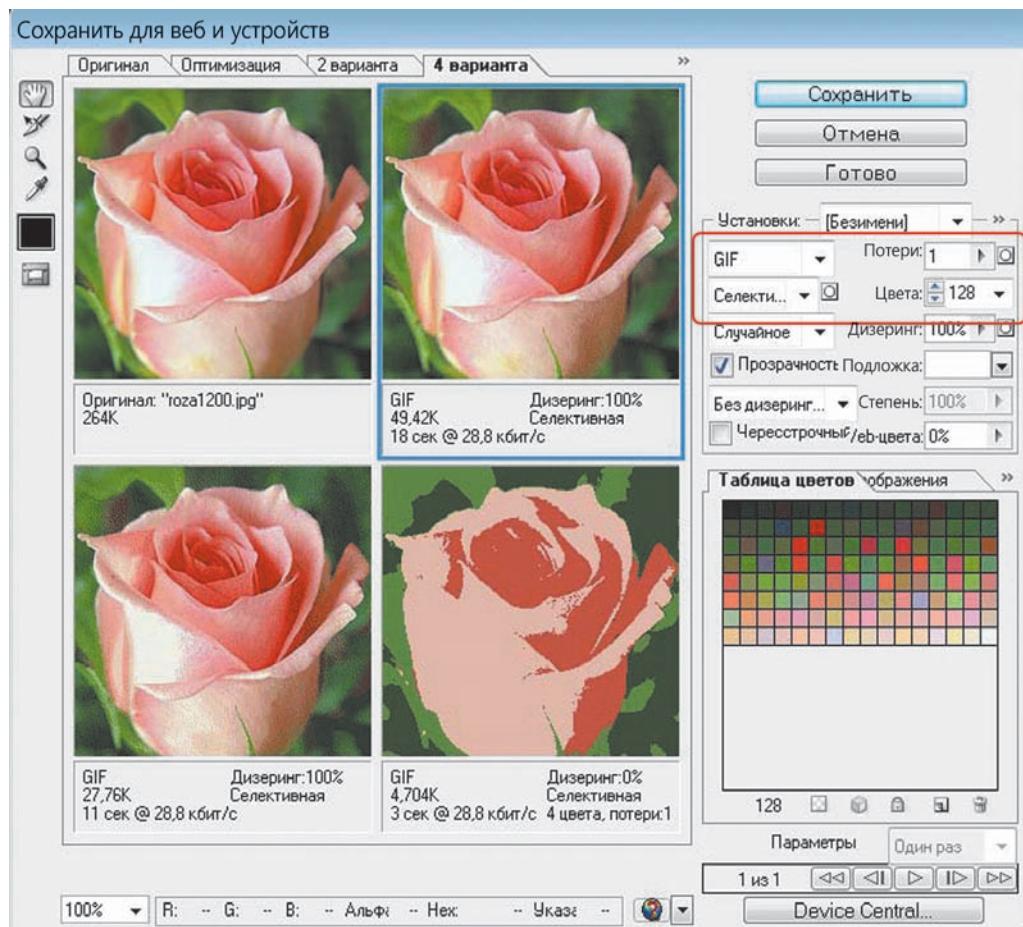
Для змянення памеру відарыса з меню **Ізображение (Image)** выклічам акно **Размер изображения (Image Size)** і ўстановім значэнне **Ширина (Width)** 300 пікселяў (рыс. 1.27). Пры ўключаным флажку **Сохранить пропорции** ў такой жа пропорцыі зменіцца і вышыня.

З меню **Файл (File)** выклічам акно **Сохранить для веб и устройств (Save for Web)** і з дапамогай укладкі **4 варианта** атрымаем чатыры відарысы: зыходны (злева ўверсе) і аптымізаваныя з якасцю 65% (справа ўверсе), 30% і 10% (унізе) (рыс. 1.28).



Рыс. 1.28

Правообладатель Народная асвета



Рыс. 1.29

Будзем памяншаць паказчык **Качество** да з'яўлення бачных змяненняў на аптымізаваным відарысе. Гэта можна зрабіць груба, выбраўшы ў полі **Установки (Setting)** адзін з паказчыкаў якасці (**Максимальное, Высокое, Среднее, Низкое**), і дакладна, задаўшы прыдатнае лікавае значэнне. Спынімся, напрыклад, на значэнні 30 і націснем кнопку **Сохранить (Save)**.

Аб'ём аптымізаванага файла не перавышае 10 Кбайт, т. з.н. ён паменшыўся амаль у 75 разоў. Не забудзем увесці новае імя файла, напрыклад roza300-30.jpg.

Аналагічным спосабам аптымізуець відарысы ў фармаце GIF. У гэтым выпадку памяншаюць колькасць колераў ад 256 да значэння, пры якім з'яўлена

ляюцца бачныя скажэнні. На рысунку 1.29 паказаны чатыры відaryсы: зыходны (злева ўверсе) і аптымізаваныя з колькасцю колераў 128 (справа ўверсе), 32 (злева ўнізе, страты якасці невялікія) і 4 (справа ўнізе, якасць непрымальная).



1. Чаму неабходна памяншаць аб'ёмы файлаў з відaryсамі для вэб-старонак?
2. У якіх фарматах рэкамендуецца захоўваць відaryсы для вэб-старонак?
3. У чым заключаецца аптымізацыя відaryсаў для Інтэрнэта?

Практыкаванні

1. Аптымізуйце відaryсы, захаваныя ў прапанаваных настаўнікам файлах.
2. З дапамогай клавішы Print Screen змясціце ў буфер абмену відaryс Ва-шага экрана, напрыклад Рабочы стол. Адкрыйце рэдактар Paint, з дапамогай камбінацыі клавіш Ctrl + V устаўце відaryс з буфера і захавайце пад іменем ris.bmp. Захавайце гэты ж відaryс у фарматах JPG і GIF. Параўнайце аб'ёмы файлаў.

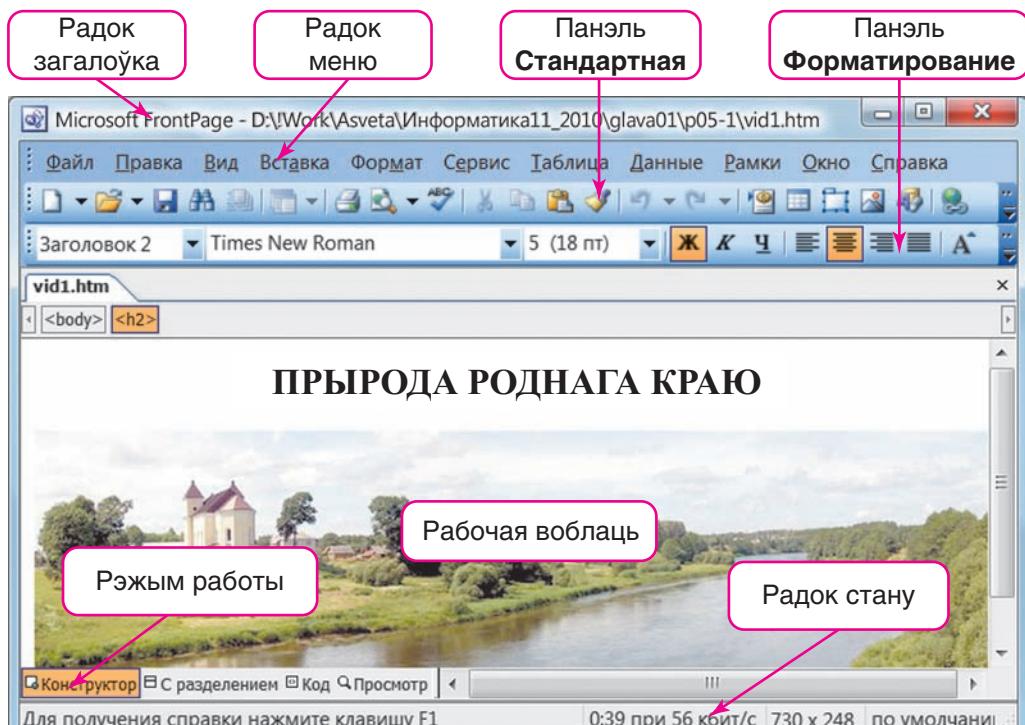
§ 5. Вэб-канструяванне ў рэдактары FrontPage

5.1. Асноўныя элементы інтэрфейса

Вы ўжо навучыліся ствараць вэб-старонкі ў праграмах Microsoft Office, а таксама з выкарыстаннем мовы разметкі HTML у найпрацэснейшым тэкстовым рэдактары **Блокнот**. Разгледзім цяпер магчымасці спецыяльнага вэб-рэдактара Microsoft FrontPage, які прызначаны для распрацоўкі вэб-сайтаў і адносіцца да праграмных сродкаў візуальнага вэб-канструявання. Пры работе з вэб-рэдактаром FrontPage можна абысціся без ведання мовы разметкі гіпертэкстовых дакументаў HTML. Вэб-старонка проста канструюецца на экране і захоўваецца ў фармаце HTML. Пры афармленні вэб-документа ў FrontPage выконваюцца дзеянні, якія нагадваюць работу ў тэкстовым рэдактары Microsoft Word.

Інтэрфейс вэб-рэдактара FrontPage дастаткова прости, каб карыстальнік мог хутка засвоіць асноўныя прыёмы работы. Пасля запуску FrontPage адкрываецца акно, асноўныя элементы якога прадстаўлены на рысунку 1.30.

Меню і панэлі інструментаў **Стандартная**, **Форматирование**, **Рисование** па сваіх магчымасцях і прыёмах выкарыстання нагадваюць аналагічныя панэлі тэкставага рэдактара Microsoft Word.



Рыс. 1.30

Выгляд **Рабочай вобласці** залежыць ад выбранага рэжыму работы.

У рэжыме **Конструктор** вэб-старонка будзеца з тэкставых блокаў і графічных аб'ектаў. Пры гэтым аўтаматычна генерыруецца яе HTML-код, які можна праглядаць і рэдагаваць у рэжыме **Код**.

Рэжым **С разделением** (рыс. 1.31) з'яўляецца камбінацыяй гэтых двух рэжымаў.

Рэжым **Просмотр** дазваляе праглядаць створаныя старонкі.

З дапамогай меню **Вид** можна выбраць і іншыя карысныя рэжымы работы.

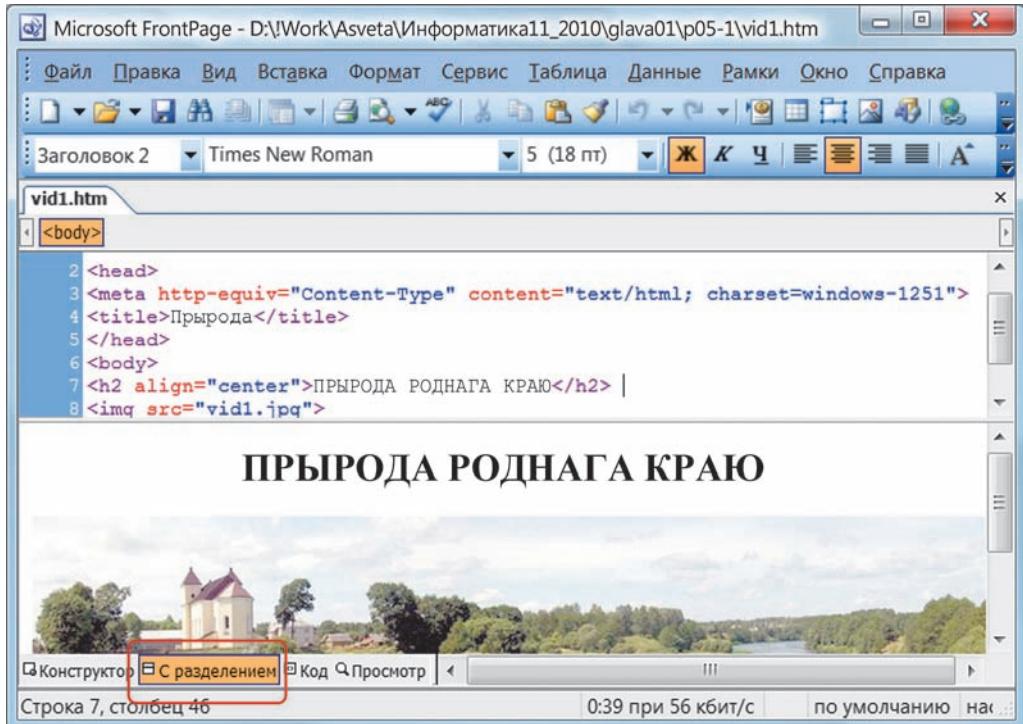
Рэжым **Страница** прызначаны для стварэння і рэдагавання вэб-старонак.

У рэжыме **Папкі** праглядаецца структура папак сайта, выконваюцца файлавыя аперацыі.

Рэжым **Переходы** адлюстроўвае структуру сувязей паміж старонкамі і дазваляе яе змяніць.

Рэжым **Гипэрсылкі** наглядна адлюстроўвае сістэму спасылак, забяспечвае іх праверку і рэдагаванне.

Правообладатель Народная асвета



Рыс. 1.31

5.2. Работа ў рэдактары FrontPage

Разгледзім асноўныя прыёмы работы ў рэдактары FrontPage на прыкладзе распрацоўкі вэб-сторонкі «Озера Беларусі». Яна будзе змяшчаць тэкст і фатографіі, якія трэба падрыхтаваць загадзя.

Пачнём са стварэння і аформлення тэкставага дакумента.

Прыклад 1. Стварыць вэб-сторонку «Озера Беларусі» (рыс. 1.32).

Запусцім рэдактар FrontPage. Адкрыеца пустая старонка з устаноўленымі па ўмаўчанні параметрамі.

Перад пачаткам работы карысна праверыць настройкі рэдактара, і, перш за ёсё, кадзіроўку. Для гэтага з дапамогай каманды **Файл → Свойства** адкрыем акно **Свойства страницы** і на ўкладцы **Язык** установім **кириллица** (рыс. 1.33).

Будзем працаўваць у рэжыме **Конструктор**. Аднак пры неабходнасці можам праглядаць і рэдагаваць аўтаматычна згенерыраваны HTML-код у рэжымах **Код** або **С разделением**.

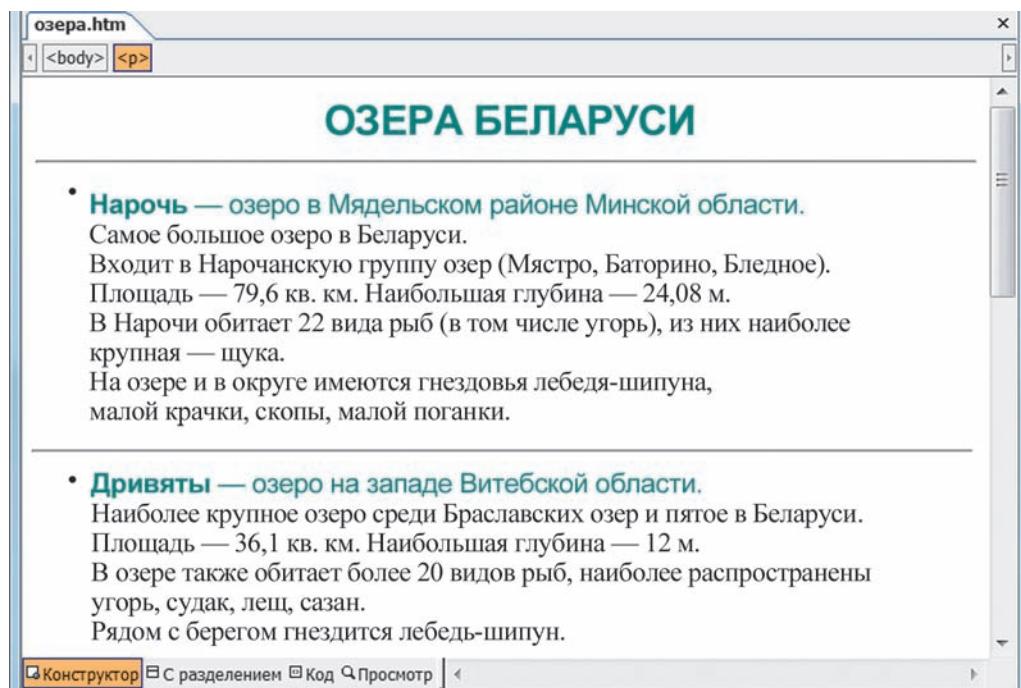


Рис. 1.32

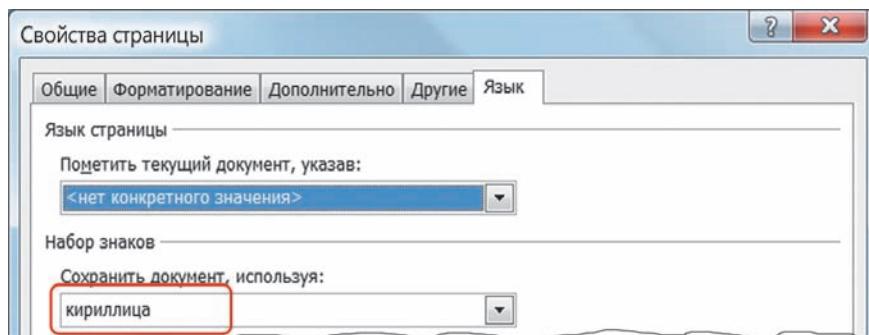
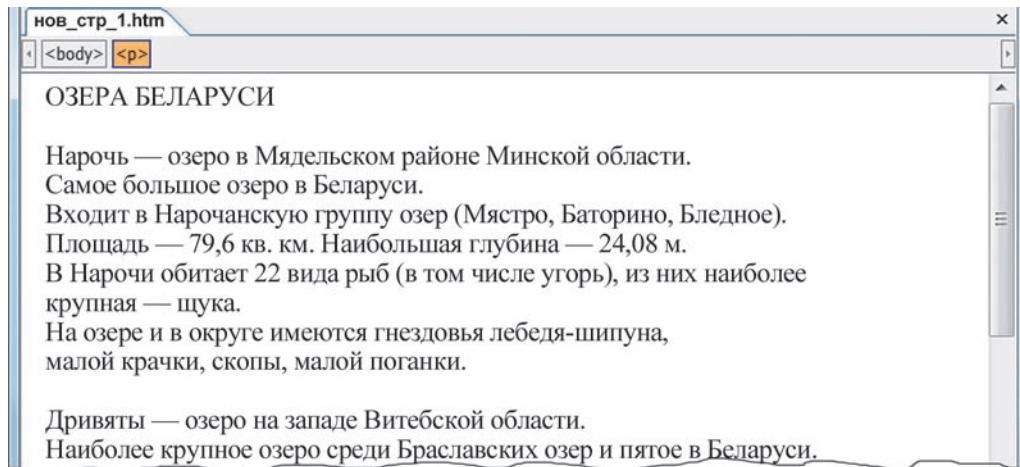


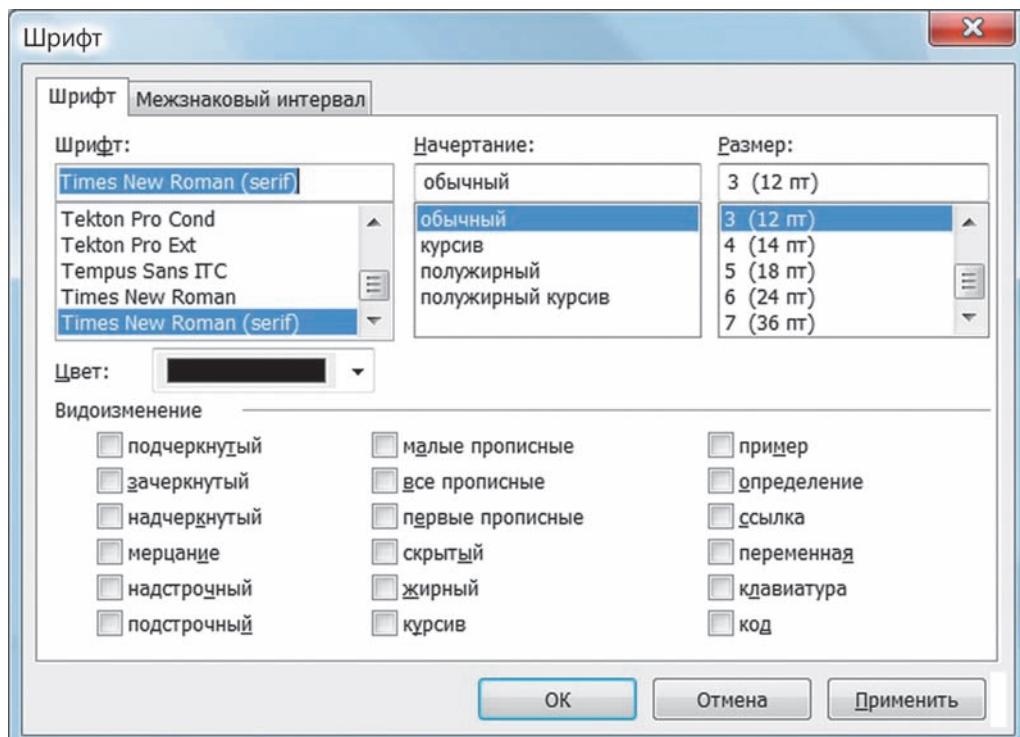
Рис. 1.33

Пачнём з размяшчэння і афармлення тэксту, які быў загадзя захаваны ў файле озера.txt. Адкрыем яго ў рэдактары **Блокнот**, скапіруем патрэбныя фрагменты і з дапамогай буфера абмену ўставім на ствараемую старонку. Тэкст будзе аформлены прынятym па ўмаўчанні стылем: шрыфт Times, памер 3 (12 пт), выраўноўванне па левым краі (рыс. 1.34).

Правообладатель Народная асвета



Рыс. 1.34



Рыс. 1.35

Правообладатель Народная асвета

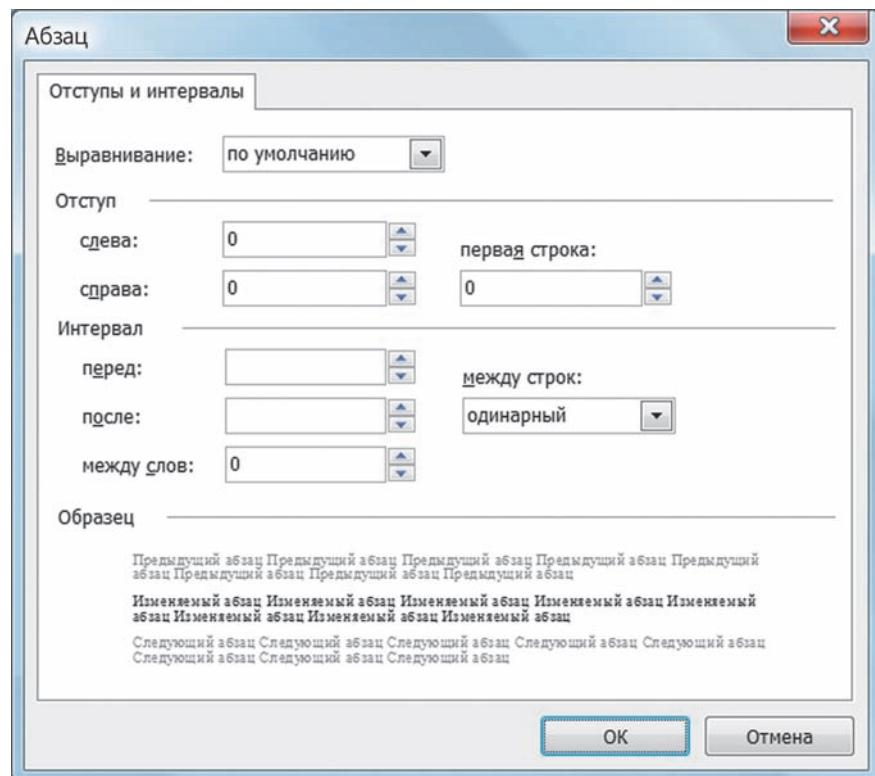


Рис. 1.36

Важна памятаць, што адлюстраванне вэб-старонкі на экране залежыць ад параметраў манітора і настройкі браўзера. Таму памер шрыфту прынята задаваць ва ўмоўных адзінках ад 1 да 7. Калі памер шрыфту не пазначаны, то па ўмаўчанні ён прымаецца роўным 3, што пры ўстаноўцы ў браўзера значэння **Размер шрифта** → **средний** адпавядзе 12 пунктам.

Для фарматавання тэксту выкарыстаем панэль інструментаў **Форматирование**, а таксама вокны **Шрифт** (рыс. 1.35) і **Абзац** (рыс. 1.36) з меню **Формат**.

Установім параметры загалоўка: шрыфт Arial, памер 5 (18 pt), колер бірузовы, выраўноўванне па цэнтры. Параметры асноўнага тэксту пакінем без змяненняў, толькі вылучым першыя радкі кожнага абзака (шрыфт Arial, колер бірузовы). Назвы азёраў вылучым паўгустым напісаннем.

Павышэнню зручнасці чытання тэксту і прывабнасці старонак спрыяе выкарыстанне спіскаў, а таксама падзел вялікіх старонак на асобныя блокі з дапамогай гарызантальных раздзяляльных ліній.

Маркіраваныя і нумараваныя спіскі ствараюцца з дапамогай панэлі інструментов **Правообладатель Народная асьвета**

ментаў **Форматирование**, а таксама дыялогавага акна **Список**, якое адкрываецца з меню **Формат**. Для афармлення вэб-сторонкі «Озера Беларуси» выкарыстаем графічныя маркёры.

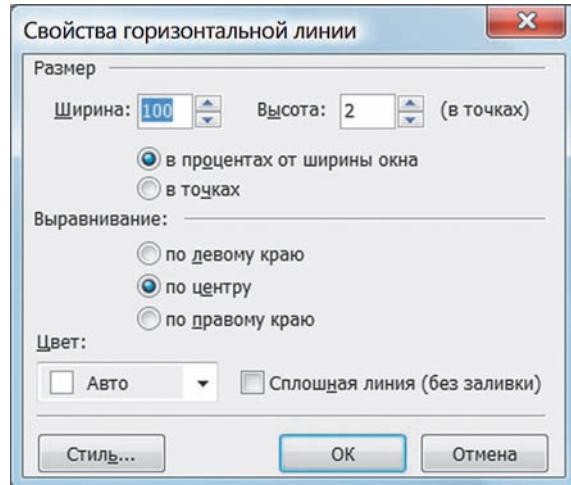
З дапамогай каманды **Вставка** → **Горизонтальная линия** змесцім на старонцы гарызантальную раздзяляльную лінію. У акне **Свойства горизонтальной линии** зададзім параметры лініі ў адпаведнасці з рысункам 1.37.

Цяпер старонка будзе выглądaць так, як на рысунку 1.32.

Захаваем створаны дакумент. Прапанаванае па ўмаўчанні імя нов_стр_1.htm заменім на озера.htm.

Пярэдзем да размяшчэння на створанай старонцы відарысаў. Яшчэ раз падкрэслім, што ёсце відарысы, якія мы бачым на вэб-сторонках, захоўваюцца ў асобных файлах, а на самой старонцы ёсць толькі спасылкі на адпаведныя файлы.

Прыклад 2. Змясціць на вэб-сторонцы «Озера Беларуси» відарысы ў адпаведнасці з рысункам 1.38.



Рыс. 1.37

озера.htm

<body> <p>

ОЗЕРА БЕЛАРУСИ

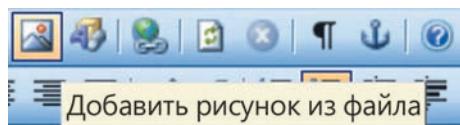
- **Нарочь — озеро в Мядельском районе Минской области.**
Самое большое озеро в Беларуси.
Входит в Нарочанскую группу озер (Мястро, Баторино, Бледное).
Площадь — 79,6 кв. км. Наибольшая глубина — 24,08 м.
В Нарочи обитает 22 вида рыб (в том числе угорь), из них наиболее крупная — щука.
На озере и в округе имеются гнездовья лебедя-шипуна,
малой крачки, скопы, малой поганки.
- **Дриవяты — озеро на западе Витебской области.**
Наиболее крупное озеро среди Браславских озер и пятое в Беларуси.
Площадь — 36,1 кв. км. Наибольшая глубина — 12 м.
В озере также обитает более 20 видов рыб, наиболее распространены угорь, судак, лещ, сазан.
Рядом с берегом гнездится лебедь-шипун.

Конструктор С разделением Код Просмотр | <

Рыс. 1.38

Правообладатель Народная асвета

Паставім курсор у тое месца тэксту, да якога будзе прывязаны рысунак, напрыклад перад пачаткам першага радка. З дапамогай каманды **Вставка → Рисунок** або кнопкі на панэлі інструментаў (рыс. 1.39) выклічам акно **Рисунок** і выберам патрэбны файл з відарысам, напрыклад пагач.jpg.



Рыс. 1.39

Як і ў дакументах MS Word, відарысы могуць размяшчацца адносна тэксту па-рознаму. Па ўмаўчанні рысунак адлюстроўваецца ў выбраным месцы радка без абсякання тэкстам (рыс. 1.40, а). Пры гэтым можна выбраць спосаб выраўноўвання.

На рысунку 1.40, а тэкст і відарыс маюць выраўноўванне па левым краі. Рысунак 1.40, б ілюструе абсяканне відарыса злева. (На гэта паказвае недрукуюмы сімвал стрэлка ўлева у пункце прывязкі. Пры неабходнасці гэты пункт можна перасоўваць з дапамогай мышы.)



- **Нарочь — озеро**

Самое большое озеро в Беларуси.
Входит в Нарочанскую группу озер

- **Нарочь — озеро в Мядельском районе**



Самое большое озеро в Беларуси.
Входит в Нарочанскую группу озер

Площадь — 79,6 кв. км.
В Нарочи обитает 22 вида
крупная — щука.

На озере и в округе имеются гнездовья лебедей,
малой крачки, скопы, малой поганки.

Рыс. 1.40

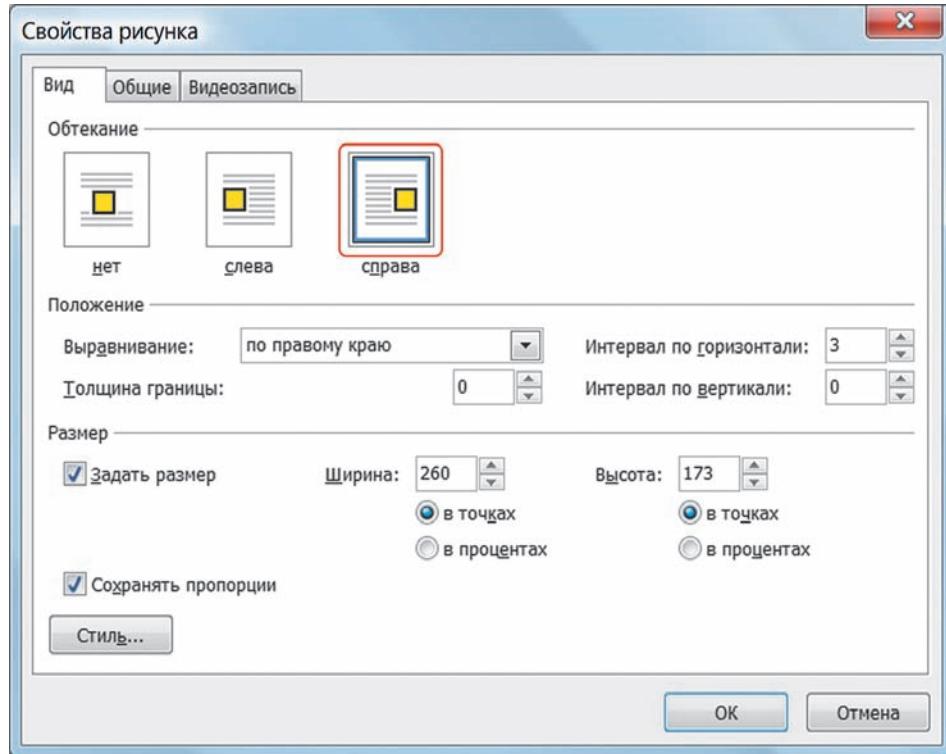
а

б

Параметры відарыса настройваюцца ў дыялогавым акне **Свойства рисунка** (рыс. 1.41). Выклічам гэта акно двайнай пstryчкай мышы па выбраным рысунку або з дапамогай кантэкстнага меню. Зададзім **Обтекание: справа, Выравнивание: по правому краю, Толщина границы: 0** і іншыя параметры ў адпаведнасці з рысункам 1.41. Тут жа можна пазначыць памер відарыса, але прасцей падабраць памер рысунка, вылучыўши яго і перасунуўши мышшу маркёры, як у любым рэдактары.

Аналагічным спосабам уставім другі рысунак з файла braslav.jpg.

У рэдактары FrontPage прадугледжана магчымасць найпрасцейшай апрацоўкі відарысаў (карэкцыя яркасці і кантрастнасці, павароты, абрэзка і інш.). З дапамогай меню **Вид** выклічам панэль інструментаў **Рисунки**. Павялічым



Рыс. 1.41

яркасць і паменшым контрастнасць відарысаў, націснуўшы адпаведна кнопкі і .

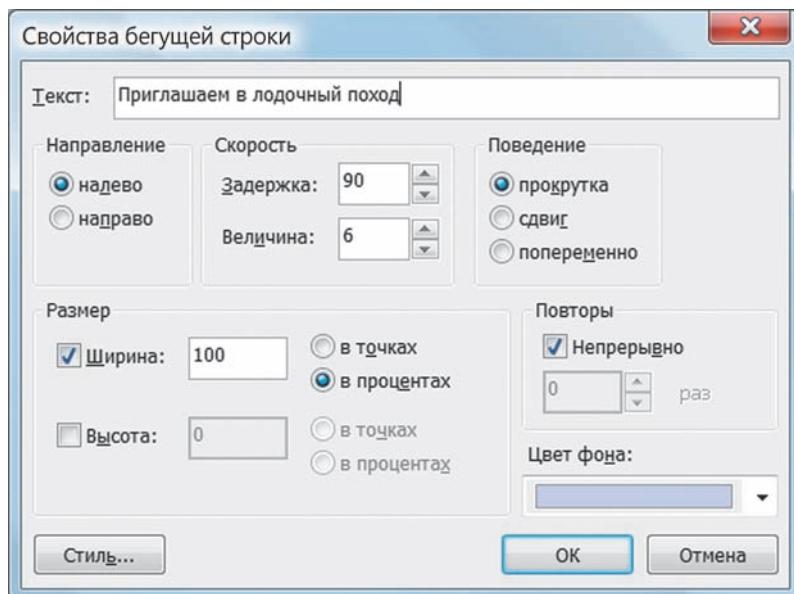
Старонка набудзе патрэбны выгляд (гл. рыс. 1.38). Захаваем яе пад іменем озера2.htm і праглядзім у браўзеры.

Для прыцягнення ўвагі да інфармацыі можна стварыць бягучы радок.



Прыклад 3. На вэб-старонцы «Озера Беларуси» змясціць бягучы радок «Приглашаем в лодочный поход».

Паставім курсор мыши пад ніжній гарызантальны лініяй. Выканаем паслядоўнасць дзеянняў: **Вставка → Веб-компонент → Бегущая строка**. У акне **Свойства бегущей строки** ўвядзём тэкст «Приглашаем в лодочный поход». Зададзім яе параметры ў адпаведнасці з рысункам 1.42. Шрыфт, памер і напісанне тэксту ўстановім з дапамогай меню **Шрифт**, якое выклікаецца кнопкай **Стиль**.



Рыс. 1.42

У выніку ў ніжняй частцы старонкі з'явіцца светла-блакітная паласа, па якой справа налева будзе рухацца тэкст

Приглашаем в лодочный поход



1. Якое прызначэнне рэдактара FrontPage?
2. Якія магчымасці фарматавання тэксту мае FrontPage?
3. Як задаецца памер шрыфту на вэб-старонках?

Практыкаванне

Стварыце адну з вэб-старонак: Рэкі Беларусі, Запаведнікі, Паркі.

Выкарыстайце фатаграфіі і тэксты з прapanаваных настаўнікам файлаў. Афор-міце старонку ў адпаведнасці з узорам на рысунку 1.38.

5.3. Выкарыстанне табліц

Змяшчаць тэкст у некалькі калонак, а таксама ўстаўляць рысункі і тэкст у пэўныя месцы вэб-старонак зручна з дапамогай табліц.

Для ўстаўкі табліцы прасцей за ўсё выкарыстоўваць стандартную панэль інструменталаў, аднак найбольш шырокая магчымасці работы з табліцамі прадстаўлены ў меню **Таблица**. Напрыклад, у дыялогавых вокнах **Вставка таблицы** і **Свойства таблицы**

Правообладатель Народная асвета

ства таблицы можна задаць не толькі колькасць радкоў і слупкоў табліцы, але і спосаб выраўнення, колер фону ячэек, колер і таўшчыню меж, інтэрвал паміж ячэйкамі і іншыя параметры.

Заўважым, што змяніць параметры можна і пасля стварэння табліцы. У гэтым выпадку зручна выкарыстоўваць панэль **Таблицы** (рыс. 1.43), якая выклікаецца з дапамогай каманды **Вид → Панели инструментов**.

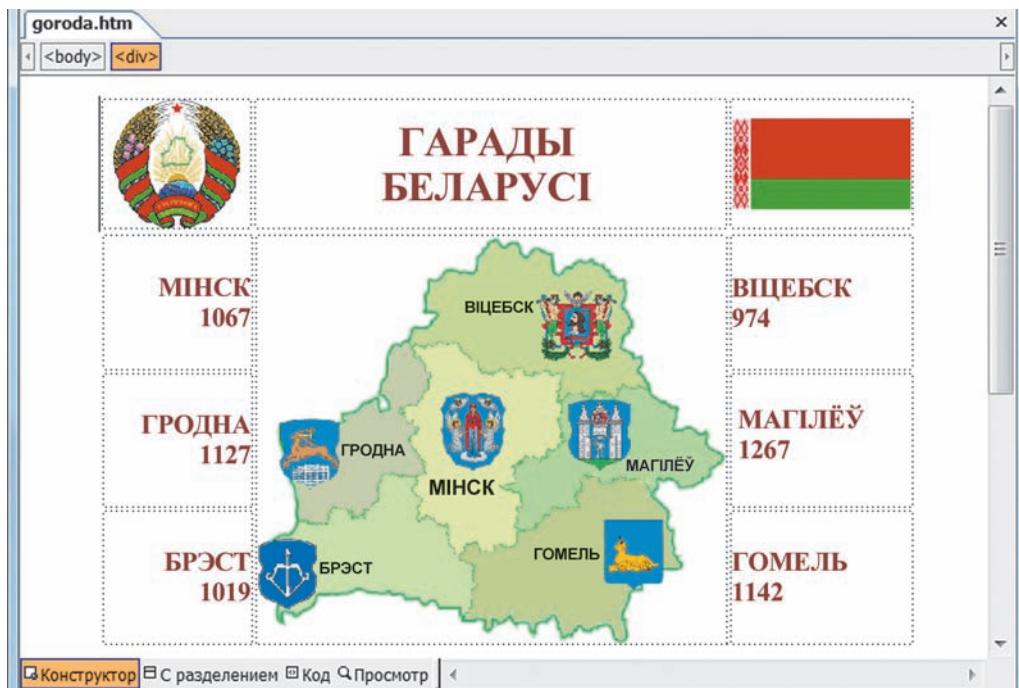


Рыс. 1.43

Разгледзім на прыкладзе, як выкарыстоўваюцца табліцы для размяшчэння аб'ектаў на вэб-старонках.

Прыклад 1. Стварыць вэб-старонку «Гарады Беларусі», на якой паказана карта Беларусі з гербамі гарадоў і пазначаны гады іх заснавання (рыс. 1.44).

Адкрыем рэдактар MS FrontPage. З дапамогай кнопкі на панэлі інструментоў або меню **Таблица** ўставім табліцу з чатырох радкоў і трох слупкоў.



Рыс. 1.44

Правообладатель Народная асвета

У сярэдняй ячэйцы першага радка набяром загаловак «Гарады Беларусі». Астатнія ячэйкі другога слупка аб'яднаем. Для гэтага вылучым гэтыя ячэйкі і выканаем каманду **Таблица → Объединить ячейки**. У атрыманую ячэйку ўставім відарыс карты Беларусі з файла belarus2.jpg.

У ячэйкі 1 і 3 першага радка ўставім відарысы герба і флага Беларусі. У астатніх ячэйках змесцім назвы гарадоў і гады іх заснавання. Адфарматуем надпісы і ўстановім памеры ячэек у адпаведнасці з рэсункам 1.44. Прасцей за ўсё змяніць шырыню слупкоў, перасоўваючы іх межы з дапамогай мышы.

Зробім нябачнымі межы ячэек. Вылучым табліцу і выклічам дыялогавае акно **Свойства таблицы** (рыс. 1.45). У полі **Положение** ўстановім **Выравнивание: по центру**, **Размер: 0**. Такія межы ў рэдактары FrontPage па-

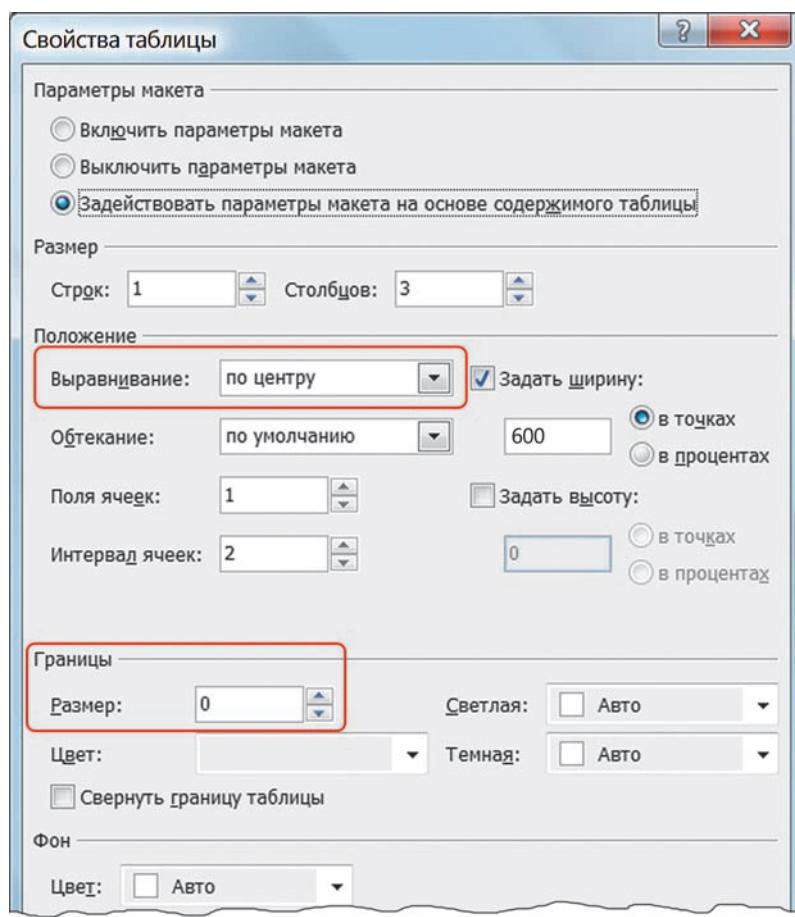


Рис. 1.45

значаюцца штрыхавымі лініямі, а пры праглядзе вэб-старонкі ў браўзера нябачныя.

Цяпер старонка будзе выглядаць так, як на рысунку 1.44. Захаваем яе пад іменем goroda.htm і прагледзім у браўзера.



Для пабудавання табліцы ў мове HTML выкарыстоўваюцца тэгі `<table>` — табліца; `<tr>` — радок табліцы; `<td>` — ячэйка табліцы.

У гэтых тэгах можна выкарыстоўваць атрыбуты `width` — шырыня табліцы або ячэйкі, `bgcolor` — колер фону табліцы або ячэек, `border` — таўшчыня рамкі, `bordercolor` — колер рамкі і іншыя (гл. Дадатак 1).

Шырыня табліцы можа быць зададзена ў пікселях або ў працэнтах адносна шырыні старонкі, напрыклад `width=450` або `width=40%`.

Па ўмаўчанні пры адлюстраванні табліцы ствараецца эфект выпукласці яе меж. Калі задаць атрыбуты `border=1` і `bordercolor=green`, то эфект выпукласці зникне і табліца будзе абмежавана двайной рамкай зялёнаага колеру. Атрыбут `cellspacing=0` задае адзінарную рамку.



Прыклад 2. У акне кода рэдактара FrontPage стварыць табліцу з чатырох слупкоў і аднаго радка шырынёй 600 пікселяў з рамкай зялёнаага колеру таўшчынёй 1 піксель. Змясціць у ячэйках табліцы відарысы (рыс. 1.46).

tabl2.htm

```

<html>
<head> <title>табліца 2</title> </head>
<body>
    <table width="600" border="1" bordercolor=green>
        <tr>
            <td><img src = ornament.gif></td>
            <td><img src = romashka.jpg></td>
            <td><img src = vasilek.jpg></td>
            <td><img src = ornament.gif></td>
        </tr>
    </table>
</body>
</html>

```

Конструктор С раздзеленіем Код Просмотр

Рыс. 1.46

Правообладатель Народная асвета

Адкрыем рэдактар MS FrontPage. Пяройдзем у рэжым **С разделением**. У верхнім акне набяром HTML-код. У ніжнім акне будзе адлюстраваны вынік (гл. рыс. 1.46). Захаваем дакумент пад іменем tabl2.htm. Прагледзім вэб-старонку ў браўзеры.



1. Для чаго выкарыстоўваюцца табліцы на вэб-старонках?
2. Як уставіць табліцу ў рэдактары FrontPage?

Практыкаванні

1. Стварыце вэб-старонку «Разложение белого света в спектр».

Для гэтага выкарыстайце табліцу з чатырох слупкоў і сямі радкоў. У аб'яднаныя ячэйкі левага і правага слупкоў устаўце відарысы прызмы і вясёлкі з файлаў rgizma.jpg і raduga.jpg. У ячэйках другога слупка змясціце назвы колераў і за-ліце іх адпаведным фонам. У ячэйках трэцяга слупка змясціце мнеманічнае пра-віла размяшчэння колераў спектра.



2. Стварыце вэб-старонку па адной з тэм: Прырода роднага краю. Расліны. Жывёлы. Птушкі. Рыбы. Дары лесу. Грыбы. Ягады. Наш сад. Агародніна. Сада-віна. Кветкі.

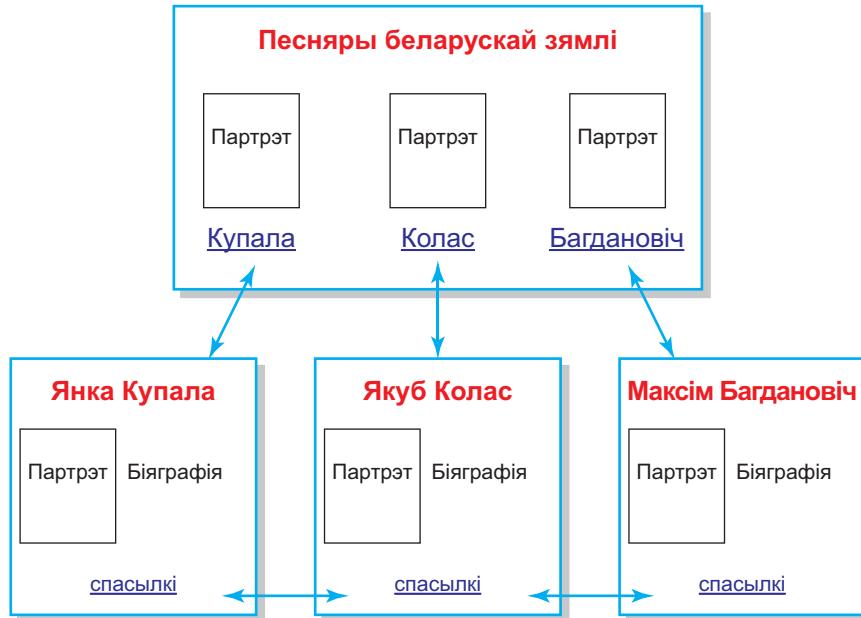
Тэксты і відарысы з файлаў змясціце ў табліцы з 4-6 ячэек. У якасці ўзору афармлення выкарыстайце вэб-старонкі з прыкладаў 1 і 2.

5.4. Распрацоўка вэб-сайта

Разгледзім асаблівасці распрацоўкі вэб-сайта ў рэдактары FrontPage на прыкладзе сайта «Песняры беларускай зямлі».

Этап праектавання сайта не адрозніваецца ад апісанага раней. Створым сайт з чатырох старонак (галоўнай і трох персанальныx: «Якуб Колас», «Янка Купала» і «Максім Багдановіч»). Усе старонкі сайта будзем захоўваць у файлах з адпаведнымі імёнамі ў адной папцы, напрыклад *ristmen*. Структуру сайта пакажам у выглядзе двухузроўневай схемы (рыс. 1.47).

Правообладатель Народная асвета



Рыс. 1.47

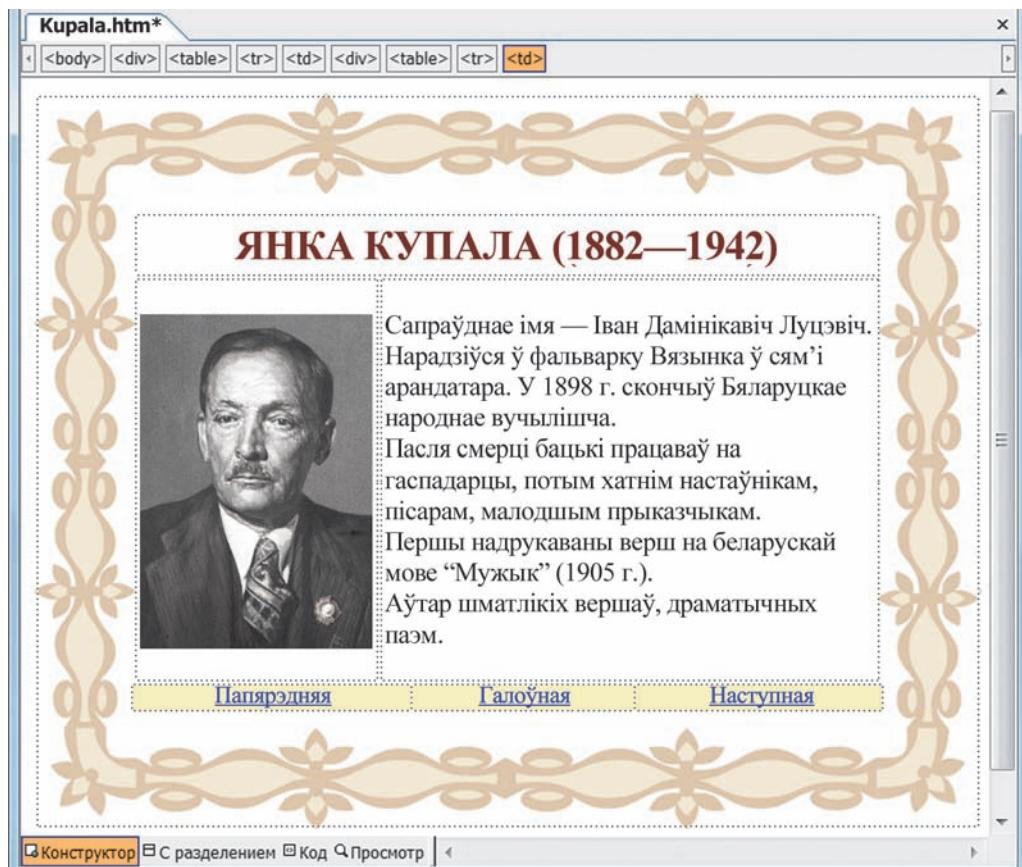
На галоўнай старонцы змесцім загаловак «Песні беларускай зямлі» і гіперспасылкі, якія дазваляюць адкрываць старонкі другога ўзроўню. У якасці гіперспасылак будзем выкарыстоўваць не толькі тэкст (напрыклад, прозвішчы), але і відарысы (напрыклад, невялікія партрэты).

Кожная персанальная старонка будзе змяшчаць тэкставую (напрыклад, біяграфічныя звесткі) і графічную інфармацыю (напрыклад, партрэты, фатаграфіі памятных месцаў). Унізе будуць змешчаны трох тэкставыя гіперспасылкі: адна для вяртання на галоўную старонку сайта і дзве для пераходу з адной персанальной старонкі на другую.

Загадзя падрыхтаем усе відарысы. Змесцім іх ва ўкладзенай папцы *pismen/Ris/*.

Непасрэднае канструяванне пачнём з персанальных старонак, якія аформім у адзіным стылі.

Напомнім, што для палягчэння падбору элементаў афармлення можна выкарыстоўваць гатовы шаблон афармлення. У рэдактары FrontPage пропануецца некалькі шаблонаў — **Тэм**. Тэму можна прымяняць да асобных старонак, а таксама да ўсяго сайта. У апошнім выпадку пры стварэнні кожнай новай старонкі тэма задаецца аўтаматычна.



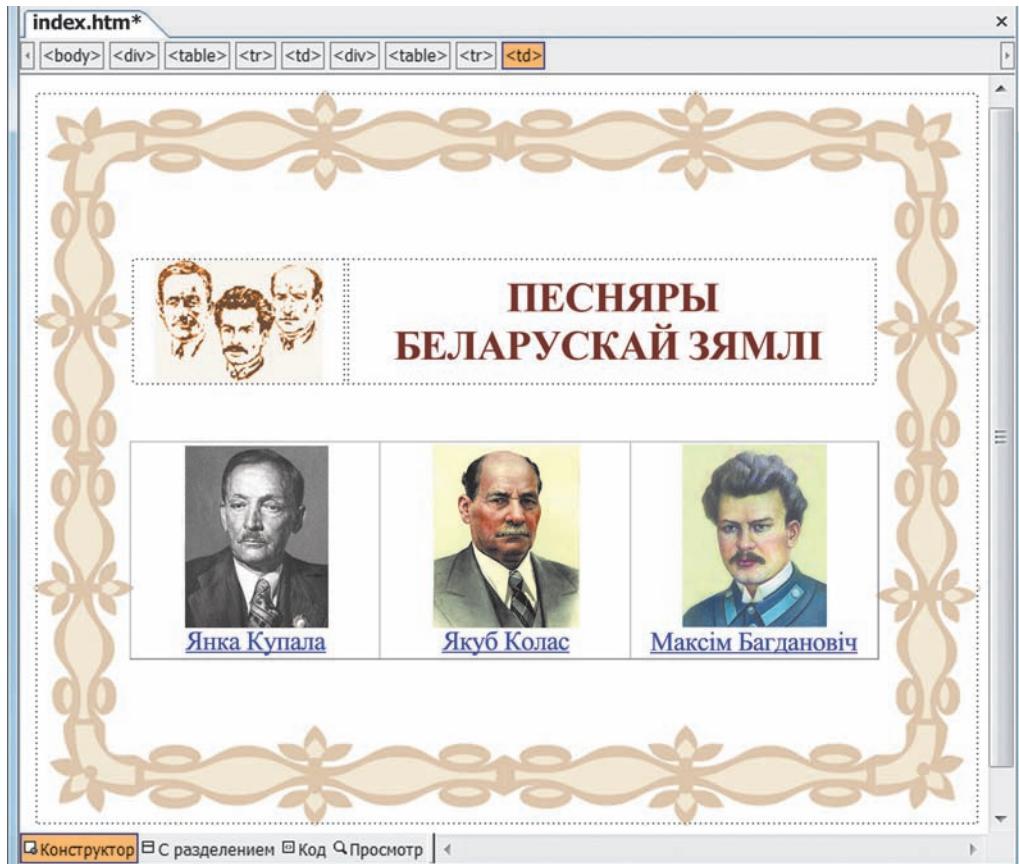
Рыс. 1.48

Прыклад 1. Стварыць вэб-сторонку «Янка Купала» (рыс. 1.48).

Запусцім рэдактар FrontPage. Адкрыецца пустая старонка з устаноўленымі па ўмаўчанні параметрамі. Праверым кадзіроўку (**Windows-1251 — Кириллица**).

Колер фону або фонавы рысунак усёй старонкі задаюцца ў дыялогавым акне **Свойства страницы** на ўкладцы **Форматирование**, якая выклікаецца з дапамогай меню **Формат → Фон**. Мы створым свой шаблон афармлення старонак на аснове відарыса рамкі. Выкарыстаем яго ў якасці фону табліцы. Уставім табліцу з адной ячэйкі. У дыялогавым акне **Свойства таблицы** зададзім яе шырыню (720 пікселяў) і вышыню (560 пікселяў) у адпаведнасці з памерамі рамкі, уключым флагок **Использовать фоновый рисунок** і пазначым імя файла *gamka.gif*. У гэту табліцу ўставім яшчэ трэй табліцы шырынёй 80 %: з адной ячэйкі — для раз-

Правообладатель Народная асвета



Рыс. 1.49

мяшчэння загалоўка, з дзвюх — для рэсунка і біяграфіі, з трох — для тэкставых гіперспасылак.

Аформім старонку ў адпаведнасці з рэсункам 1.48: скапіруем загадзя падрыхтаваны тэкст з файла biograf.txt, уставім рэсунак з файла kupala.jpg. Заўважым, што можна выкарыстоўваць і адну табліцу з аб'яднанымі ячэйкамі, але ў ёй больш цяжка размяшчаць аб'екты ў запланаваных месцах. Захаваем старонку пад іменем Kupala.htm у папцы pismen.

Аналагічна аформім персанальныя старонкі «Якуб Колас» (файл Kolas.htm) і «Максім Багдановіч» (файл Bagdanovich.htm). Галоўную старонку сайта (файл index.htm) аформім у адпаведнасці з рэсункам 1.49.

Напомнім, што сайт уяўляе сабой сукупнасць вэб-старонак, звязаных гіперспасылкамі. Для стварэння гіперспасылкі трэба вылучыць тэкст або рэсунак і за-

Правообладатель Народная асвета

даць адрас пераходу (адрас вэб-сторонкі, імя файла або дакумента) па дадзенай спасылцы. У якасці гіперспасылак могуць выкарыстоўвацца слова, групы слоў, відарысы.

Тэкставыя спасылкі звычайна вылучаюцца колерам і падкрэсліваннем, а графічныя — рамкай. Дзякуючы гэтаму яны адразніваюцца ад звычайнага тэксту і іншых элементаў старонкі. Па ўмаўчанні спасылкі на непрагледжаныя старонкі адлюстроўваюцца сінім колерам, а на ўжо прагледжаныя — фіялетавым.

Прыклад 2. Стварыць тэкставыя і графічныя гіперспасылкі паміж старонкамі вэб-сайта «Песні беларускай зямлі».

Для стварэння гіперспасылкі вылучым партрэт Янкі Купалы і подпіс. З дапамогай кнопкі **Добавление гиперссылки**  на стандартнай панэлі інструментаў або каманды **Вставка → Гиперссылка** выклічам дыялогавае акно, у якім пазначым адрас пераходу Kupala.htm (рыс. 1.50). Аналагічным чынам створым усе спасылкі ў адпаведнасці са схемай навігациі па сайце (гл. рис. 1.47).

Выкліканая пstryчкай мыши па спасылцы старонка па ўмаўчанні адкрываецца ў бягучым акне браўзера. Можна зрабіць так, каб яна адкрывалася ў новым акне браўзера. Для гэтага ў дыялогавым акне **Добавление гиперссылки** пstryкаюць па кнопкы **Выбор рамки** і ў акне **Конечная рамка**, што адкрыеца, выбіраюць варыянт **Новое окно**.

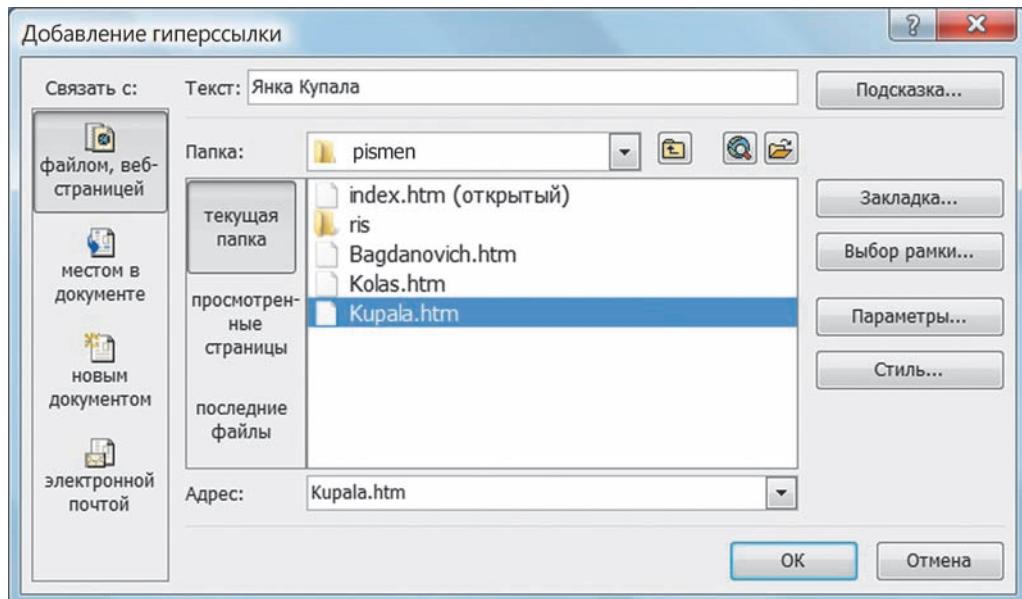


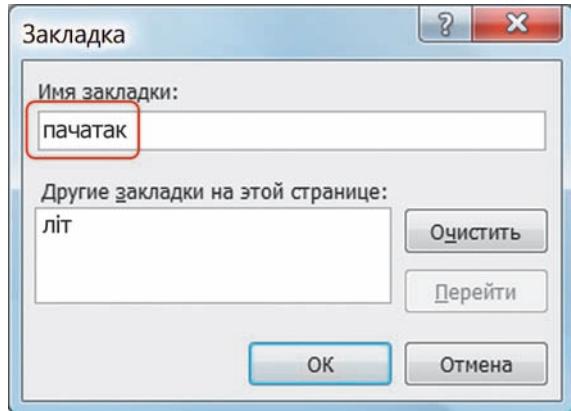
Рис. 1.50

Заўважым, што ў рэдактары FrontPage стварэнне гіперспасылак выконваеца ў рэжыме работы **Конструктор**, а праверыць работу гіперспасылак можна ў рэжыме **Просмотр**.

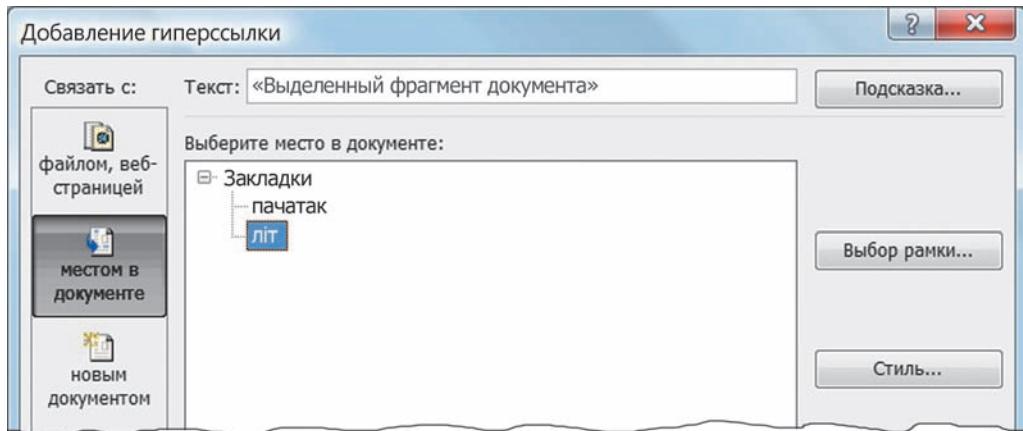
Спасылкі могуць забяспечваць пераход да вэб-старонак або іншых дакументаў, напрыклад рысункаў, презентацый, відэафрагментаў, размешчаных як на дадзеным сайце (*унутраныя спасылкі*), так і на іншых сайтах (*знешнія спасылкі*). Пераходы ўнутры дакумента выконваюць *унутрыстаронковыя* гіперспасылкі. Яны робяць больш лёгкай навігацыю па старонцы, напрыклад забяспечваюць хуткі пераход з канца старонкі ў пачатак.



Перад стварэннем унутрыстаронковых спасылак трэба спачатку расставіць **закладкі**. Для гэтага курсор мыши змяшчаюць у тое месца старонкі, куды мяркуеца пераход па спасылцы. Затым з дапамогай каманды **Вставка** → **Закладка** выклікаюць дыялогаве акно **Закладка**, у якім набіраюць імя закладкі, напрыклад «пачатак» (рыс. 1.51). Далей у акне **Добавление гиперссылки** выбіраюць варыянт **Связать с местом в документе** і пазначаюць патрэбную закладку (рыс. 1.52).



Рыс. 1.51



Рыс. 1.52



У якасці гіперспасылак на вэб-старонках часта выкарыстоўваюць кнопкі, якія пры навядзенні на іх курсора могуць змяніць свае колер і форму. У рэдактары FrontPage прадугледжана магчымасць размішчэння такіх кнопкак. Для гэтага ў меню **Вставка** → **Веб-компонент** выбіраюць кампанент **Інтерактивная кнопкa**. Затым у дыялогавым акне **Меняючыся кнопкa**, што адкрыцца, выбіраюць кнопкку патрэбнай формы і задаюць яе параметры і эфекты.



1. Якія элементы можна выкарыстоўваць у якасці гіперспасылак?
2. На якія аб'екты можа паказваць гіперспасылка?

Практыканне

1. Стварыце фрагмент вэб-сайта «Галерэя вучоных» (пісьменнікаў, спартсменаў, музыкантаў і да т. п.), аформіце яго па прапанаваным ўзоры. Гіперспасылкі на галоўнай старонцы зрабіце ў выглядзе мініяцюраў з подпісамі ўнізе.

	МІКАЛАЙ КАПЕРНІК (1473—1543)
---	-------------------------------------

<p>Мікалай Капернік</p> <p>першы працаваў геліяцэнтрыйную мадэль Сонечнай сістэмы, згодна з якой у цэнтры знаходзіцца Сонца, а Зямля і іншыя планеты рухаюцца вакол па замкнутых арбітах, а не прымацаваны да нябачных крышталічных сфер. Забараняў публікацыю сваіх прац да смерці.</p>	
---	---

 Да спіса вучоных	 Назад	 Наперад
--	---	---

	ГАЛЕРЭЯ ВУЧОНЫХ			
				
Аристоцель	Капернік	Галілей	Ньютан	

Правообладатель Народная асвета

5.5. Публікацыя сайта

Публікацыя сайта называюць яго размяшчэнне на серверы або лакальным камп'ютары з магчымасцю выкліку з сеткі (глабальнай або лакальнай).

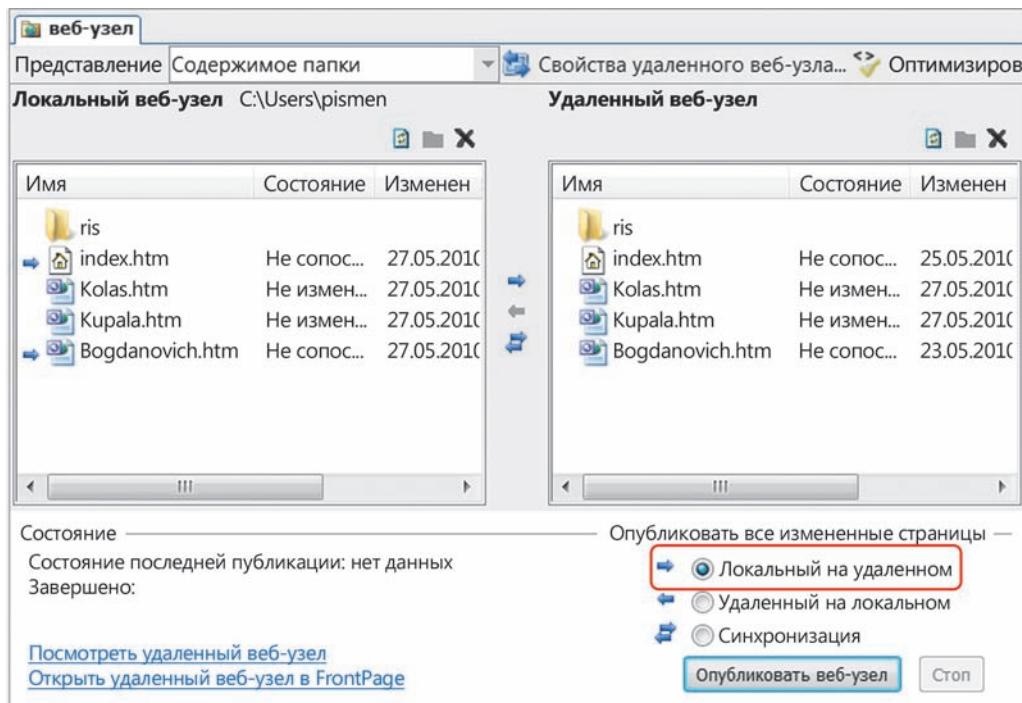
Працэс публікацыі сайта заключаецца ў пераносе файлаў сайта на сервер. Адрас перанесенага сайта можа мець, напрыклад, такі выгляд: www.poets.webserver.by.

Для публікацыі сайта, падрыхтаванага ў рэдактары FrontPage, неабходна з дапамогай каманды **Файл → Опубликовать узел** выклікаць дыялогавае акно **Свойства удаленного веб-узла**.

Для размяшчэння створанага сайта на серверы ў радку **Расположение удаленного веб-узла** трэба пазначыць адрес ftp-вузла, які з'яўляецца аднолькавым для ўсіх карыстальнікаў, і націснуць кнопкі OK.

У акне **Требуются имя и пароль** уводзяць імя і пароль, якія спатрэбяцца ў далейшым для абнаўлення, дабаўлення або выдалення размешчаных на сайце матэрыялаў.

Націсканне кнопкі OK адкрывае акно, на левай панэлі якога (**Локальный веб-узел**) (рыс. 1.53) адлюстроўваюцца файлы і папкі, што змяшчаюць старонкі



Рыс. 1.53

Правообладатель Народная асвета

і рысункі створанага фрагмента вэб-сайта. На правай панэлі гэтага акна (**Удаленный веб-узел**) адлюстроўваюцца файлы і папкі, ужо размешчаныя на вэб-серверы, напрыклад старонкі сайта «Песняры беларускай зямлі» (гл. рис. 1.53).

Калі публікацыя выконваецца ўпершыню, правая панель пустая. Выберам патрэбны рэжым, напрыклад **Все файлы локального веб-узла копировать на сервер**, і націснем кнопкі **Опубликовать веб-узел**.

Як толькі капіраванне ўсіх файлаў скончыцца, карыстальнік зможа праглядзець размешчаны ў сетцы сайт. Для гэтага ў адрасным радку браўзера ўводзіцца адрас сайта, напрыклад www.poets.webserver.by.

- ?** 1. Што разумеюць пад публікацыяй сайта?
- 2. Якія дзеянні трэба выкананы для публікацыі сайта, падрыхтаванага ў рэдактары FrontPage?

§ 6. Стварэнне фрагментаў сайтаў па розных предметных абласцях

Вывучыўшы этапы распрацоўкі вэб-сайтаў і асноўныя прыёмы работы ў рэдактары Front Page, мы можам ствараць сайты, прысвечаныя розным предметным абласцям.

Прыклад. Стварыць фрагмент вэб-сайта «Замкі Беларусі» (рис. 1.54, 1.55).



Рис. 1.54

Правообладатель Народная асвета



Рыс. 1.55

Створым фрагмент сайта з пяці старонак: галоўнай старонкі з маленькімі карцінкамі і чатырох старонак, прысвечаных замкам. Усе старонкі будзем захоўваць у файлах з адпаведнымі імёнамі ў адной папцы, напрыклад замki. Загадзя падрыхтаваныя відарысы змесцім ва ўкладзеную папку замki/.gif/.

Аформім старонкі ў адным стылі ў адпаведнасці з рисункам 1.55. Уставім табліцу з адной ячэйкай. Установім яе шырыню (640 пікселяў) і вышыню (480 пікселяў). У якасці фону табліцы выкарыстаем відарыс скрутка з файла svitok.gif.

Уставім яшчэ адну табліцу з аднаго слупка і чатырох радкоў. У першым радку змесцім загаловак. У другі радок уставім рысунак, напрыклад з файла nesvizh.jpg, у трэці — тэкст з файла zamki.txt. У чацвёртым радку паставім раздзяляльную лінію, тэксты спасылак і відарысы стрэлак з файлаў або Қалекцыі кліпаў.

Галоўную старонку аформім у адпаведнасці з рисункам 1.54. У табліцу з адной ячэйкі з фонавым відарысам скрутка ўставім яшчэ дзве. У першай табліцы змесцім загаловак, у другой — выкарыстаныя ў якасці гіперспасылак карцінкі з подпісамі. Створым гіперспасылкі для пераходаў са старонкі на старонку.

Прагледзім старонкі ў браўзеры. Праверым работу гіперспасылак.

Правообладатель Народная асьвета

Практыкаванне

Стварыце фрагмент сайта (не менш за 3-4 старонкі, што змяшчаюць тэкст і відарысы) па адной з тэм вучэбных прадметаў.

Беларуская літаратура. Жыццё Ефрасінні Полацкай. Жыццё і творчасць Кірылы Тураўскага. Францыск Скарына — першадрукар і асветнік. Мікола Гусоўскі. Песня пра зубра. Адам Міцкевіч. Францішак Багушэвіч. Кандрат Крапіва. Васіль Быкаў. Беларуская паэзія XX стагоддзя.

Русская літаратура. А. С. Пушкин. М. Ю. Лермонтов. Н. В. Гоголь. Н. А. Некрасов. А. Н. Островский. И. С. Тургенев. Ф. М. Достоевский. Л. Н. Толстой. А. П. Чехов. М. Горький. Серебряный век русской культуры. Русская поэзия XX века.

Сусветная гісторыя. Першая сусветная вайна. Каstryчніцкая рэвалюцыя. Утварэнне СССР. Пачатак Вялікай Айчыннай вайны. Блакада Ленінграда. Сталинградская бітва. Курская дуга. Партызанскі рух. Аперацыя «Баграціён». Ка пітуляцыя Германіі. Заснаванне ААН. Распад СССР. Утварэнне СНД.

Гісторыя Беларусі. Грунвальдская бітва. Першая сусветная вайна на беларускіх землях. Гады Грамадзянскай вайны. Вялікая Айчынная вайна. Партызанскі рух. Вызваленне Беларусі. Помнікі гісторыі. Стварэнне Рэспублікі Беларусь.

Геаграфія. Вялікія геаграфічныя адкрыцці. Краіны і кантыненты (Прыродныя ўмовы. Надвор'е і клімат. Насельніцтва. Прамысловасць. Сельская гаспадарка). Геаэкалагічныя праблемы. Змяненне клімату. Забруджванне навакольнага асяроддзя. Ахова прыроды.

Біялогія. Гісторыя адкрыцця клеткі. Будова клеткі. Абмен рэчываў. Фотасінтэз. Экасістэма. Паходжанне жыцця. Біялагічная эвалюцыя. Паходжанне і эвалюцыя чалавека. Біясфера — жывая абалонка планеты. Чалавек і навакольнае асяроддзе. Уплыў розных фактараў асяроддзя на чалавека.

Матэматыка. Графікі функцый. Трыганаметрычныя функцыі. Паказальнай і лагарыфмічная функцыі. Правільныя многавугольнікі. Прасторавыя фігуры. Шматграннікі і іх відарысы. Цэлы вярчэння.

Фізіка. Электрамагнітная індукцыя. Гарманічныя ваганні. Маятнікі. Хвалі. Гук. Электрамагнітныя ваганні. Вытворчасць і перадача электрычнай энергіі. Электрамагнітныя хвалі. Шкала электрамагнітных хваль. Інтэрферэнцыя. Дыфракцыя. Законы праламлення. Аптычныя прыборы.

Хімія. Перыядычна сістэма хімічных элементаў Д. I. Мендзялеева. Атамы і малекулы. Неарганічныя злучэнні. Кісларод. Сера. Азот. Фосфар. Вуглярод. Крэмній. Металы. Хімічная будова арганічных злучэнняў. Вуглевадароды. Перапрацоўка нафты. Сінтэтычныя мылочыя сродкі. Вугляводы. Арганічныя злучэнні з азотам. Амінакіслоты. Сінтэтычныя высокамалекулярныя злучэнні. Прымяненне палімераў. Бялкі.

АСНОВЫ АЛГАРЫТМІЗАЦЫІ І ПРАГРАМАВАННЯ

У гэтым раздзеле разгледжаны прыклады рашэння задач з розных прадметных абласцей.

Рашэнне любой задачы з дапамогай камп'ютара змяшчае некалькі этапаў: аналіз зыходных даных і магчымых вынікаў, пабудаванне матэматычнай мадэлі вывучаемага аб'екта, выбар методу рашэння, састаўленне алгарытму рашэння, напісанне і наладка праграмы, тэсціраванне праграмы.

Асобую цікавасць прадстаўляе рашэнне задач з выкарыстаннем графічных магчымасцей мовы праграмавання. Графічныя элементы паляпшаюць нагляднасць і выразнасць падаваемай інфармацыі ў любой прадметнай вобласці.

§ 7. Выкананне практычных заданняў з розных прадметных абласцей

7.1. Пабудаванне геаметрычных фігур

Мы ўжо ўмеем працаваць з растрывай і вектарнай графікай у графічных рэдактарах, а таксама праграмаваць рысаванне простых відарысаў. Выкарыстаем нашы ўменні для пабудавання геаметрычных фігур.

Пачнём з чарцяжа простай дэталі. Чарцёж з'яўляецца асноўным дакументам любой вытворчасці. Па чарцяжах вырабляюць дэталі, збіраюць машины, будуюць дамы, шыюць вопратку. На ўроках чарчэння Вы вучыліся выконваць чарцяжы на паперы. Сучасная ж вытворчасць немагчыма без камп'ютарных сістэм аўтаматычнага праектавання (напрыклад, AutoCAD, Kompas).

Напомнім, што ў асяроддзі праграмавання PascalABC відарысы можна фарміраваць з простых геаметрычных фігур (графічных прымітываў), напрыклад кропак, адрезкаў, прамавугольнікаў, акружнасцей, эліпсаў. Рысаванне ажыццяўляеца ў графічным акне. Пачатак сістэмы каардынат знаходзіцца ў левым верхнім вугле акна, вось абсцыс накіравана ўправа, а вось ардынат — уніз. Бібліятэка стандартных графічных працэдур захоўваецца ў модулі GraphABC (Дадатак 3).

Пры рабоце з відарысамі ў асяроддзі праграмавання PascalABC вельмі зручна ўсе даныя ўводзіць і выводзіць у графічным акне. Сумяшчаецца работу з тэкстам і графікай ў адным акне можна, падключыўшы модулі crt і GraphABC адначасова.

Правообладатель Народная асвета

Прыклад 1. Саставіць праграму, якая выконвае чарцёж дэталі (рыс. 2.1) і вылічвае яе плошчу. Маштаб: 1 клетка — 20 пікселяў. Радыус адтуліны r у сантиметрах уводзіцца з клавіатуры, $AB = BC = 8$ см.

Вызначым зыходныя даныя. Радыус адтуліны будзем уводзіць у зменную r , памер AB і плошчу фігуры будзем захоўваць у зменных a і S . Усе зменныя будуть мець тып `real`.

Чарцёж дадзенай дэталі можна выканаць рыхаваннем пяці графічных прымітываў: адрезкаў AB , BC , CD , акружнасці з цэнтрам O і дугі AED .

Вызначым экранныя каардынаты пунктаў. Няхай пункт O мае каардынаты $(160, 120)$ (гл. рис. 2.1). Тады з улікам маштабу каардынаты астатніх пунктаў наступныя: $A(80, 120)$, $B(80, 280)$, $C(240, 280)$, $D(240, 120)$. Радыус дугі AED роўны 80. Радыус акружнасці на экране вылічым множаннем уведзенага значэння r на маштаб. Дробавую частку адкінем, каб забяспечыць цэлалікавасць каардынат для працэдур рыхавання: `trunc(r*20)`.

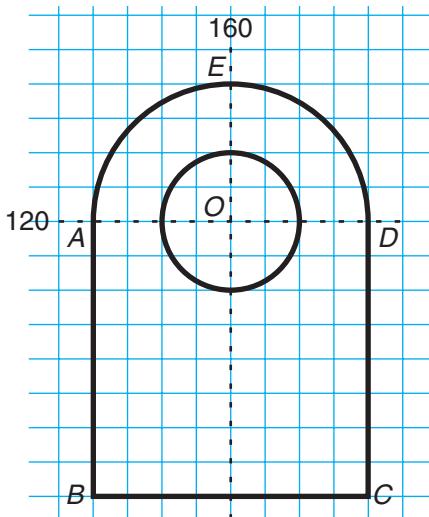
Плошча дэталі складаецца з плошчы квадрата $ABCD$ са стараной a і плошчы паўкруга AED дыяметрам a за вылікам плошчы круга радыусам r .

Праграма можа выглядаць так:

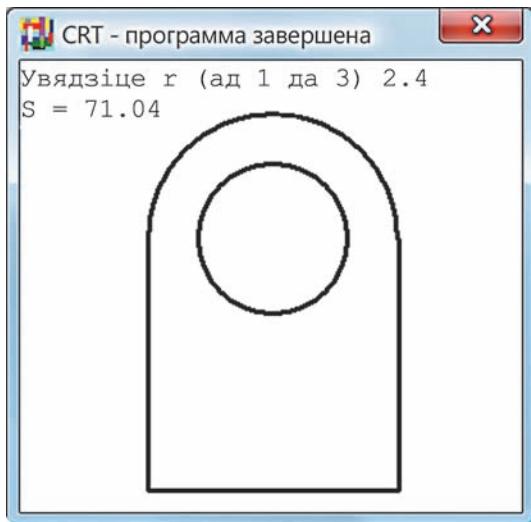
```
program Figural;
uses crt, GraphABC;                                {Падключэнне модуляў}
var r, a, S: real;
begin
  SetWindowSize(320,320);                            {Памеры акна}
  write('Увядзіце r (ад 1 да 3) ');               {Увод радыуса}
  read(r);                                         {Таўшчыня пяра}
  SetPenWidth(3);
  line(80,120, 80,280);                           {Рыхаванне адрезкаў}
  line(80,280, 240,280);
  line(240,280, 240,120);
  circle(160,120, trunc(r*20));                  {Рыхаванне акружнасці}
  arc(160,120, 80, 0,180);                         {Рыхаванне дугі}
  a:=8; S:=a*a+Pi*a*a/8-Pi*r*r;                 {Вылічэнне плошчы}
  write('S = ', S:2:2);                            {Фарматны вывад}
end.
```

Рэзультат работы праграмы для $r = 2,4$ паказаны на рыйсунку 2.2.

Вядома, рыхаваць падобныя фігуры можна ў любым з вывучаных Вамі графічных рэдактараў і нават з дапамогай інструментаў дадаткаў MS Office. Праграмаванне неабходна ў тых выпадках, калі трэба змяніць уласцівасці аб'ектаў



Рыс. 2.1



Рыс. 2.2

або рабіць вылічэнні. Так, у нашым прыкладзе з клавіятуры ўводзяцца значэнні радыуса адтуліны і вылічваецца плошча дэталі. Няцяжка, напрыклад, вылічыць яшчэ і масу дэталі, задаўшы яе таўшчыню і шчыльнасць матэрыялу.



На ўроках матэматыкі Вы навучыліся будаваць просторавыя фігуры: прызму, піраміду, цыліндр, конус, шар. Разгледзім пабудаванне просторавых фігур у асяроддзі PascalABC.

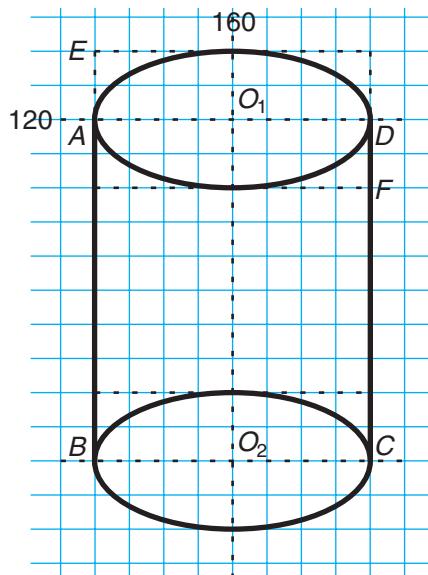


Прыклад 2. Саставіць праграму, якая рысуюе цыліндр (рыс. 2.3) і вылічвае яго аб'ём. Маштаб: 1 клетка — 20 пікселяў. Радыус цыліндра — 4 см. Вышыня цыліндра ў сантиметрах уводзіцца з клавіятуры.

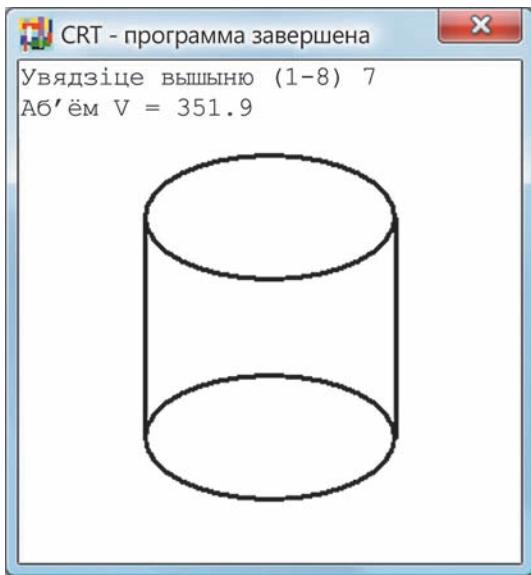
Вызначым зыходныя даныя. Вышыню цыліндра будзем уводзіць у зменную h , радыус і аб'ём будзем захоўваць у зменных r і V . Зменныя h і r будуць мець тып `integer`, а V — тып `real`.

Пры пабудаванні просторавых фігур акружнасці, што не ляжаць у плоскасці чарцяжа, паказваюцца эліпсамі. Такім чынам, пабудаванне цыліндра зводзіцца да рисавання чатырох графічных прымітываў: двух адрезкаў AB , CD і двух эліпсаў з цэнтрамі O_1 і O_2 .

Заўважым, што ў задачах з пабудаваннем фігур лепш спачатку вылічваць неабходныя велічыні, выкарыстоўваючы зададзенныя і ўведзенныя значэнні, і толькі пасля пераходзіць да экранных каардынат з улікам маштабу.



Рыс. 2.3



Рыс. 2.4

Вылічым аб'ём цыліндра, як здабытак плошчы асновы (круга) і вышыні $V = \pi r^2 h$.

Вызначым экранныя каардынаты пунктаў. Няхай пункт O_1 мае каардынаты $x = 160$, $y = 120$. Зробім пералік радыуса і вышыні з улікам маштабу: $r = 20r$ і $h = 20h$. Паколькі маштаб улічаны, каардынаты пунктаў выражаюцца простымі формуламі: $O_2(x, y + h)$, $A(x - r, y)$, $B(x - r, y + h)$, $C(x + r, y + h)$, $D(x + r, y)$.

Нарэшце нарысуем адрезкі і эліпсы. Працэдура `Ellipse` рysуе эліпс, які задаецца апісаным вакол яго прамавугольнікам з каардынатамі процілеглых вяршины, напрыклад: $E(x - r, y - 40)$ і $F(x + r, y + 40)$.

Програма можа выглядаць так:

```
program Figura2;
uses crt, GraphABC;
var x, y, h, r: integer; V: real;
begin
  SetWindowSize(320,320);           {Памеры акна}
  write('Увядзіце вышыню (1-8) ');
  readln (h);                      {Увод вышыні}
  r:=4; V:=Pi*r*r*h;               {Вылічэнне}
```

```

write('Аб'''ём V = ', V:2:1);           {Вывад}
SetPenWidth(3);                         {Таўшчыня пяра}
SetBrushStyle(bsClear);                  {Стыль пяра}
x:=160; y:=100;                        {Экранныя каардынаты}
r:=20*r; h:=20*h;                      {Улік маштабу}
Ellipse(x-r,y-40, x+r,y+40);          {Рысаванне эліпсаў}
Ellipse(x-r,y+h-40, x+r,y+h+40);      {Рысаванне эліпсаў}
line(x-r,y, x-r,y+h);                 {Рысаванне адрезкаў}
line(x+r,y, x+r,y+h);
end.

```

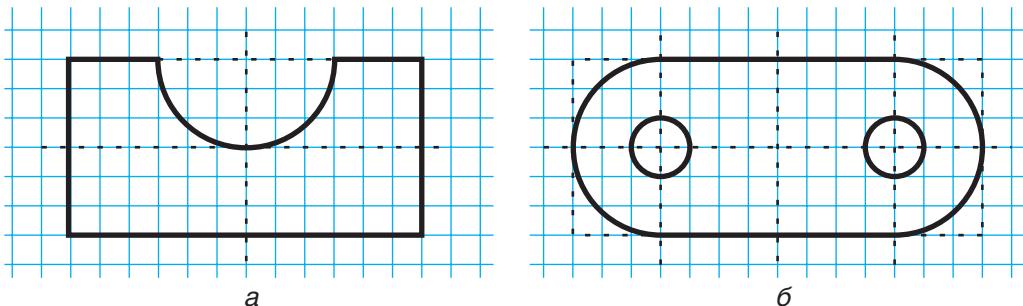
Рэзультат работы праграмы для $h = 7$ паказаны на рымунку 2.4.



- Якія графічныя прымітывы можна выкарыстоўваць пры пабудаванні відараўсяў?
- Якія модулі неабходна падключачы, каб працаўваць з тэкстам і графікай у адным акне?

Практыкаванні

- Састаўце праграму, якая выконвае чарцёж адной з дэталей і вылічвае яе плошчу. Маштаб: 1 клетка — 20 пікселяў.





2. Састаўце праграму, якая рысую адну з прасторавых фігур:

а) куб (даўжыня канта ўводзіцца з клавіятуры);

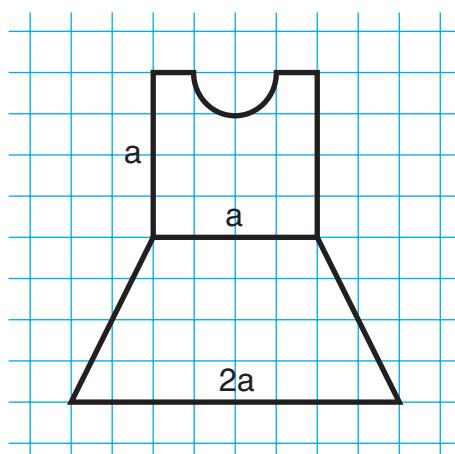
б) конус (значэнні радыуса і вышыні ўводзяцца з клавіятуры).



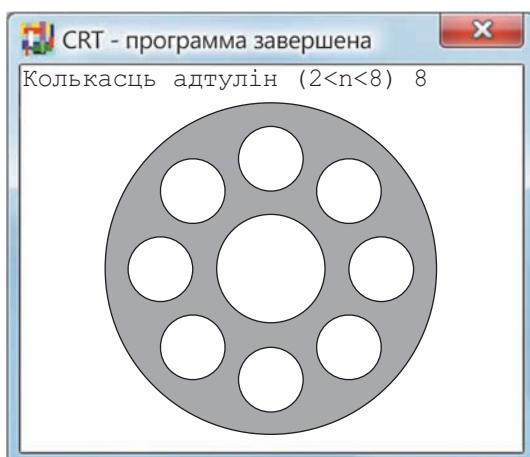
3. Састаўце праграму, якая рысую:

а) выкрайку сукенкі (памер a уводзіцца з клавіятуры);

б) дэталь у выглядзе кольца знешнім радыусам 150 і ўнутраным радыусам 50, у якім выразаны адтуліны радыусам 30. Іх цэнтры знаходзяцца на адлегласці 100 ад цэнтра кольца (колькасць адтулін n уводзіцца з клавіятуры).



a



b

7.2. Выкарыстанне растрравых відарысаў

Графічныя магчымасці мовы праграмавання пашираюцца выкарыстанем растрравых відарысаў. Асабліва цікавае выкарыстанне фатаграфій рэальных аб'ектаў і з'яў жывой і нежывой прыроды (машын, механізмаў, людзей, жывёл, раслін), якія практична немагчыма стварыць з дапамогай стандартных графічных прымітываў.

Працэдуры і функцыі для работы з растрравымі відарысамі змяшчаюцца ў модулі *GraphABC*.

Функцыя *LoadPicture(fname)* загружаем ў аператыўную памяць рысунак з файла з іменем *fname*. Загружаць можна рысункі ў форматах BMP, JPG, GIF, PNG. Кожнаму загружанаму рысунку, які захоўваецца падчас работы праграмы ў аператыўнай памяці, прысвойваецца цэлалікавы апісальнік (дэскрыптар) *nm:=LoadPicture(fname)*. Ён перадаецца ў якасці першага параметра ва ўсе астатнія працэдуры і функцыі работы з рысункамі.

Працэдура `DrawPicture(nm,x,y)` выводзіць рыsunак з апісальнікам `nm` у пазіцыю `(x,y)` графічнага акна, а працэдура `DrawPicture(nm,x,y,w,h)` яшчэ і маштабуе відарыс: задае яго шырыню `w` і вышыню `h`. Калі `w<0`, то рыsunак люстрана адбіваецца адносна вертыкальнай прамой, што праходзіць праз `x`, калі ж `h<0`, то рыsunак люстрана адбіваецца адносна гарызантальнай прамой, што праходзіць праз `y`.

Прыклад 1. Саставіць праграму, якая ілюструе прынцып дзеяння праекцыйнага апарата.

У праекцыйным апараце прадмет (слайд, або рыsunак на празрыстай плёнцы) змяшчаюць паміж фокусам і двайным фокусам лінзы. На экране атрымліваецца павялічаны адваротны відарыс. Для атрымання на экране прамога відарыса слайд пераварочваюць.

Алгарытм праграмы заключаецца ў загрузцы з файла растрравага відарыса і вывадзе яго на экран пасля неабходных пераўтварэнняў (павелічэнне і пераварот). Адзіная зменная `pic` будзе мець тып `integer`.

Перш за ўсё зададзім памеры акна 640×520 і чорны колер фону. Нарысуем «лінзу»: `Ellipse(106,220,122,300)`.

Загрузім відарыс з файла (напрыклад, `robotA.jpg`) і змесцім яго апісальнік у зменную `pic:=LoadPicture('robotA.jpg')`.

З дапамогай працэдуры `DrawPicture(pic,60,310,-50,-100)` выведзем перавернуты відарыс памерам 50×100 пікселяў у пазіцыю з каардынатамі верхняга левага вугла $(60, 310)$, паставіўши перад значэннямі 50 і 100 знакі «мінус». Яшчэ раз выведзем гэты відарыс, але ўжо ў пазіцыю з каардынатамі $(380, 10)$, змяніўши яго шырыню да 250 пікселяў, а вышыню — да 500 пікселяў, г. зн. павялічыўши яго ў 5 разоў: `DrawPicture(pic,380,10,250,500)`.

Засталося правесці дапаможныя лініі, што ілюструюць пабудаванне відарыса, і адкарэкціраваць месцазнаходжанне «лінзы».

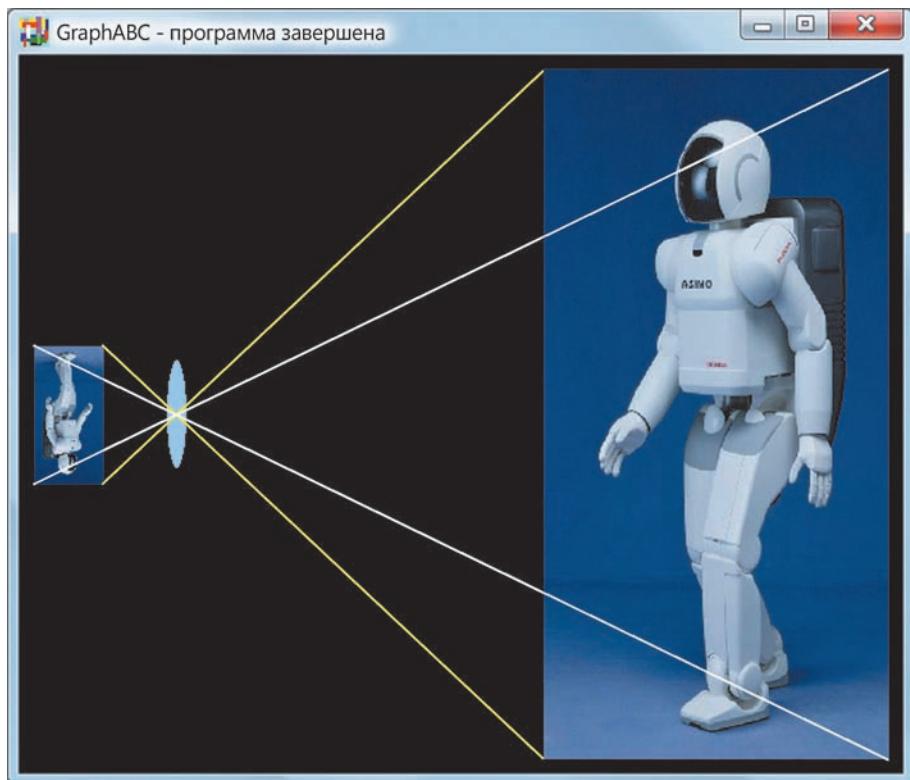
Праграма можа выглядаць так:

```
program Proector;
uses GraphABC;
var pic: integer;
begin
  SetWindowSize(640,520);           {Памеры акна}
  ClearWindow(clBlack);           {Чорны фон}
  SetBrushColor(clSkyBlue);
  Ellipse(106,220,122,300);       {Лінза}
  pic:=LoadPicture('robotA.jpg');   {Загрузка відарыса з
                                    файла}
```

```
DrawPicture(pic,60,310,-50,-100); {Вывад відарыса, пераворот}
DrawPicture(pic,380,10,250,500); {Павелічэнне}
SetPenColor(clYellow);           {Дапаможныя лініі}
line(60,310, 380,10); line(60,210, 380,510);
SetPenColor(clWhite);
line(10,310, 630,10); line(10,210, 630,510);
end.
```

Рэзультат работы праграмы паказаны на рымунку 2.5.

Растрэвавія відарысы выводзяцца ў прамавугольныя вобласці, якія пры стварэнні графічных кампазіцый могуць перакрываць адна адну. Працэдура `SetPictureTransparent(nm,b)` падключочае рэжым празрыстасці відарыса з апісальнікам `nm`. Па ўмаўчанні рэжым празрыстасці адключаны (`b=False`). Калі



Рыс. 2.5

Правообладатель Народная асвета

`b=True`, то пры рысаванні фон не адлюстроўваецца. Фонавым лічыцца колер левага ніжняга пікселя рысунка.

Растравыя відарысы можна выкарыстоўваць у якасці ўзору для зафарбоўкі фігур пэндзлем. Працэдура `SetBrushPicture(fname)` зафарбоўвае фігуры фонавым рысункам, выкарыстоўваючы ўзор, які захоўваецца ў файле `fname`.



У апошні час шырокое распаўсюджанне атрымалі праграмы, прызначаныя для аўтаматычнай генерацыі разнастайных тыповых дакументаў, напрыклад віншавальных паштовак, рэкламных плакатаў, фінансовых дакументаў (накладных, каштарысаў), презентацый. Алгарытмы іх работы заснаваны на выкарыстанні шаблонаў, якія можна дапаўняць некаторым зместам і элементамі афармлення.



Прыклад 2. Саставіць праграму, якая рысуе віншавальныя паштоўкі. З клавіятуры выбіраецца тэма і ўводзіцца тэкст віншавання.

Будзем афармляць усе паштоўкі па шаблоне: злева — заліты растравым відарам прамавугольнік, справа — рысунак казачнага персанажа, зверху — тэкст віншавання.

Нумар тэм з будзе ўводзіцца з клавіятуры (зменная `tema`). Для кожнай тэмы з дапамогай алгарытмічнай канструкцыі выбару `case...of` зададзім імёны файлаў (зменныя `fzal`, `fris`), у якіх будуць захоўвацца прыдатныя відарамі. Напрыклад, для тэмы 1 (зіма) выкарыстаем сняжынкі (`fzal:='snow.gif'`) як фонавы рысунак і Бураціна (`fris:='buratino.gif'`), для тэмы 2 (вясна) — сонейка (`fzal:='sun.gif'`) і Папугая (`fris:='popugai.gif'`), для тэмы 3 (лета) — рамонкі (`fzal:='daisy.gif'`) і Карлсана (`fris:='karlson.gif'`), для тэмы 4 (весень) — лісткі (`fzal:='list.gif'`) і Лісу (`fris:='lisa.gif'`).

Затым будзе ўводзіцца тэкст віншавання (зменная `s`), напрыклад «З Новым годам!».

Зменныя `tema` і `ris` будуць мець тып `integer`, а зменныя `fzal`, `fris`, `s` — тып `string`.

Праграма можа выглядаць так:

```
program Otkrytka;
uses crt, GraphABC;
var tema, ris: integer; fzal, fris, s: string;
begin
  SetWindowSize(600,400);
  write('Увядзіце нумар тэмы: 1-зіма, 2-весна, 3-лета, 4-весень ');
  read(tema);
  fzal:=''; fris:='';
  case tema of
    1: fzal:='snow.gif'; fris:='buratino.gif';
    2: fzal:='sun.gif'; fris:='popugai.gif';
    3: fzal:='daisy.gif'; fris:='karlson.gif';
    4: fzal:='list.gif'; fris:='lisa.gif';
  end;
  writeln(fzal, fris);
  s:='З Новым годам!';
  write(s);
end.
```

```

write('Увядзіце віншаванне '); read(s);
  case tema of
    1: begin fzal:='snow.gif'; fris:='buratino.gif'; end;
    2: begin fzal:='sun.gif'; fris:='popugai.gif'; end;
    3: begin fzal:='daisy.gif'; fris:='karlson.gif'; end;
    4: begin fzal:='list.gif'; fris:='lisa.gif'; end;
  end;
ClearWindow;
SetPenColor(clLtGray);           {Выбар імя файла}
SetBrushPicture(fzal);          {Колер пяра}
SetRectangle(0,0,120,400);        {Устаноўка заліўкі}
ris:=LoadPicture(fris);          {Рысаванне прамавуголь-
                                  ніка}
DrawPicture(ris,380,70, 200,320); {Загрузка відарыса з
                                  файла}
SetBrushStyle(bsClear);          {Вывад відарыса}
SetFontColor(clRed);            {Празрысты фон пэндзля}
FontSize(34);                  {Колер шрыфту}
FontStyle(fsBold);              {Памер шрыфту}
TextOut(60,40,s);               {Паўтлустае напісанне}
                                {Вывад тэксту}
end.

```

Результаты работы программы показаны на рисунке 2.6.

Часта ў задачах трэба даследаваць працяканне працэсаў і з'яў у часе. Для нагляднай дэманстрацыі рашэння такіх задач выкарыстоўваюць метады анимацыі.



Рис. 2.6

Правообладатель Народная асвета

Напомнім, што **анімацыяй** называюць імітацыю руху і змянення формы і іншых бачных уласцівасцей аб'ектаў з цягам часу. У 9-м класе мы навучыліся ствараць анімацыі ў спецыяльным графічным рэдактары Flash.

У асяроддзі PascalABC відарысы можна «ажывіць» шляхам шматразовага рысавання і сцірання. Пазбегнуць пры гэтым маргання экрана дазваляе выкарыстанне працэдуры перарысоўкі `Redraw`.

Прыклад 3.

Саставіць праграму, якая імітуе рух аўтамабіля.

Няхай аўтамабіль перамяшчаецца на фоне будынка і дрэў злева направа на адлегласць 400 пікселяў.

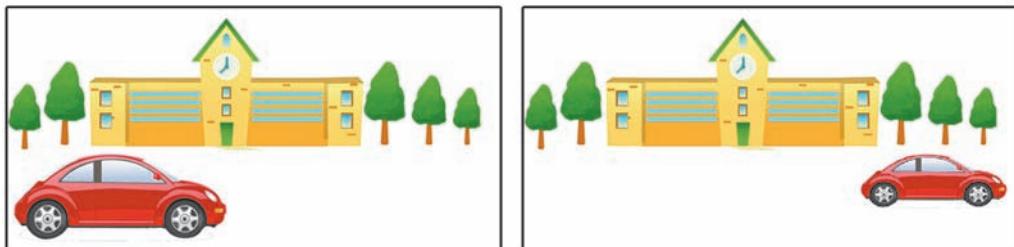
Загрузім відарысы фону і аўтамабіля з файлаў `gorod.gif` і `avto.jpg`, змясціўшы іх апісальнікі ў зменныя `fon` і `avto`. Установім празрыстасць фону для відарыса аўтамабіля: `SetPictureTransparent(avto, true)`. Зададзім пачатковыя каардынаты (x, y), шырыню w і вышыню h відарыса аўтамабіля. Усе зменныя будуть мець тып `integer`.

Працэдуры рысавання і сцірання будзем паўтараць у цыклі з прадумовай `While`, пакуль аўтамабіль не перасунецца на 400 пікселяў. На кожным кроку цыкла каардынату x левага верхняга вугла відарыса будзем павялічваць на 10. Шырыню відарыса будзем памяншаць на 2 пікселі, а вышыню — на 1 піксель для памяншэння аўтамабіля пры аддаленні.

Заўважым, што плаўнасць і працягласць атрыманай дэманстрацыі залежыць ад выбару величыні кроку (павелічэння каардынаты x), колькасці кадраў (паўтарэння цыклу), часу паказу кадра (затрымкі), а таксама хуткадзеяння камп'ютара.

Праграма можа выглядаць так:

```
program Avto3;
uses GraphABC;
var fon, avto, x, y, w, h: integer;
begin
  SetWindowSize(600,300);
  fon:=LoadPicture('gorod.gif');           {Загрузка відарысаў}
  avto:=LoadPicture('avto.jpg');
  SetPictureTransparent(avto,true);         {Празрысты фон}
  x:=0; y:=170; w:=240; h:=100;
  While x<400 do
    begin
      ClearWindow;
      DrawPicture(fon,0,0);                  {Вывад відарысаў}
      DrawPicture(avto,x,y,w,h);
    end;
end.
```



Рыс. 2.7

```

x:=x+10; w:=w-2; h:=h-1;
sleep(20); {Затрымка на 20 мс}
Redraw; {Перарысоўка}
end;
end.

```

Першы і апошні кадры «анімацыі» (пачатковая і канечная месцазнаходжанні аўтамабіля) паказаны на рисунку 2.7.

Праграма ў асяроддзі PascalABC можа маніпуліраваць некалькімі відарысамі, загружаючы іх адразу або па чарзе. Пры выкарыстанні некалькіх графічных аб'ектаў у адным акне і імпартаванні розных відарысаў можна незалежна кіраваць іх уласцівасцямі.



Цікавыя фотарэалістычныя мадэлі атрымліваюцца пры сумесным выкарыстанні растрравай і вектарнай графікі.



Прыклад 4. Саставіць праграму «Секундамер»: на фоне фатаграфіі цыферблата секундамера рухаюцца стрэлкі, якія спыняюцца пры націсканні любой клавішы.

Загрузім відарыс цыферблата з файла Timer.jpg, змясцішы яго апісьальнік у зменную `fon`. Зададзім каардынаты цэнтра вярчэння стрэлак ($x0, y0$) і пачатковыя значэнні секунд `sec:=0` і мінут `min:=0`.

На кожным кроку цыкла з умовай `repeat...until` будзем павялічваць значэнне секунд на 1, пакуль час не перавысіць 60 мін або не будзе націснута любая клавіша (`keyPressed`). Значэнні мінут будзем вылічваць цэлалікавым дзяленнем секунд на 60 (`min:=sec div 60`).

Секундную стрэлку будзем рысаваць лініяй даўжынёй 120 і таўшчынёй 3 пікселя, а мінутную — 100 і 7 пікселяў. Радыянная мера вугла павароту секундной стрэлкі роўна $\pi * sec / 30$, а мінутнай — $\pi * min / 30$. Каардынаты канцоў стрэлак будзяць зменшаны на 50 пікселяў.

лак (x, y) будзем вылічваць па формулах трыганаметры і акругляць да цэльых.
Напрыклад, для секунднай стрэлкі выкарыстаем формулы:

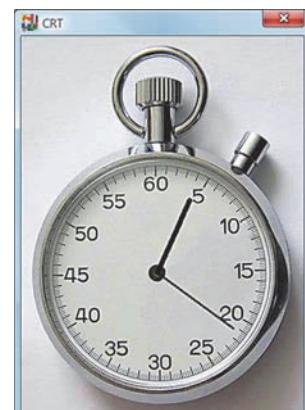
```
x:=x0+Round(120*sin(Pi*sec/30));
y:=y0-Round(120*cos(Pi*sec/30)).
```

Праграма можа выглядаць так:

```
program Timer;
uses crt, GraphABC;
var fon, x0, y0, x, y, sec, min: integer;
begin
  SetWindowSize(360,480);           {Памеры акна}
  HideCursor;                     {Хаванне курсора}
  x0:=173; y0:=300; sec:=0; min:=0;
  fon:=LoadPicture('timer.jpg');   {Загрузка відарыса}
  repeat
    ClearWindow;
    DrawPicture(fon,0,0,360,480);   {Рысаванне цыферблата}
    sec:=sec+1; min:=sec div 60;    {Разлік секунд і мінут}
    x:=x0+Round(120*sin(Pi*sec/30)); {Секундная стрэлка}
    y:=y0-Round(120*cos(Pi*sec/30)); {Каардынаты канца}
    SetPenWidth(3); Line(x0,y0,x,y); {Рысаванне}
    x:=x0+Round(100*sin(Pi*min/30)); {Мінутная стрэлка}
    y:=y0-Round(100*cos(Pi*min/30)); {Каардынаты канца}
    SetPenWidth(7); Line(x0,y0,x,y); {Рысаванне}
    sleep(1000);
    Redraw;
  until (min>=60) or keyPressed;
end.
```

Рэзультат работы праграмы паказаны на рыйсунку 2.8.

- 1. Відарысы якіх фарматай можна загружаць у асяроддзі PascalABC?
- 2. З дапамогай якой працэдуры выводзяцца разрывыя відарысы?
- 3. Як можна імітаваць рух відарысаў у асяроддзі PascalABC?
- 4. Як пазбегнуць маргання экрана пры імітацыі руху відарысаў?



Рыс. 2.8

Практыкаванні

1. Састаўце праграму, якая загружаема з файлаў і выводзіць на экран:
- партрэты вучоных (напрыклад, Эйнштэйна, Рэзерфарда, Бора), пісьменнікаў (напрыклад, Купалы, Коласа, Быкаў), мастакоў, кампазітараў;
 - відарысы жывёл, раслін, помнікаў архітэктуры.



2. Састаўце праграму, якая:

- рысуюе рэкламныя плакаты (турфірмы, канцэрта, спартыўнага свята);
- генерыруе меню школьнай столовай.



3. Састаўце праграму, якая імітуе падзенне кінутага з вежы яблыка.



7.3. Пабудаванне графікаў і дыяграм

Даследаванне законаў прыроды і грамадства немагчыма без пабудавання разнастайных графікаў і дыяграмм. На ўроках матэматыкі Вы даследуецце функцыянальныя залежнасці і будуецце графікі. Выдатнымі магчымасцямі апрацоўкі і прадстаўлення даных у выглядзе разнастайных графікаў і дыяграмм валодаюць электронныя табліцы MS Excel, якія Вы вивучалі ў 10-м класе. Пазнаёмімся з пабудаваннем графікаў і дыяграмм у асяроддзі PascalABC.

Прыклад 1. Саставіць праграму, якая будзе графік залежнасці $y(x) = 0,4x + 2\sin 3x$ на прамежку $[-15; 15]$.

Вызначым зыходныя даныя. Перш за ёсё зададзім памеры акна і пабудуем восі каардынат. Няхай пачатак адліку мае экранныя каардынаты $x_0=320$, $y_0=200$.

Для пабудавання графіка неабходна выбраць маштаб. Зададзім множнік $k=20$ (адзінцы на графіку адпавядаюць 20 пікселяў на экране).

Выконваць вылічэнні і пабудаванні будзем у цыклe While, пакуль $x \leq 15$. Спачатку запішам функцыю $y := 0.4 * x + 2 * \sin(3 * x)$. Затым пярайдзем да экранных каардынат: $x1 := \text{trunc}(x0 + x * k)$; $y1 := \text{trunc}(y0 - y * k)$. Дробавую частку адкінем, каб забяспечыць цэлалікавасць. На кожным кроку цыкла будзем рysаваць акружнасці невялікага радыуса (напрыклад, 2) з каардынатамі цentra $(x1, y1)$, якія сальюцца ў лінію. Для гэтага прырашчэнне аргумента выберам маленькім, напрыклад $x := x + 0.02$.

Пачатковое значэнне $x = -15$ зададзім перад цыклам While.

Зменныя x_0 , y_0 , $x1$, $y1$, k будуть мець тып integer, а зменныя x , y — тып real.

Праграма можа выглядаць так:

```
program Grafik;
uses GraphABC;
var x0, y0, x1, y1, k: integer; x, y: real;
begin
  SetWindowSize(640, 400);
  x0:=320; y0:=200; k:=20;
  line(20,y0, 620,y0); {Вось абсцыс}
  line(x0,20, x0,380); {Вось ардынат}
  setPenColor(clRed); {Колер пяра}
  x:=-15; {Пачатковое значэнне аргумента}
  While x<=15 do
```

```

begin
    y:=0.4*x+2*sin(3*x);      {Функция}
    x1:=trunc(x0+x*k);        {Экранные координаты}
    y1:=trunc(y0-y*k);
    circle(x1,y1,2);
    x:=x+0.02;
end;
end.

```

Результат работы программы показаны на рисунку 2.9.

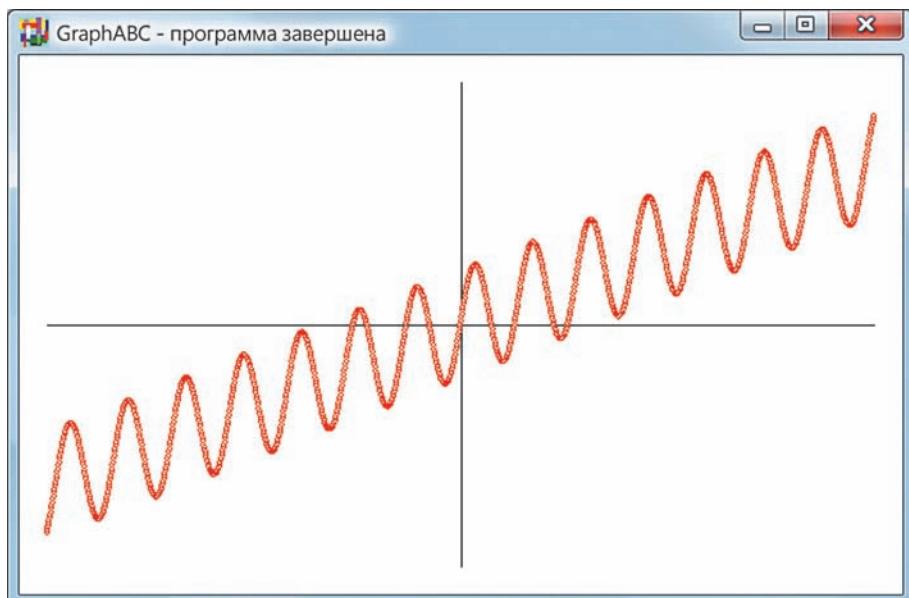


Рис. 2.9



Пры неабходнасці на восіх каардынат няцяжка нарысаваць дзяленні. На прыклад, з дапамогай цыкла `for` з параметрам `i`, які змяняеца ад 0 да 30, на восі абсцыс можна адзначыць 31 дзяленне праз 20 пікселяў. На восі ардынат з дапамогай цыкла `for` з параметрам `i`, які змяняеца ад 0 да 20, можна адзначыць 21 дзяленне.

Фрагмент программы можа выглядаць так:

```

for i:=0 to 30 do line(20+k*i,195, 20+k*i,205);
{Дзяленні па восі абсцыс}
for i:=0 to 20 do line(315,20+k*i,325,20+k*i);
{Дзяленні па восі ардынат}

```



Эфектыўным сродкам нагляднага прадстаўлення даных з'яўляюцца дыяграмы. Яны робяць больш лёгкім парашуннанне даных, дапамагаюць знаходзіць заканамернасці іх змянення. Замест маруднага парашуннання лікаў дастаткова кінучь позірк на дыяграму, каб убачыць, паніжаецца ці расце тэмпература, ураджайнасць, цана або аб'ём продажу.

У 10-м класе Вы пазнаёміліся з пабудаваннем дыяграмм у электронных табліцах. Разгледзім цяпер прыклад праграмавання кругавых дыяграмм.

Кругавыя дыяграмы выкарыстоўваюць у выпадках, калі неабходна паказаць долю кожнай складаючай у агульным аб'ёме даных, напрыклад структуру пасеваў, размеркаванне адзнак па класах і прадметах, цану кожнага тавару ў агульным кошце, склад сплаву або сумесі і да т. п.

Напомнім, што даныя, якія мяркуецца выкарыстоўваць шматразова, зручна захоўваць у масівах.



Прыклад 2. У масівах захоўваюцца даныя аб плошчах пасеваў зерневых культур. Саставіць праграму, якая вылічвае, які працэнт у агульнай плошчы пасеваў зерневых складае плошча пасеваў кожнай культуры, выводзіць даныя і будзе кругавую дыяграму.

Даныя аб плошчах пасеваў зерневых культур будзем захоўваць у масіве цэлалікавых канстант А, а іх назвы — у масіве радковых канстант В:

```
Const A: array[1..5] of integer = (47, 63, 22, 27, 12);
    B: array[1..5] of string = ('Жыта ', 'Пшаніца',
        'Авёс ', 'Ячмень ', 'Грэчка ');
```

Кругавая дыяграма будзе складацца з 5 сектараў. Градусная мера кожнага сектара будзе адпавядаць значэнню пэўнага элемента масіву.

Перш за ёсё ў цыклі `for` вылічым сумарную плошчу ўсіх пасеваў S (ёй адпавядае вугал 360°). Плошчы i -й культуры будзе адпавядаць вугал $A[i]*360/S$.

Для пабудавання кожнага сектара трэба ведаць значэнні двух вуглоў: пачатковага α_0 і канечнага α .

Першы сектар будзем будаваць ад пачатковага вугла $\alpha_0 := 0$. У цыклі `for` будзем вылічваць значэнне канечнага вугла $\alpha := \alpha_0 + round(A[i]*360/S)$. Наступны сектар будзем будаваць ужо ад гэтага вугла, таму яго пачатковы вугал будзе роўны канечнаму вуглу папярэдняга сектара $\alpha_0 := \alpha$.

Колер сектара будзем задаваць выпадковым чынам.

Зменныя α , α_0 , S , i , а таксама элементы масіву А будуть мець тып `integer`, элементы масіву В — тып `string`.

Программа можа выглядаць так:

```
program Diagram2;
uses crt, GraphAbc;
const A: array[1..5] of integer = (47, 63, 22, 27, 12);
      B: array[1..5] of string = ('Жыта ','Пшаніца','Авёс ',
      'Ячмень ', 'Грэчка ');
var u, u0, S, i: integer;
begin
  SetWindowSize(400,400);  S:=0; u0:=0;
  for i:=1 to 5 do S:=S+A[i];    {Вылічэнне сумарнай пло-
  шчы пасеваў}
  for i:=1 to 5 do
    begin
      Writeln(B[i],' ',A[i],' га ',round(100*A[i]/S), '%');
      SetBrushColor(rgb(random(255),
      random(255),random(255)));
      u:=u0+round(A[i]*360/S);   {Вылічэнне канечнага вуг-
      ла сектара}
      Pie(220,240,140,u0,u);   {Рысаванне сектара}
      u0:=u;
      {Новае значэнне пачатко-
      вага вугла наступнага
      сектара}
    end;
end.
```

Результат работы программы показаны на рисунку 2.10.

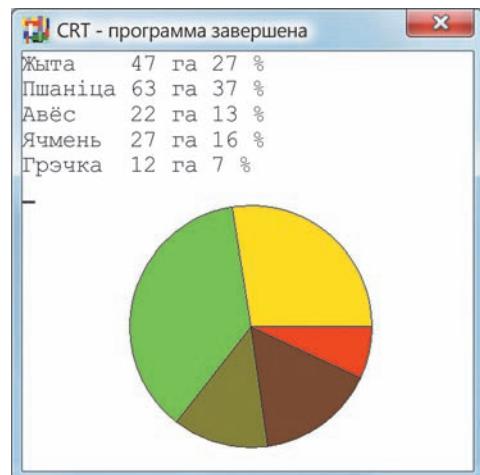
У некаторых задачах можа спатрэбіцца пэўны колер кожнага сектара. Яго зручна вызначаць у масіве цэлалікавых канстант, напрыклад так:

```
col: array[1..5] of integer =
=(clYellow,clLime,colive,
  clBrown,ClRed);
```

і задаваць у цыкле так:

```
SetBrushColor(col[i]);
```

Правообладатель Народная асвета



Рыс. 2.10



У выпадках, калі неабходна ўказваць не толькі лікавыя значэнні, але і напрамак даследуемых велічынь, выкарыстоўваюцца **пялёткавыя дыяграмы**. Такія дыяgramы Вы выкарыстоўвалі ў геаграфіі пры пабудаванні ружы вятроў.



Прыклад 3. Саставіць праграму, якая будзе ружу вятроў (рыс. 2.11). Даная метэралагічных назіранняў за месяц захоўваюцца ў масіве.

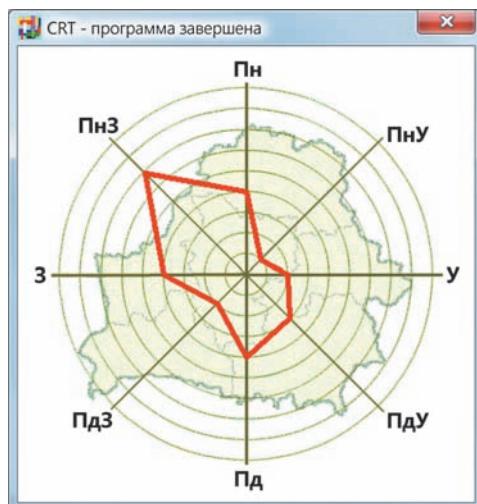
Даная назіранняў аб колькасці дзён з пераважнымі вятрамі пэўнага напрамку будзем захоўваць у масіве канстант N тыпу `integer`.

Дыяграму будзем будаваць на фоне відарыса контурнай карты Беларусі. Загрузім відарыс з файла `karta.gif`, змясціўшы яго апісальнік у зменную `p`. Лініі на карце адпавядаюць восьмі асноўным напрамкам вятроў (`У`, `ПнУ`, `Пн`, `ПнЗ`, `З`, `ПдЗ`, `Пд`, `ПдУ`), а радыусы акружнасцей — колькасці дзён з перавагай гэтых вятроў.

Няхай цэнтр дыяграмы мае каардынаты x_0, y_0 . З дапамогай цыкла `for` з параметрам k , які змяняеца ад 1 да 8 у адпаведнасці з колькасцю напрамкаў ветру, будзем вылічваць каардынаты пунктаў, што адпавядаюць колькасці дзён з вятрамі кожнага напрамку: $x := x_0 + \text{trunc}(20 * N[k] * \cos(u))$; $y := y_0 - \text{trunc}(20 * N[k] * \sin(u))$; дзе напрамак ветру вызначаецца вуглом u (адлічваеца ад усходняга). Злучым гэтыя пункты лініямі: `lineTo(x, y)`.

Праграма можа выглядаць так:

```
program roza_vetrov;
uses crt, GraphABC;
const N: array[1..8] of integer = (2, 1, 4, 7, 4, 2, 4, 3);
var x0, y0, x, y, k, p: integer; u: real;
begin
  SetWindowSize(420,420); HideCursor;
  p:=LoadPicture('karta.gif');
  DrawPicture(p,10,10); {Вывад відарыса карты}
  SetPenColor(clRed); SetPenWidth(5);
```



Рыс. 2.11

```

x0:=210; y0:=210;           {Пачатак сістэмы каардынат}
u:=0; moveTo(x0+20*N[1],y0);
for k:=1 to 8 do begin
    x:=x0+trunc(20*N[k]*cos(u));
    y:=y0-trunc(20*N[k]*sin(u));
    lineTo(x,y); u:=u+pi/4;
end;
lineTo(x0+20*N[1],y0);
end.

```

Результат работы программы показаны на рисунку 2.11.

Практикаванні

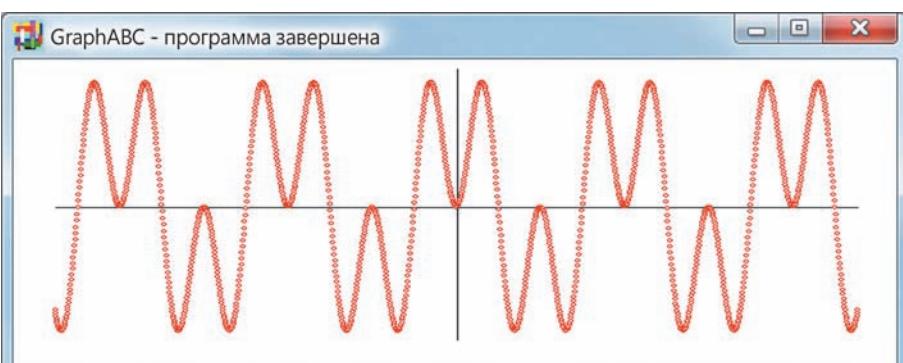
1. Составьте программу, которая будет графиком зависимости:

- a) $y(x) = 0,5x \cdot \cos 2x$ на промежку $[-12; 12]$;
- б) $y(x) = 8 \sin x \cdot \sin 2x$ на промежку $[-15; 15]$.

а



б





2. Састаўце праграму, якая будзе кругавую дыяграму, што ілюструе:
- размеркаванне вынікаў тэсціравання па трох прадметах (даныя ўводзяцца з клавіятуры);
 - хімічны склад зямной кары (даныя ўводзяцца з клавіятуры: кісларод — 47,2 %, крэмній — 27,6 %, алюміній — 8,3 %, жалеза — 5,1 %, кальций — 3,6 %, іншыя элементы — 8,2 %).

§ 8. Выкананне практычных заданняў па тэмах вучэбных прадметаў

8.1. Астрономія

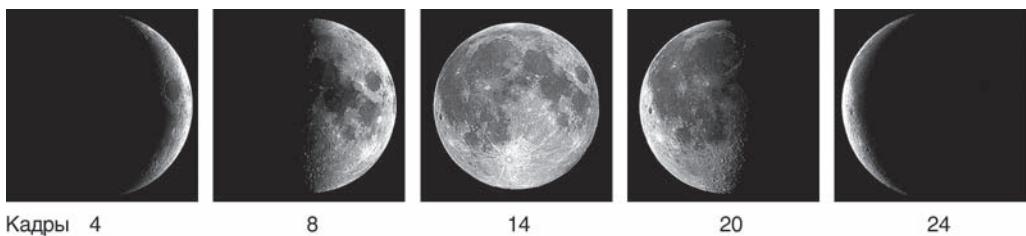
Прыклад 1. Саставіць праграму, якая дэманструе фазы Месяца (відарысы Месяца загружаема з файлаў).

Дэманстрацыя фаз Месяца заключаецца ў змене на экране 28 відарысаў Месяца, што адпавядаюць кожнаму дню месячнага цыкла. Гэтыя відарысы будуць па чарзе загружаемы з файлаў і дэманстравацца з затрымкай у 500 мс.

Выкарыстаем цыкл `for` з параметрам `i`, які змяняецца ад 1 да 28 у адпаведнасці з нумарам дня. Імя файла `imf` будзе мець тып `string`. Сфарміруем яго шляхам зліцця слова '`luna`', пераўтворанага ў радок нумара дня `IntToStr(i)` і пашырэння імя файла з відарысам у фармаце GIF (перед «`gif`» не забудзем пасставіць кропку): `imf:='luna'+IntToStr(i)+'.gif'`.

Праграма можа выглядаць так:

```
program Luna;
uses GraphABC;
var p, i: integer; imf: string;
begin
  SetWindowSize(300,300);           {Памеры акна}
  for i:=1 to 28 do
    begin
      imf:='luna'+IntToStr(i)+'.gif';
                                {Фарміраванне імя файла}
      p:=LoadPicture(imf);        {Загрузка відарыса з файла}
      DrawPicture(p,0,0);         {Вывад відарыса}
      sleep(500);
    end;
end.
```



Рыс. 2.12

Пратэсціруем праграму. На рысунку 2.12 паказаны кадры 4, 8, 14, 20 і 24 (фазы Месяца ў адпаведныя дні месячнага цыкла).

Пры мадэліраванні і дэманстрацыі розных працэсаў і з'яў іх параметры могуць задавацца шляхам уводу з клавіятуры.



Прыклад 2. Саставіць праграму, якая дэманструе павелічэнне відарыса Месяца ў тэлескопе. Каэфіцыент павелічэння ўводзіцца з клавіятуры.

Загрузім відарысы зорнага неба і Месяца з файлаў `sky.gif` і `luna2.gif`, змясціўшы іх апісьльнікі ў зменныя `nebo` і `luna`. Зададзім празрыстасць фону для відарыса Месяца: `SetPictureTransparent(luna, true)`.

Зададзім пачатковыя каардынаты (`x:=200; y:=200`), шырыню і вышыню відарыса (`w:=100; h:=100`).

Каэфіцыент павелічэння будзем уводзіць з клавіятуры і змяшчаць у зменную `k`. Зменныя `nebo`, `luna`, `x`, `y`, `w`, `h` будуць мець тып `integer`, зменная `k` — тып `real`.

Працэдуры рысавання і сцірання будзем паўтараць у цыклে з прадумовай `While`. Няхай на кожным кроку цыкла шырыня і вышыня відарыса павялічваюцца на 10 пікселяў, пакуль не будзе дасягнута патрэбнае павелічэнне `k`, г. зн. пакуль шырыня відарыса не стане роўнай `k*100`. Каб месцазнаходжанне цэнтра відарыса Месяца пры гэтым заставалася нязменным, каардынаты левага верхняга вугла відарыса на кожным кроку будзем памяншаць на 5.

Праграма можа выглядаць так:

```
program Teleskop;
uses crt, GraphABC;
var nebo, luna, x, y, w, h: integer; k: real;
begin
  SetWindowSize(500,500);
  nebo:=LoadPicture('sky.gif');
  DrawPicture(nebo,0,20,500,500);
```

```

luna:=LoadPicture('luna2.gif');
SetPictureTransparent(luna,true);
x:=200; y:=200;           {Пачатковае месцазнаходжанне}
w:=100; h:=100;            {Памеры відарыса}
DrawPicture(luna,x,y,w,h);
write('Увядзіце павелічэнне (<=5). k = ');
read(k);                  {Увод павелічэння}
while w<k*100 do
begin
    w:=w+10; h:=h+10; x:=x-5; y:=y-5;
    DrawPicture(luna,x,y,w,h);
    sleep(20);           {Затрымка на 20 мс}
    Redraw;               {Перарысоўка}
end;
end.

```

Відарысы Месяца на першым і апошнім кроках дэманстрацыі пры $k = 4$ паказаны на рысунку 2.13.



Рыс. 2.13



Прыклад 3. Стварыць праграму, якая дэманструе рух Месяца вакол Зямлі па кругавой арбіце.

Параметры арбіты будзем задаваць яе радыусам R і каардынатамі (x, y) левага верхняга вугла прамавугольніка з відарысам планеты.

Каардынаты пункта, які рухаецца адносна цэнтра вярчэння (x_0, y_0) , будзем вылічваць па формулах трыганаметрыі:

$$x := x_0 + R * \cos(\pi * u / 180) \quad i \quad y := y_0 - R * \sin(\pi * u / 180),$$

дзе u — вугал у градусах, і акругляць да цэлых лікаў.

Правообладатель Народная асвета

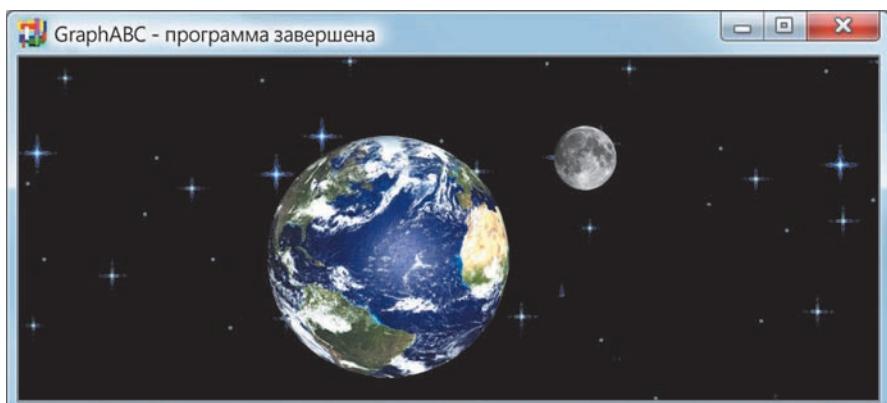
Для аднаго абароту працэдуры рысавання і сірання будзем паўтараць 360 разоў з крокам 1 градус у цыклі For.

Усе зменныя будуць мець тып integer.

Праграма можа выглядаць так:

```
program planety;
uses GraphABC;
var nebo, zem, luna, x0, y0, R, x, y, u: integer;
begin
    SetWindowSize(600,400);
    x0:=300; y0:=200; {Каардынаты цэнтра}
    R:=160; {Радыус арбіты}
    nebo:=LoadPicture('sky.gif'); {Загрузка відарысаў}
    zem:=LoadPicture('earth.gif'); {з файлаў}
    luna:=LoadPicture('luna3.gif');
    for u:=1 to 360 do
        begin
            DrawPicture(nebo,0,0,600,400);
            DrawPicture(zem,200,120,200,200);
            x:=round(x0+R*cos(pi*u/180)); {Каардынаты Месяца}
            y:=round(y0-R*sin(pi*u/180));
            DrawPicture(luna,x,y,50,50); {Рысаванне Месяца}
            sleep(10); {Затрымка на 10 мс}
            Redraw; {Перарысоўка}
        end;
end.
```

Фрагмент дэманстрацыі руху Месяца вакол Зямлі паказаны на рымунку 2.14.

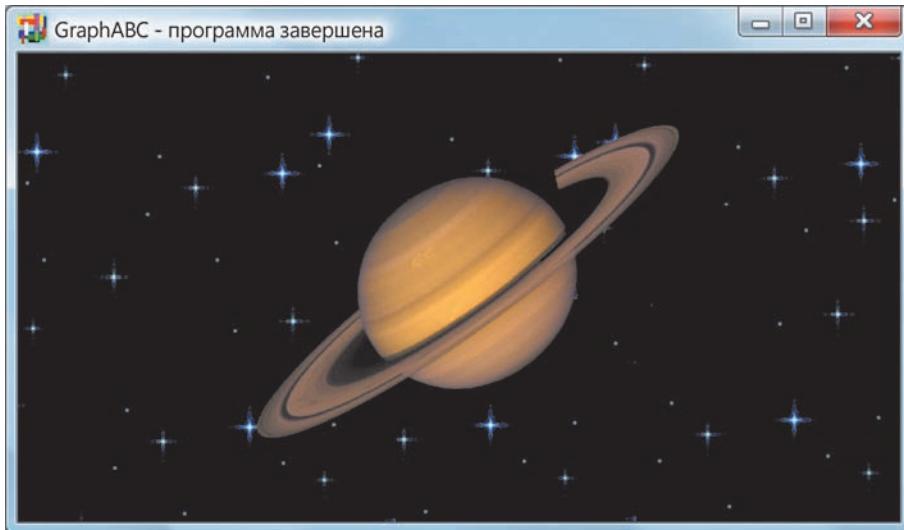


Рыс. 2.14

Правообладатель Народная асвета

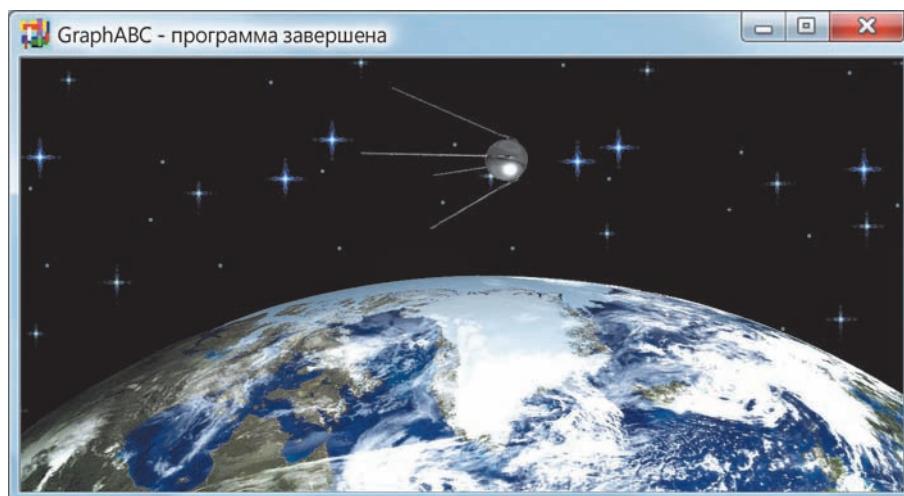
Практыкаванні

1. Састаўце праграму, якая загружаема з файлаў і выводзіць на экран відарыс адной з планет Сонечнай сістэмы на фоне зорнага неба.



2. Састаўце праграму, якая дэманструе:

- а) прамалінейны рух спадарожніка на фоне зорнага неба і Зямлі;
- б) рух планет вакол Сонца па кругавых арбітах.



8.2. Географія

Прыклад 1. Саставіць праграму, што абазначае на карце Беларусі гарады кругамі, плошча якіх прапарцыянальна колькасці жыхароў (Мінск — 1700 тыс. чал., Гомель — 500 тыс. чал., Магілёў — 330 тыс. чал., Віцебск — 340 тыс. чал., Гродна — 320 тыс. чал., Брэст — 300 тыс. чал.).

Загрузім відарыс контурнай карты Беларусі з файла belarus.gif, змясціўшы яго апісальнік у зменную karta.

У найпрасцейшым выпадку будзем паслядоўна прысвойваць зменнай A значэнні колькасці жыхароў гарадоў і запісваць працэдуры рысавання кругоў адпаведнага радыуса R . Радыус будзем вылічваць з умовы $A = k\pi R^2$, дзе коефіцыент прапарцыянальнасці k падбіраецца для атрымання нагляднасці. Зададзім $k = 1$ і акруглім значэнне радыуса $R := \text{round}(\sqrt{A/\pi})$. Каардынаты цэнтраў кругоў адпавядаюць каардынатам гарадоў на карце і падбіраюцца доследным шляхам.

Праграма можа выглядаць так:

```
program Geograf1;
uses GraphABC;
var karta, R, A: integer;
begin
  SetWindowSize(520,480);
  karta:=LoadPicture('belarus.gif');
  DrawPicture(karta,10,10);
  setBrushColor(clRed);
  A:=1700; R:=round(sqrt(A/pi)); circle(250,225,R); {Мінск}
  A:=500; R:=round(sqrt(A/pi)); circle(405,345,R); {Гомель}
  A:=330; R:=round(sqrt(A/pi)); circle(375,215,R); {Магілёў}
  A:=340; R:=round(sqrt(A/pi)); circle(360,105,R); {Віцебск}
  A:=320; R:=round(sqrt(A/pi)); circle( 55,235,R); {Гродна}
  A:=300; R:=round(sqrt(A/pi)); circle( 40,385,R); {Брэст}
end.
```

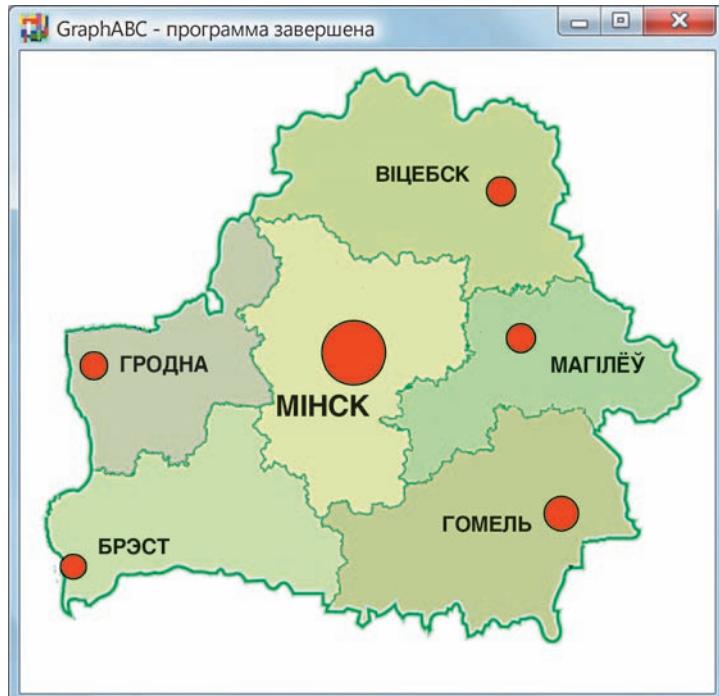
Рэзультат работы праграмы паказаны на рэсунку 2.15.



Прыклад 2. Саставіць праграму, якая выводзіць фізічную карту Беларусі, даныя аб даўжыні рэк і будзе лінейчастую дыяграму.

Даныя аб даўжыні рэк на тэрыторыі Беларусі будзем захоўваць у масіве канстант A тыпу `integer`, а іх назвы — у масіве B тыпу `string`.

Правообладатель Народная асвета



Рыс. 2.15

Відарыс фізічнай карты Беларусі загрузім з файла `belfiz.gif`, змясціўши яго апісальнік у зменную `karta`.

Вывад даных і ліній дыяграмы выканаем з дапамогай цыкла `For`. Даўжыня лініі працаўся нальна даўжыні ракі $A[i]$. Каэфіцыент працаўся нальнасці за дадзім роўным 0,5, абсцысу левага канца — 160. Па вертыкалі лініі будзем размяшчаць праз 20 пікселяў. Іх ардынаты выразім праз параметр цыкла i .

Праграма можа выглядаць так:

```
program Geograf2;
uses crt, GraphABC;
const A: array[1..5] of integer = (690, 613, 495, 493, 459);
      B: array[1..5] of string = ('Дняпро', 'Бярэзіна', 'Прыпяць',
      'Сож', 'Нёман');
var karta, i: integer;
begin
  SetWindowSize(600,520);
  karta:=LoadPicture('belfiz.gif');
  DrawPicture(karta,30,120); {Карта}
```

```

SetPenColor(clBlue);           {Колер пяра}
SetPenWidth(5);               {Таўшчыня пяра}
Writeln('Даўжыні рэк на тэрыторыі Беларусі, тыс. км');
For i:=1 to 5 do begin
    Writeln(B[i],', ',A[i]);   {Вывад данных}
    Line(160, 10+20*i, 160+round(0.5*A[i]), 10+20*i);
                                {Рысаванне ліній}
end;
end.

```

Результат работы программы показаны на рисунку 2.16.

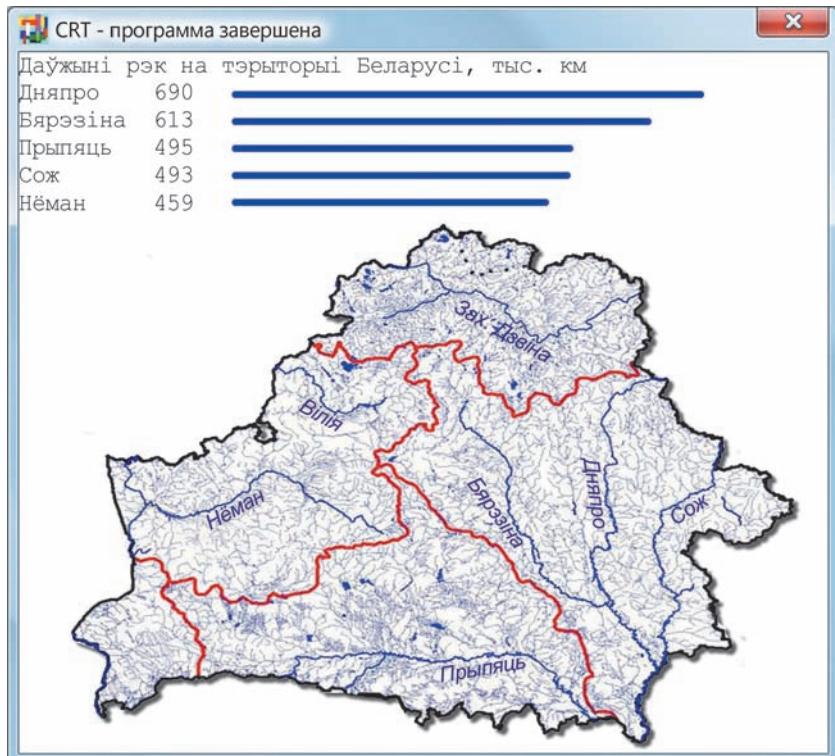


Рис. 2.16



Даныя, што захоўваюцца ў масівах, можна падвяргаць далейшай апрацоўцы. Напрыклад, па колькасці жыхароў і плошчы тэрыторыі краіны (вобласці, раёна) можна вылічыць сярэднюю шчыльнасць насельніцтва.



Прыклад 3. Саставіць праграму, якая вылічвае і паказвае на карце сярэднюю шчыльнасць насельніцтва абласцей Беларусі.

Правообладатель Народная асвета

Неабходныя даныя будзем захоўваць у масівах канстант: назвы абласцей — у масіве $A[i]$, колькасць насельніцтва — у масіве $N[i]$, плошчу — у масіве $S[i]$, каардынаты сімвалу — у масівах $x[i]$ і $y[i]$.

Загрузім відарыс карты і сімвал чалавечка з файлаў *belarus.gif* і *man.gif*, змясціўшы іх апісальнікі ў зменныя *karta* і *ris*. Зададзім празрыстасць фону для відарыса чалавечка.

Вылічэнні шчыльнасці насельніцтва, вывад данных, маштабаванне і размежэнне сімвалу на карце будзем выконваць з дапамогай цыкла *For*. Прымем шырыню сімвала роўнай k , а вышыню — $2k$.

Праграма можа выглядаць так:

```
program Geograf3;
uses crt, GraphABC;
const A: array[1..6] of string = ('Брэсцкая', 'Віцебская',
    'Гомельская', 'Гродзенская', 'Мінская', 'Магілёўская');
    N: array[1..6] of integer = (1440, 1300, 1500, 1100,
    1500, 1200);
    S: array[1..6] of real = (32.8, 40.1, 40.4, 25.1,
    40.8, 29.1);
    x: array[1..6] of integer = (140, 315, 365, 65, 200,
    380);
    y: array[1..6] of integer = (320, 80, 290, 230, 180,
    175);
var karta, ris, k, i: integer; p: real;
begin
    setWindowSize(520,480);
    karta:=LoadPicture('belarus.gif');
    DrawPicture(karta,10,30); {Карта}
    ris:=LoadPicture('man.gif');
    SetPictureTransparent(ris,true);
    Writeln('Шчыльнасць насельніцтва, тыс. чал. на кв. км');
    for i:=1 to 6 do begin
        p:=N[i]/S[i]; k:=round(p); {Разлік шчыльнасці і
        маштабу}
        Writeln(A[i]:11, p:5:1); {Вывад}
        DrawPicture(ris,x[i],y[i],k,2*k); {Вывад відарысаў}
    end;
end.
```



Рыс. 2.17

Рэзультат работы праграмы паказаны на рымунку 2.17.

Практыкаванне

Састаўце праграму, якая:

а) выводзіць карту Беларусі і абазначае нацыянальныя паркі квадратамі, памеры якіх прапарцыянальны плошчы парка: Белавежскай пушча — 87,5 тыс. га, Braslauskiaj azery — 71,5 тыс. га, Prypyatki — 82,4 тыс. га, Naarachanskij — 94 тыс. га;

б) выводзіць карту Беларусі, будзе табліцу з данымі аб плошчы і глыбіні азёр, слупковую або лінейчастую дыяграму (па ўказанні настаўніка);

в) выводзіць карту Еўропы і абазначае краіны (сталіцы дзяржаў) сімваламі, памеры якіх прапарцыянальны колькасці насельніцтва.

8.3. Біялогія і экалогія

Прыклад 1. Састаўіць праграму, якая вылічвае, праз колькі дзён узваровень радыяцыі ў зоне заражэння ўпадзе да бяспечнага значэння ў 10 адзінак, калі кожны дзень ён памяншаецца на 5 % у параўнанні з папярэднім

Правообладатель Народная асвета

днём. Пачатковае значэнне — ад 100 да 10 000 адзінак — уводзіца з клавіятуры.

Пачатковае значэнне ўзроўню радыяці будзем уводзіць у зменную `ur`, а працэкт штодзённага памяншэння ўзроўню, бяспечнае значэнне і колькасць дзён будзем захоўваць адпаведна ў зменных `p`, `ur0` і `t`. Зменныя `ur`, `ur0` будуць мець тып `real`, а працэкт `p` і колькасць дзён `t` — тып `integer`.

Кожны дзень узровень радыяці памяншаецца на $ur * p / 100$ адносна ўзроўню папярэдняга дня, г. зн. вызначаецца выразам $ur := ur - ur * p / 100$.

Вылічэнні будзем паўтараць у цыклі `while`, пакуль будзе выполнваша ўмова $ur >= ur0$, г. зн. пакуль не будзе дасягнуты бяспечны ўзровень радыяці.

Праграма можа выглядаць так:

```
program Ecolog;
var ur0, ur: real; p, t: integer;
begin
    write('Пачатковы ўзровень ');
    readln(ur);
    p:=5; ur0:=10; t:=0; {Пачатковыя значэнні}
    while ur>=ur0 do
        begin
            ur:=ur-ur*p/100; t:=t+1;
        end;
    Write('Праз ', t, ' дзён.')
end.
```

Рэзультат работы праграмы можа выглядаць так:

Пачатковы ўзровень 3000
Праз 112 дзён.



Прыклад 2. Саставіць праграму, якая імітуе назіранне клеткі пад мікроскопам. Для павелічэння выкарыстоўваюцца клавішы кіравання курсорам.

Загрузім відарыс клеткі з файла `kletka.jpg`, змясціўши яго апісальнік у зменную `img`. Зададзім пачатковыя каардынаты `x`, `y` левага верхняга вугла відарыса, яго шырыню `w` і вышыню `h`.

Працэдуры рысавання будзем паўтараць у цыклі з умовай `repeat...until`. З дапамогай функцыі `ReadKey` сімвал націснутай клавішы будзе счытвацца і змяншчацца ў зменную `c` тыпу `char`. Пры націсканні клавішы кіравання курсорам «стрэлка ўверх» (`c=#72`) шырыня і вышыня відарыса павялічвацца на $2 * d = 10$ пікселяў (`d=5`), а пры націсканні клавішы «стрэлка ўніз» (`c=#80`) — памен-

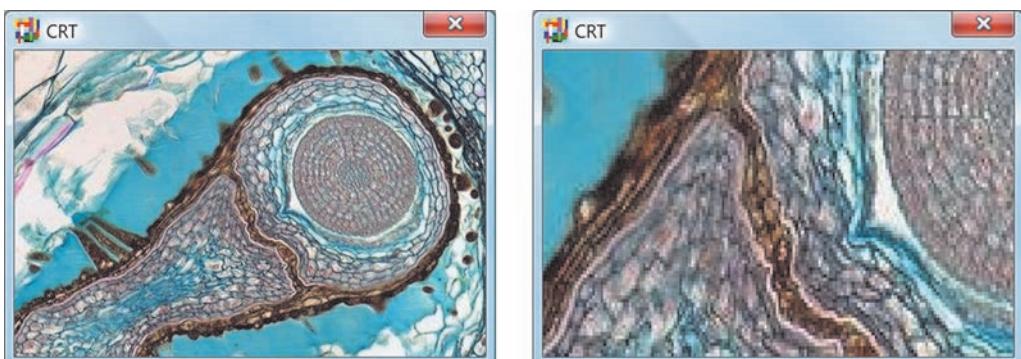
шашца ($d=-5$). Для таго каб месцазнаходжанне цэнтра відарыса пры гэтым заставалася нязменным, будзем змяніць адпаведным чынам каардынаты яго левага верхняга вугла. Пры націсканні клавішы Esc ($c=\#27$) работа праграмы будзе завершана.

Зменныя img , x , y , w , h , d будуць мець тып `integer`, зменная c — тып `char`.

Праграма можа выглядаць так:

```
program Microscop;
uses crt, GraphABC;
var img, x, y, w, h, d: integer; c: char;
begin
  SetWindowSize(400,260);
  img:=LoadPicture('kletka.jpg');
  x:=0; y:=0; {Пачатковыя каардынаты}
  w:=400; h:=260; {Памеры відарыса}
  repeat
    ClearWindow(clSkyBlue);
    DrawPicture(img,x,y,w,h); {Вывад відарыса}
    d:=0; c:=ReadKey;
    if c=#72 then d:=5; {Стрэлка ўверх – павелічэнне}
    if c=#80 then d:=-5; {Стрэлка ўніз – памяншэнне}
    w:=w+2*d; h:=h+2*d; x:=x-d; y:=y-d;
  until c=#27;
end.
```

Рэзультаты работы праграмы паказаны на рымунку 2.18.



Рымунок 2.18

Правообладатель Народная асвета



Прыклад 3. Першапачаткова ўчастак лесу змяшчае 80 тыс. куб. м драўніны. Штогадовы натуральны прырост складае $P\%$, а вырубка — D тыс. куб. м. Саставіць праграму, якая вылічвае штогадовыя аб'ёмы драўніны для T гадоў і вынікі выводзіць у табліцу (рыс. 2.19). Значэнні P (ад 2 да 8 %), D (ад 3 да 8 тыс. куб. м) і T (ад 2 да 20 гадоў) уводзяцца з клавіятуры.

Значэнні аб'ёму драўніны будзем захоўваць у зменай V , а прырост, аб'ём вырубкі і колькасць гадоў будзем уводзіць з клавіятуры ў зменныя P , D і T . Зменныя P , D , V будуть мець тып `real`, а колькасць гадоў T і лічыльнік цыкла i — тып `integer`.

За кошт натуральнага прыросту аб'ём драўніны штогод павялічваецца на $V*P/100$ у параўнанні з папярэднім, а ў выніку вырубкі памяншаецца на D , г. зн. вызначаецца выразам

$$V := V * (1 + P / 100) - D;$$

Рыс. 2.19

Вылічэнні і вывад рэзультатаў у радкі табліцы будзем паўтараць у цыклі `while`, пакуль не будзе дасягнута зададзеная колькасць гадоў ($i \leq T$) і лес не будзе вырублены ($V >= 0$).

Праграма можа выглядаць так:

```
program Les;
uses crt, GraphABC;
var P, D, V: real; T, i: integer;
begin
  write('Прырост (ад 2 да 8 %) ');
  readln(P); {Увод данных}
  write('Вырубка (ад 5 да 15 тыс. куб. м) ');
  readln(D);
  write('Час (ад 2 да 20 гадоў) ');
  readln(T);
  v:=80; i:=0; {Пачатковыя значэнні}
  writeln('-----');
  writeln(' Год Аб''ём, тыс. куб. м'); {Загаловак табліцы}
```

Правообладатель Народная асвета

CRT - программа завершена	
Прырост (ад 2 да 8 %) 3	
Вырубка (ад 5 да 15 тыс. куб. м) 11	
Час (ад 2 да 20 гадоў) 15	
Год	Аб'ём, тыс. куб. м
0	80.0
1	71.4
2	62.5
3	53.4
4	44.0
5	34.3
6	24.4
7	14.1
8	3.5

```

while (i<=T) and (V>=0) do
begin
    writeln(i:3, V:12:1);           {Вывад рэзультатаў}
    V:=V*(1+P/100)-D; i:=i+1;
end;
end.

```

Рэзультат работы праграмы паказаны на рэсунку 2.19.

Зайважым, што паўтарэнне цыкла спынілася раней за ўведзены час (15 гадоў), паколькі ўжо к 9-му году лес будзе вырублены.

Практыкаванне

Састаўце праграму, якая вылічвае:

а) праз колькі дзён узровень забруджвання вады адходамі паменшицца ў N разоў, калі кожны дзень ён памяншаецца на $P\%$ (значэнні N і P уводзяцца з клавіятуры);

б) праз колькі дзён колькасць атамаў радыектыўнага ізатопа Ёд-131 паменшицца ў 100 разоў, калі перыяд паўраспаду (час памяншэння колькасці атамаў у 2 разы) складае 8,14 дні;

в) праз колькі дзён паправіцца хворы, г.зн. праз колькі дзён канцэнтрацыя хваробатворных бактэрый у яго крыві паменшицца ад 50 да 12 адзінак. У выніку прыёму лекаў канцэнтрацыя бактэрый штодзень памяншаецца на 20% у параўнанні з папярэднім днём.

8.4. Фізіка

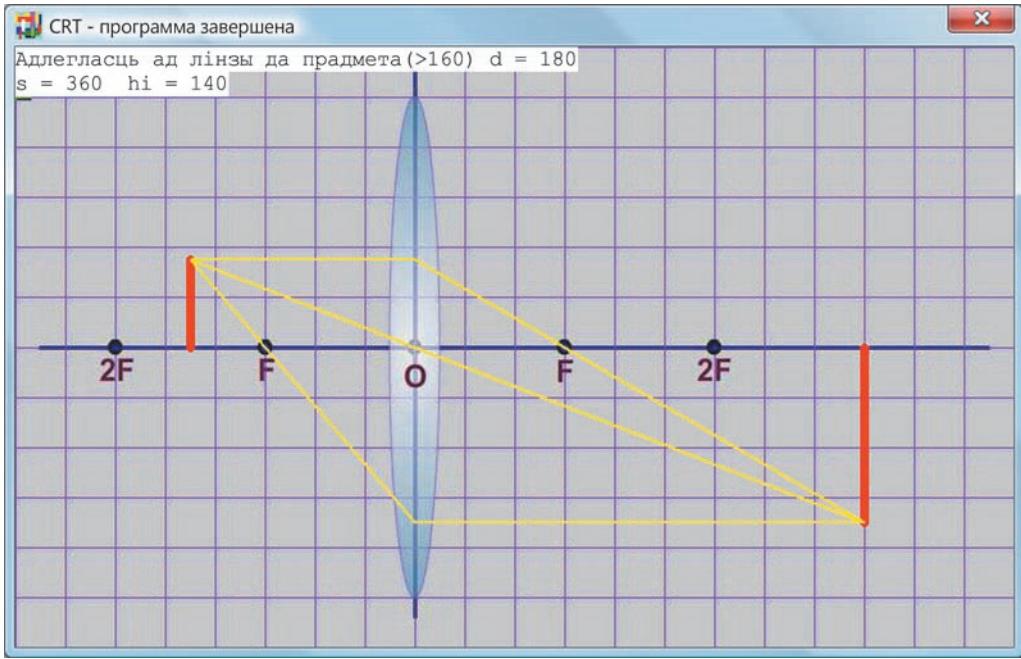
Пабудаванне відарысаў у лінзе. На ўроках фізікі відарысы ў лінзах будуюцца з дапамогай лінейкі і алоўка. Пры гэтым для кожнага новага месца занходжання прадмета прыходзіцца паўтараць усе вылічэнні і пабудаванні. З дапамогай камп'ютарнай праграмы можна лёгка будаваць відарысы, змяняючы адлегласць d ад лінзы да прадмета, а таксама яго вышыню h .

Прыклад 1. Саставіць праграму, якая дэманструе пабудаванне відарыса ў збіральний лінзе (рыс. 2.20).

Загрузім з файла Linza.jpg відарыс збіральний лінзы на фоне каардынатнай сеткі $p:=LoadPicture('Linza.jpg')$.

З дапамогай працэдуры $DrawPicture(p, 0, 0)$ выведзем гэты відарыс у пазіцыю з каардынатамі левага верхняга вугла сеткі $(0, 0)$.

Памеры відарыса будуть адпавядаць памерам графічнага акна 800×480 , адно дзяленне сеткі будзе роўна 40 пікселям. Няхай цэнтр лінзы O мае каарды-



Рыс. 2.20

наты $x_0=320$, $y_0=240$, фокусная адлегласць роўна $F = 120$. На аптычнай восі лінзы адзначаны фокусы і двайнныя фокусы.

Будзем змяшчаць прадмет вышынёй h (напрыклад, адрэзак чырвонага колеру таўшчынёй 7 і вышынёй 70 пікселяў) на адлегласці d ад лінзы: `line(x0-d,y0, x0-d,y0-h)`. Вышыню прадмета будзем змяняць у праграме, а адлегласць d будзем уводзіць з клавіатуры. Межы змянення d і h падбяром так, каб відарыс не выходзіў за межы экрана, напрыклад для $h = 70$ адлегласць $d > 160$.

На падставе формулы тонкай лінзы вылічым адлегласць ад лінзы да відарыса $s := F \cdot d / (d - F)$ і вышыню відарыса $hi := h \cdot s / d$. Акруглім атрыманыя значэнні да цэлых і выведзем на экран.

Пабудуем відарыс — адрэзак `line(x0+s,y0, x0+s,y0+hi)`.

Нарэшце правядзём дапаможныя лініі, якія праходзяць праз фокусы або аптычны цэнтр лінзы і ілюструюць пабудаванне відарыса.

Праграма можа выглядаць так:

```
program Linza;
uses crt, GraphABC;
var p, x0, y0, F, h, d, s, hi: integer;
```

```

begin
    SetWindowSize(800,480);
    p:=LoadPicture('linza.jpg'); DrawPicture(p,0,0); {Лінза}
    x0:=320; y0:=240; F:=120;
    h:=70; {Вышыня прадмета}
    write('Адлегласць ад лінзы да прадмета (>160) d = ');
    readln(d);
    SetPenColor(clRed); SetPenWidth(7);
    line(x0-d,y0, x0-d,y0-h); {Рысаванне прадмета}
    s:=round(F*d/(d-F)); {Разлікі і акругленне}
    hi:=round(h*s/d);
    writeln('s = ',s, ' hi = ',hi); {Вывад значэння s і hi}
    line(x0+s,y0, x0+s,y0+hi); {Рысаванне відарыса}
    SetPenColor(clYellow); SetPenWidth(2);
    line(x0-d,y0-h, x0+s,y0+hi); {Рысаванне ходу праменя}
    line(x0-d,y0-h, x0,y0-h); line(x0,y0-h, x0+s,y0+hi);
    line(x0-d,y0-h, x0,y0+hi); line(x0,y0+hi, x0+s,y0+hi);
end.

```

Результат работы программы показаны на рисунку 2.20.



Найбольш цікавы і зручны для карыстальніка варыянт праграмы можна атрымаць, калі замяніць увод параметраў прадмета з клавіатуры на іх змяненне клавішамі кіравання курсорам.

З гэтай мэтай працэдуры рысавання і разлікі змесцім у цыкл з умовай `repeat...until`. З дапамогай функцыі `ReadKey` будзем счытваць сімвал націснутай клавіши і змяншаць яго ў зменную с тыпу `char`. Пры націсканні клавіши «стрэлка ўверх» (`c=#72`) вышыня відарыса павялічыцца, а пры націсканні клавіши «стрэлка ўніз» (`c=#80`) — паменшыцца. Націсканне клавіши «стрэлка ўлева» (`c=#75`) прывядзе да аддалення прадмета ад лінзы, націсканне клавіши «стрэлка ўправа» (`c=#77`) — да яго набліжэння. Пры націсканні клавіши `Esc` (`c=#27`) работа праграмы будзе завершана.

Мадыфікацыйны фрагмент праграмы можа выглядаць так:

```

h:=70; d:=200; {Параметры прадмета}
Repeat
    DrawPicture(p,0,0); {Вывад відарыса лінзы}
    SetPenColor(clRed); SetPenWidth(7);

```

```

... {Разлікі, рисаванне прадмета, відарыса і ходу праменяў}
c:=ReadKey;
if c=#77 then d:=d+5; {Набліжэнне прадмета}
if c=#75 then d:=d-5; {Аддаленне прадмета}
if c=#72 then h:=h+2; {Павелічэнне вышыні прадмета}
if c=#80 then h:=h-2; {Памяншэнне вышыні прадмета}
Until c=#27;

```



Замест вываду значэнняў s і hi з дапамогай працэдуры `writeln` больш зручна адлюстроўваць параметры прадмета (d, h) і відарыса (s, hi) у загалоўку акна. Для гэтага іх пераўтвараюць у радковы тып і з дапамогай аперацый зліцца змяшчаюць у зменную `str` тыпу `string`:

```

str:='d='+IntToStr(d)+'h='+IntToStr(h)+'s='
+IntToStr(s)+'hi='+IntToStr(hi);
SetWindowCaption(str);

```



Пабудаванне інтэрферэнцыйнай карціны. У курсе фізікі Вы пазнаёміліся са з'явай інтэрферэнцыі. Напомнім, што **інтэрферэнцыяй** называюць з'яву складання кагерэнтных хваль, што прыводзіць да ўтварэння ў просторы ўстойлівой карціны максімумаў і мінімумаў, якія чаргуюцца.

Залежнасць зруху s пункта, што знаходзіцца на адлегласці r ад крыніцы ваганняў, ад часу t апісваецца ўраўненнем хвалі:

$$s(t, r) = A \sin(\varpi t - kr), \text{ где } k = 2\pi/\lambda, \lambda \text{ — даўжыня хвалі.}$$



Прыклад 2. Стварыць праграму, якая дэманструе з'яву інтэрферэнцыі.

Пры пабудаванні любой мадэлі прыходзіцца прымати нейкія дапушчэнні і спрашчэнні. Так, у нашай мадэлі выкарыстоўваюцца кропкавыя крыніцы хваль. Амплітуда ваганняў у межах вобласці назіранняў лічыцца нязменнай.

Для прастаты будзем разглядаць крыніцы ваганняў аднолькавай амплітуды і частаты з нулявой пачатковай фазай у момант часу $t = 0$, якія знаходзяцца ў пунктах з коардынатамі (x_1, y_1) і (x_2, y_2) .

Запішам ураўненні хваль у пункце, якія знаходзіцца на адлегласці r_1 і r_2 ад крыніц ваганняў:

$$s_1(r_1) = A \sin(-2\pi r_1/\lambda) \quad \text{і} \quad s_2(r_2) = A \sin(-2\pi r_2/\lambda).$$

Засталося звязаць амплітуду выніковага вагання з яркасцю пікселяў на экране манітора. Можна выкарыстоўваць шэры колер (яркасць змяняеца ад чорнага

да белага) або колер адной з трох складаючых у колеравай мадэлі RGB. Калі мінімальнаму значэнню зруху паставіць у адпаведнасць чорны колер (уровень = 0), максімальнаму — белы (уровень = 255), сярэдняму — шэры (уровень ≈ 120), то прамежкавыя значэнні будуць адлюстроўвацца рознымі адценнямі шэрага. Мы выкарыстаем чырвоны колер, прыняўшы значэнне 120 за яго сярэднюю яркасць.

Адзначым, што на хуткасць прарысоўкі істотна ўплывае памер графічнага акна і крок змянення каардынат. Зададзім памеры акна 260×300 пікселяў (300 радкоў па 260 пікселяў). Каб правесці разлікі і рысаванне ўсіх пікселяў, выкарыстаем укладзены цыкл. Ва ўнутраным цыклі з параметрам x , значэнні якога змяняюцца ад 0 да 260, будуць выконвацца дзеянні для кожнай крапкі радка. У зовнешнім цыклі з параметрам y , значэнні якога змяняюцца ад 0 да 300, будуць выконвацца дзеянні для ўсіх радкоў. Блок-схема гэтага алгарытму прыведзена на рysунку 2.21.

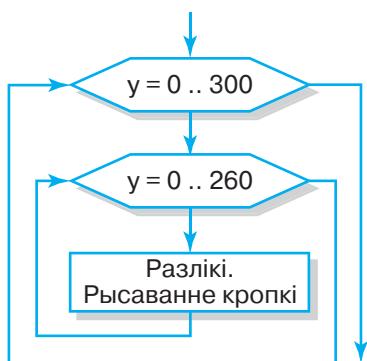


Рис. 2.21

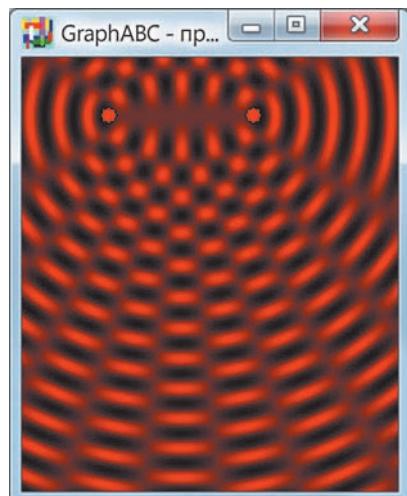


Рис. 2.22

Праграма можа выглядаць так:

```

program interfer;
uses graphABC;
var x, y, x1, y1, x2, y2, s, A, L: integer; r1, r2: real;
begin
    SetWindowSize(260,300);
    SetBrushColor(clRed); {Чырвоны колер}
    x1:=60; y1:=40; x2:=160; y2:=40; {Каардынаты крыніц}
    A:=60; L:=20; {Амплітуда і даўжыня хвалі}

```

```

for y:=0 to 300 do
    for x:=0 to 260 do begin
        {Вылічэнне адлегласцей да крыніц}
        r1:=sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
        r2:=sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
        s:=round(120+A*sin(-2*PI*r1/L)+A*sin(-2*PI*r2/L));
        SetPixel(x,y,rgb(s,0,0));
        {Рысаванне чырвонай кропкі яркасцю s}
    end;
    circle(x1,y1,6);           {Крыніцы хваль}
    circle(x2,y2,6);
end.

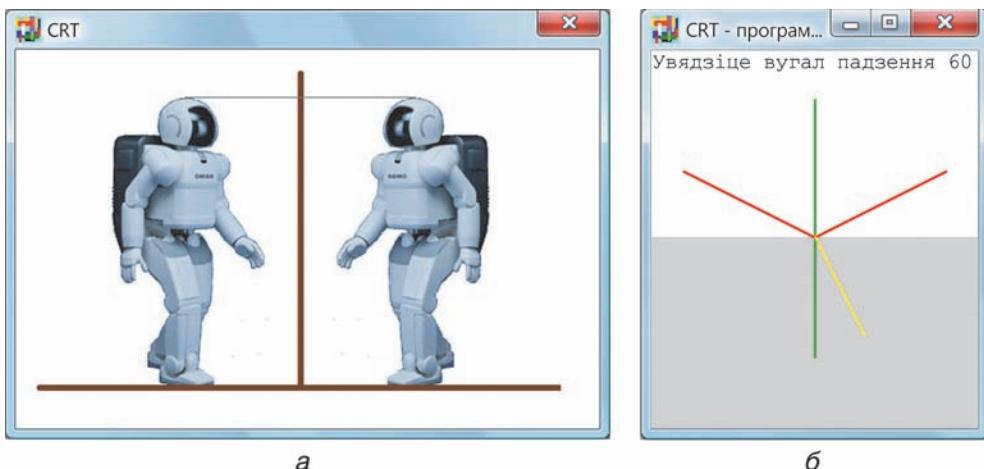
```

Атрыманая інтэрферэнцыйная карціна паказана на рымунку 2.22.

Практыкаванне

Састаўце праграму, якая:

- будзе відарысы прадмета ў плоскім лüstры (адлегласць ад прадмета да лüstра ўводзіцца з клавіятуры);
- дэманструе адбіццё і праламленне святла на плоскай паверхні (вугал падзення ўводзіцца з клавіятуры).



РАЗДЗЕЛ 3

АПРАЦОЎКА ІНФАРМАЦЫІ Ў СІСТЭМЕ КІРАВАННЯ БАЗАМІ ДАНЫХ

§ 9. Базы данных і сістэмы кіравання базамі даных

У сучасным свеце колькасць інфармацыі ўвесь час нарастае. Для ўпрадкаванага захоўвання і апрацоўкі вялікіх аб'ёмаў інфармацыі складаюцца і выкарыстоўваюцца базы даных.

База данных — гэта сукупнасць узаемазвязаных і арганізаваных пэўным чынам даных, што адлюстроўваюць стан аб'ектаў і адносін паміж імі ў якой-небудзь предметнай вобласці.

Першымі камп'ютарнымі базамі даных, якія стварыў чалавек, былі таблічныя базы даных, дзе размяшчалася ў асноўным лікавая інфармацыя. Затым пачалося асваенне чалавекам тэкстовых баз даных. З павышэннем хуткадзейння камп'ютараў і павелічэннем іх зневажнай і аператуўнай памяці пачалі стварацца і выкарыстоўвацца графічныя і мультымедыйныя базы даных.

Інфармацыя, якая захоўваецца ў базах даных, павінна быць структуравана, г. зн. арганізавана пэўным чынам (рыс. 3.1), актуальна і даступна карыстальніку.



Рыс. 3.1

Фактычна база данных, напоўненая інфармацыяй, з'яўляецца файлам.

Сярод існуючых мадэлей арганізацыі інфармацыі ў базах даных найбольшаяе распаўсюджанне атрымала **рэляцыйная мадэль** (ад англ. *relation* — зносіны).

Правообладатель Народная асьвета

У гэтай мадэлі даныя арганізаваны ў выглядзе сукупнасці табліц, паміж якімі ўсталяваны сувязі. Прызначэнне такіх сувязей і парадак іх стварэння мы разгледзім у § 11.

Сістэма кіравання базамі даных (СКБД) — гэта сукупнасць праграмных і моўных сродкаў, прызначаных для стварэння, захоўвання і апрацоўкі баз даных.

У цяперашні час спецыялістамі распрацавана каля сотні розных СКБД. Усе яны могуць быць падзелены па *спосабе доступу да баз даных* на дзве вялікія групы: *настольныя і серверныя*.

Настольныя СКБД арыентаваны на абслугоўванне аднаго карыстальніка, які працуе на пэўным камп'ютары з базамі даных у кожны рэальны момант часу. Да настольных СКБД адносяць Microsoft Access, Paradox, dBase, FoxPro.

Недахопы настольных СКБД сталі відавочнымі, калі ўзнікла неабходнасць адначасовай работы з імі вялікай колькасці карыстальнікаў. Таму наступным этапам развіцця СКБД стала стварэнне серверных СКБД.

Серверныя СКБД выкарыстоўваюць прынцып цэнтралізаванага захоўвання і апрацоўкі даных, які заснаваны на архітэктуры кліент-сервер. СКБД, што захоўвае даныя, і прыкладная праграма, што інтэрпрэтуюць гэтыя даныя, з'яўляюцца рознымі дадаткамі, г. зн. існуюць дадатак-сервер і дадатак-кліент. Да серверных СКБД адносяць Microsoft SQL Server, Informix, Sybase, DB2, Oracle і інш.

У гэтым раздзеле мы разгледзім СКБД Access 2003, якая атрымала шырокое прыменение ў нашай краіне.

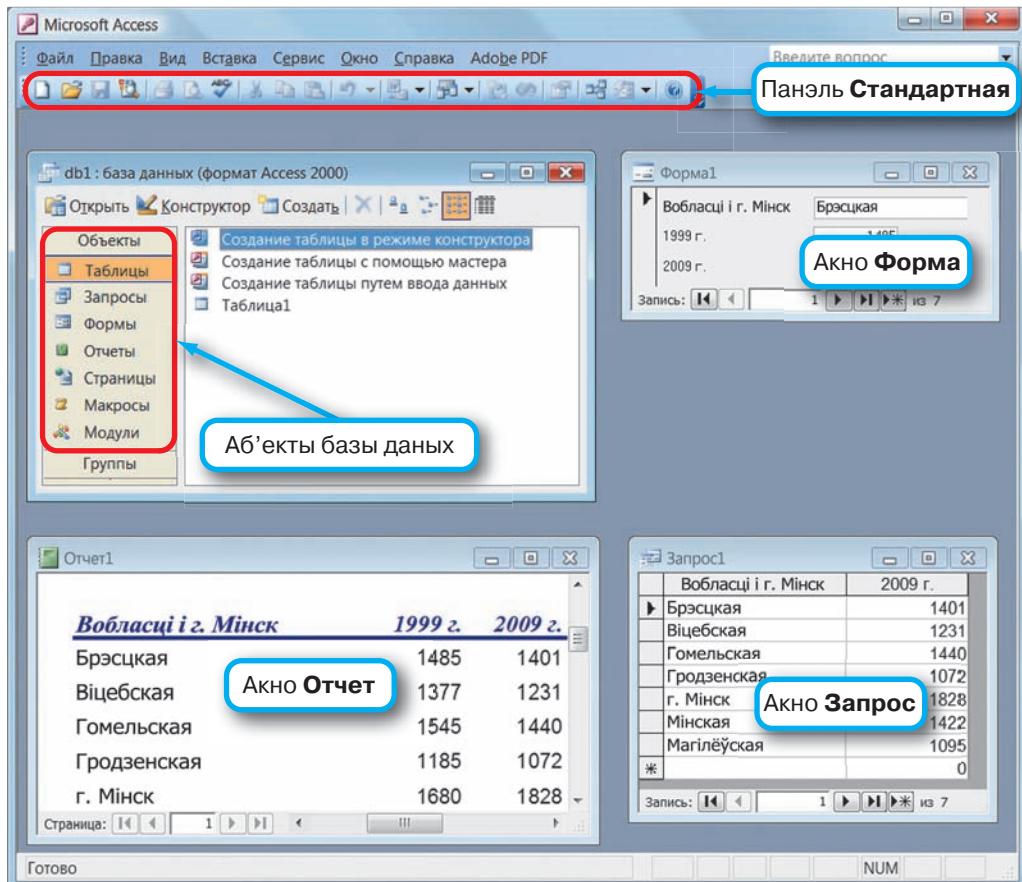
Праграма Access пасля яе ўсталявання можа загружанца з Рабочага стала двайнай пстрычкай мыши па ярлыку  або праз сістэму меню: **Пуск →**

→ Программы → Microsoft Office → Microsoft Access.

Пасля запуску СКБД Access з дапамогай каманды **Файл → Открыть** можна адкрыць базу даных. Апішам асноўныя элементы інтэрфейса акна, што адкрыеца (рыс. 3.2).

На панэлі **Стандартная** змяшчаюцца асноўныя інструменты, якія выкарыстоўваюцца пры работе з базай даных. Некаторыя інструменты гэтай панэлі аналагічны інструментам панэлі **Стандартная** тэкставага рэдактара Word. Частка інструментаў прызначана выключна для работы з базай даных. Выкарыстанне гэтых інструментаў мы разгледзім у далейшым.

У акне **База данных** размешчаны яе аб'екты: **Таблицы, Запросы, Формы, Отчеты** і інш. Усе гэтыя аб'екты, калі яны ствараліся, захоўваюцца ў агульным



Рыс. 3.2

файле базы данных на дыску з пашырэннем .mdb. Работа з кожным з аб'ектаў ажыццяўляецца ў сваім акне, як паказана на рымсунку 3.2.

Разгледзім прызначэнне кожнага аб'екта.

Аб'ект **Таблицы** з'яўляецца асноўным аб'ектам базы даных. Ён прызначаны для захоўвання даных. На аснове табліцы (рыс. 3.3) ствараюцца астатнія аб'екты базы даных.

Як правіла, для решэння складаных задач адной табліцы не хапае. Таму ў базе даных звычайна прысутнічаюць адразу некалькі табліц, звязаных паміж сабой.

Запіс рэлЯцыйнай базы даных уяўляе сабой радок табліцы, у якім сабрана інфармацыя аб адным аб'екце базы даных. У табліцы на рымсунку 3.3 такіх записаў сем.

Імёны палёў			
	Вобласці і г. Мінск	1999 г.	2009 г.
► Запіс 1	Брэсцкая	1485	1401
Запіс 2	Віцебская	1377	1231
Запіс 3	Гомельская	1545	1440
Запіс 4	Гродзенская	1185	1072
Запіс 5	г. Мінск	1680	1828
Запіс 6	Мінская	1559	1422
Запіс 7	Магілёўская	1214	1095
*		0	0

Рыс. 3.3

Поле рэляцыйнай базы даных — гэта слупок табліцы, які змяшчае значэнні пэўнай уласцівасці аб аб'ектах базы даных. У табліцы на рисунку 3.3 трывалі: Вобласці і г. Мінск, 1999 г., 2009 г.

Для апісання поля выкарыстоўваюцца наступныя характеристыстыкі: імя, тып, памер, фармат даных поля.

У розных палях табліцы могуць захоўвацца даныя розных тыпаў: тэкставыя, лікавыя, лагічныя і інш. Але ў адным полі табліцы змяшчаюцца даныя толькі аднаго тыпу. Кожная табліца мае сваю вызначаную структуру.

Больш поўнае апісанне тыпаў палёў прыведзена ў § 10.

Аб'ект **Запросы** дае магчымасць адбору даных з табліц на падставе пэўных умоў. На рисунку 3.2 паказаны запыт з іменем **Запрос 1**.

Аб'ект **Формы** адлюстроўвае даныя з табліц або запытаў. З дапамогай форм зручна ўводзіць даныя ў табліцы. На рисунку 3.2 паказана форма з іменем **Форма 1**.

Аб'ект **Отчеты** прызначаны для стварэння дакумента, які адлюстроўвае інфармацыю, адабраную з базы даных. Справаздачу можна праглядзець на экране або раздрукаваць на прынтары. На рисунку 3.2 паказана справаздача з іменем **Отчет 1**.



1. Што называюць базай даных?
2. Што называюць запісам і полем у табліцы базы даных?
3. Якія сістэмы называюць сістэмамі кіравання базамі даных?
4. Якія аб'екты прысутнічаюць у акне **База данных MS Access**? Для чаго прызначаны гэтыя аб'екты?

§ 10. Стварэнне табліцы базы даных

10.1. Праектаванне базы даных

Вывучэнне магчымасцей СКБД Access пачнём з праектавання базы даных. Вылучаюць наступныя этапы стварэння базы даных:

- стварэнне табліц базы даных, якое ўключае распрацоўку структуры кожнай табліцы і ўвод у гэтыя табліцы неабходнай інфармацыі;
- вызначэнне сувязей паміж некалькімі табліцамі.

Прыклад. Спраектаваць базу даных «Бібліятэка» (файл Library book.mdb), якая змяшчае звесткі аб выдачы кніг з бібліятэкі карыстальніка.

Асноўнай задачай пры карыстанні базай даных «Бібліятэка» з'яўляецца ад-сочванне выдачы кніг усім кліентам.

Прааналізаўшы неабходную для захоўвання інфармацыю, паспрабуем спачатку размясціць яе ў адной табліцы, палі якой падзелім на тры групы: «Звесткі аб кліентах бібліятэкі», «Звесткі аб кнігах» і «Звесткі аб выдачы кніг». Табліца базы даных у гэтым выпадку павінна мець наступную структуру:

Імя поля	Тып даных	
Прозвішча	Тэксты	Звесткі аб кліентах бібліятэкі
Імя	Тэксты	
Адрес	Тэксты	
Тэлефон	Тэксты	
Электронная пошта	Тэксты	
Аўтар кнігі	Тэксты	Звесткі аб кнігах
Назва кнігі	Тэксты	
Год выдання	Лікавы	
Кошт кнігі	Грошовы	Звесткі аб выдачы кніг
Дата выдачы	Дата/час	
Адзнака аб звароце	Лагічны	

Працаваць з такой табліцай даволі нязручна. Напрыклад, пры выдачы некалькіх кніг аднаму кліенту будзе неабходна шматразова паўтараць інфармацыю аб ім: прозвішча, імя, адрес і г.д., што прывядзе да неапрайданага павелічэння

памеру табліцы і можа павысіць імавернасць з'яўлення памылак пры ўводзе інфармацыі.

Для павышэння эфектыўнасці работы са ствараемай базай падзелім адну табліцу на тры: «Кліенты», «Кнігі», «Выдача кніг». Апішам структуру кожнай табліцы.

Табліца «Кліенты»

Імя поля	Тып даных
Код кліента	Лічыльнік
Прозвішча	Тэкставы
Імя	Тэкставы
Адрес	Тэкставы
Тэлефон	Тэкставы
Электронная пошта	Тэкставы

Табліца «Кнігі»

Імя поля	Тып даных
Код кнігі	Лікавы
Аўтар кнігі	Тэкставы
Назва кнігі	Тэкставы
Год выдання	Лікавы
Кошт кнігі	Грашовы

Табліца «Выдача кніг»

Імя поля	Тып даных
Код выдачы	Лікавы
Код кліента	Лікавы
Код кнігі	Лікавы
Дата выдачы	Дата/час
Адзнака аб звароце	Лагічны

У табліцы «Кліенты» выкарыстоўваецца тып даных **Лічыльнік** (**Счетчик**). Ён прымяняецца для захоўвання цэльых лікавых значэнняў, якія Access павяліч-

Правообладатель Народная асвета

вае пры пераходзе да кожнага новага запісу. Лічыльнік можа выкарыстоўвацца ў якасці ўнікальнага ідэнтыфікатора запісу табліцы, у якой няма другой велічыні такога тыпу. У нашым выпадку **Код кліента** будзе ўнікальным ідэнтыфікаторам, які дазволіць лёгка адрозніваць аднаго кліента ад другога.

СКБД Access дазваляе таксама выкарыстоўваць тыпы даных, апісаныя ў наступнай табліцы.

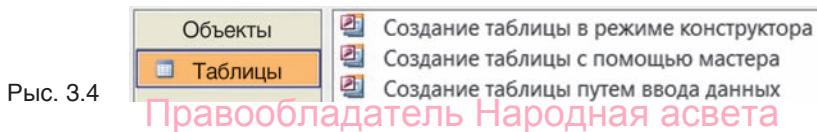
Тып даных	Апісанне
Лікавы (Числовой)	Поле лікавага тыпу, якое змяшчае адвольныя лікавыя значэнні
Тэкставы (Текстовый)	Поле тэкставага тыпу, якое змяшчае тэкставыя, лікавыя даныя, якія не патрабуюць вылічэння. Даўжыня поля не перавышае 255 сімвалаў
Дата/час (Дата/время)	Поле, якое змяшчае дату і час у розных фарматах
Грашовы (Денежный)	Поле, якое мае лікавыя грашовыя значэнні
Лагічны (Логический)	Поле, якое змяшчае лагічныя значэнні
Гіперспасылка (Гиперсылка)	Поле-спасылка на некаторы дакумент або файл
Аб'ект OLE (Объект OLE)	Поле, якое змяшчае дакументы розных тыпаў (дакументы Word, табліцы Excel, кропкавыя рыsunкі і інш.)
Поле МЕМО	Поле тэкставага тыпу, у якім размяшчаюцца даныя вялікага аб'ёму. Даўжыня поля не перавышае 65 535 сімвалаў

Перад стварэннем табліц у СКБД Access неабходна стварыць новую базу даных. Гэта можна зрабіць з дапамогай паслядоўнасці камандаў: **Файл** → **Создать** → **Новая база данных** → у акне **Файл новой базы данных** пазначыць імя файла базы, напрыклад Library book.

СКБД Access дазваляе ствараць структуру табліцы трymа спосабамі: у рэжыме **Конструктора**, з дапамогай **Мастера** або шляхам уводу даных (рэжым **Таблицы**) (рыс. 3.4).

У рэжыме **Таблицы** даныя ўводзяцца ў пустую табліцу.

Разгледзім больш падрабязна работу рэжыма **Мастэр і Конструктор**.



10.2. Стварэнне структуры табліцы

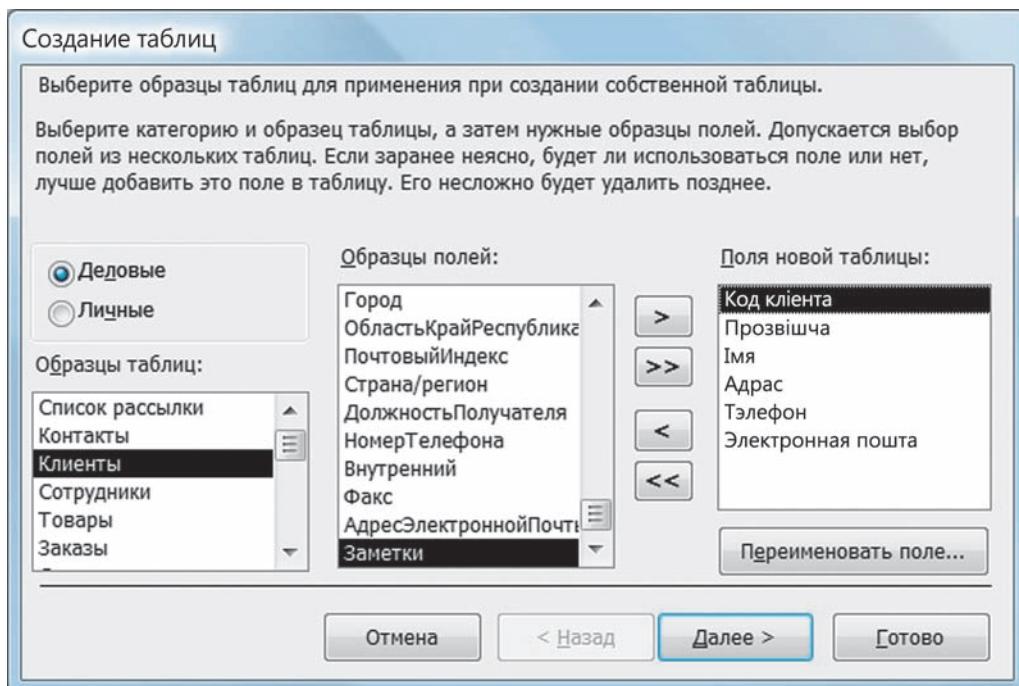
У рэжыме **Мастера** зручна ствараць структуру табліцы, у якой выкарыстоўваюцца стандартныя імёны палёў і тыпы даных у гэтых палях. **Мастэр** спачатку прануе выбраць узор табліцы, а затым адабраць палі для гэтай табліцы.

Прыклад 1. Стварыць у рэжыме **Мастера** структуру табліцы «Кліенты».

Выберам аб'ект **Таблицы** ў акне **База данных** і двойчы пstryкнем мышшу па радку **Создание таблицы с помощью мастера**.

Выберам узор табліцы **Кліенты**. Узоры палёў перамесцім з дапамогай кнопкі з акна **Образцы полей** у акно **Поля новой таблицы**, як паказана на рэсунку 3.5. Імёны перанесеных палёў перакладзём на беларускую мову з дапамогай кнопкі **Переименовать поле**. Для пераходу да наступнага кроку пstryкнем мышшу па кнопкы **Далее >**.

Зададзім імя табліцы, напрыклад **Кліенты**, і ўключым флагок **Пользователь определяет ключ самостоятельно**. Націснем кнопкку **Далее**. Выберам поле з унікальнымі для кожнага запісу данымі, для нашага прыкладу — гэта поле **Код клиента**.



Рыс. 3.5

Клиенты : таблица						
	Код клиента	Прозвішча	Імя	Адрес	Телефон	Электронная пошта
	(Счетчик)					
Запись: 1 из 1						

Рис. 3.6

ента. Націснем кнопку **Готово**. У вінку адкрыеца акно табліцы «Кліенты», готовай для запаўнення інфармацыяй (рыс. 3.6).

Пstryкнуўши мышшу па кнопкы **Конструктор** **Конструктор** (гл. рис. 3.2), праглядзім структуру створанай з дапамогай **Мастера** табліцы «Кліенты» (рыс. 3.7).

Ключевое поле		
	Имя поля	Тип данных
	Код клиента	Счетчик
	Прозвішча	Текстовый
	Імя	Текстовый
	Адрес	Текстовый
	Телефон	Текстовый
	Электронная пошта	Текстовый

Рис. 3.7

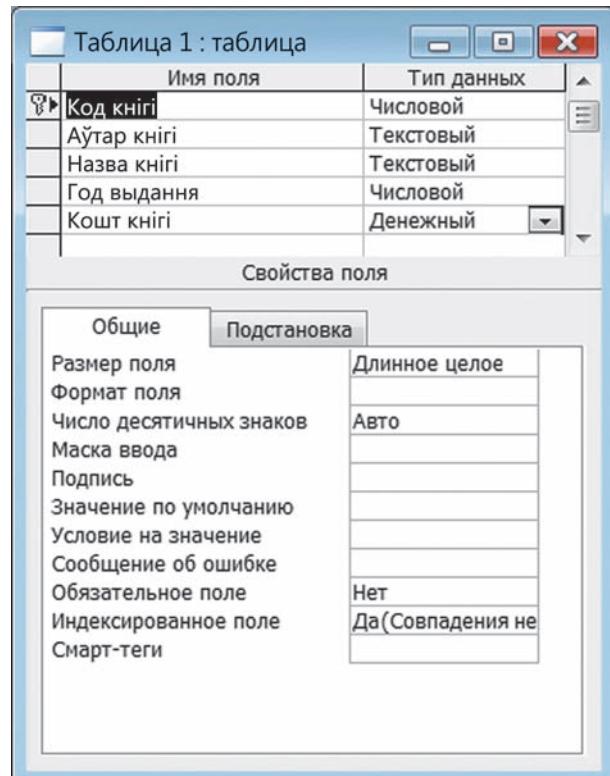
Пажадана, каб кожная табліца мела **ключ** — адно або некалькі палёў, зменшыя якіх унікальнае для кожнага запісу. Для табліцы «Кліенты» такім з'яўляецца поле **Код клиента**. Тып даных гэтага поля — **Лічыльнік**. У нашай табліцы кожны кліент будзе мець унікальны нумар і ніякія два запісы не будуць аднолькавымі.

Мастер вызначае тып даных палёў самастойна ў залежнасці ад назвы без папярэдняга вызначэння гэтых тыпаў карыстальнікам. У далейшым у рэжыме **Конструктора** тып поля можа быць зменены.

Рэжым **Конструктора** дае карыстальніку магчымасць самастойна апісаць і змяніць структуру табліцы.

Прыклад 2. Стварыць у рэжыме **Конструктора** структуру табліцы «Кнігі».

Двойной пstryчкай мыши па радку **Создание таблицы в режиме конструктора** (гл. рис. 3.4) адкрыем акно **Таблица 1**, у якім неабходна апісаць палі.



Рыс. 3.8

Табліца «Кнігі» будзе ўтримліваць пяць палёў. Набяром для кожнага з іх імя і тып даных. У ніжнай частцы акна вызначым уласцівасці поля, як паказана на рысунку 3.8.

Пры ўводзе тыпаў даных і ўласцівасцей палёў будзем карыстацца спіскамі, што раскрываюцца.

У табліцы «Кнігі» ключавым з’яўляецца поле **Код кнігі**. Каб гэта поле стала ключавым, дастаткова ў рэжыме **Конструктора** змясціц курсор у выбранае поле і націснуць кнопкі **Ключевое поле** на панэлі акна **База данных** або выкананаць каманду **Правка → Ключевое поле**. Калі неабходна вызначыць адразу некалькі ключавых палёў, то трэба вылучыць патрэбныя радкі і націснуць кнопкі **Ключевое поле**. Паўторныя аналагічныя дзеянні адмяняюць прымету ключавога поля.

Скончыўшы апісанне табліцы, захаваем яе з дапамогай паслядоўнасці камандаў: **Файл → Сохранить как →** пазначым імя табліцы, напрыклад «Кнігі» → **OK**.

Правообладатель Народная асвета

Клиенты : таблица						
	Код клиента	Прозвішча	Імя	Адрес	Тэлефон	Электронная почта
▶	1	Сідараў	Васіль	вул. Пляханава, д. 20	2954321	Sidor@tut.by
▶	2	Сімончык	Валянціна	вул. Васняцова, д. 3	2961736	sim@mail.ru
▶	3	Цітоў	Аляксей	вул. Жылуновіча, д. 6	2474331	Titov@tut.by
▶	4	Смірноў	Аляксандар	вул. Ванеева, д. 6	2912345	mal@tut.by
▶	5	Іваноў	Павел	вул. Кашавога, д. 13	2916754	Pivan@mail.ru

Рыс. 3.9

Створаныя намі табліцы «Кліенты» і «Кнігі» дабаўляюцца ў акно **База данных**.

Табліца «Выдача кніг» таксама можа быць створана ў рэжыме **Конструктора**.

Стварыўши структуру табліцы, карыстальнік можа ў рэжыме **Таблицы** прыступіць да яе запаўнення. Для гэтага ў акне **База данных** трэба двойчы пstryкнуць мышшу па назве табліцы, напрыклад «Кліенты».

Увод даных у табліцу і іх рэдагаванне ажыццяўляюцца непасрэдна ў ячэйках табліцы (рыс. 3.9). Дзеянні па ўводзе і рэдагаванні аналогічны адпаведным дзеянням у MS Excel.

Для выдалення цэлага запісу ў табліцы неабходна ў рэжыме **Таблица** паставіць курсор у любое поле выдалімага запісу і пstryкнуць мышшу па кнопкы **Удалить запись** . Для выдалення адразу некалькіх запісаў неабходна спачатку іх вылучыць. Пры выдаленні цэлага запісу ў табліцы «Кліенты» поле **Код клиента** губляе ўнікальны нумар кліента для гэтага запісу ў табліцы.

Дабаўленне запісаў у табліцу ажыццяўляецца ў рэжыме **Таблица**. Пры гэтым новы запіс размяшчаецца ў канцы табліцы.

Пасля заканчэння ўводу даных у табліцу або іх рэдагавання табліца захоўваецца з дапамогай каманды **Файл → Сохранить**.

- ?
- Што неабходна для праектавання базы даных?
 - Якія рэжымы выкарыстоўваюцца для стварэння структуры табліцы?
 - Што разумеецца пад ключавым полем у табліцы? Для чаго выкарыстоўваецца ключавое поле?
 - Якія тыпы даных можна выкарыстоўваць у СКБД Access?
 - Для чаго выкарыстоўваецца тып даных Лічыльнік?

Практыкаванні

1. Стварыце базу даных «Гарады Магілёўскай вобласці» (файл MogilevCity.mdb), якая ўтрымлівае табліцу «Гарады_Магілёўскай_вобласці».

Гарады_Магілёўскай_вобласці : таблица				
	Гарады вобласці	Насельніцтва (тыс. чал.)	Асноўныя рэкі	Заснаванне горада
▶	Бабруйск	219	Бярэзіна	1387
	Быхаў	16,1	Дняпро	1430
	Горкі	34	Проня	1544
	Клімавічы	14,5	Калініца	1581
	Крычаў	26,8	Сож	1136
	Магілёў	372	Дняпро	1267
	Мсціслаў	11,5	Віхра	1135
	Асіповічы	33,6	Свіслач	1872
	Слаўгарад	8,2	Сож	1136
	Чавусы	10,4	Проня	1581
	Шклов	17,8	Дняпро	1535
*				
Запись: [◀] [◀] 1 [▶] [▶] * из 11				

2. Стварыце базу даных «Бібліятэка» (файл Library book.mdb), якая ўтрымлівае тры табліцы: «Кліенты» (гл. рыс. 3.9), «Кнігі», «Выдача кніг». Узоры афармлення табліц «Кнігі» і «Выдача кніг» прыведзены на наступных рысунках:

Кнігі : таблица					
	Код кнігі	Аўтар кнігі	Назва кнігі	Год выдання	Кошт кнігі
+	17	В. Быкаў	Збор твораў. Т. 3	2001	15 000,00р.
♂	31	Я. Маўр	Палескія рабінзоны	1998	10 000,00р.
+	67	Ж. Верн	Плавучый остров	1990	18 000,00р.
+	105	Л. Толстой	Война и мир	2004	7 000,00р.
+	203	А. Кристи	Собр. сочинений. Т. 10	2008	6 000,00р.
+	465	Н. Кун	Легенды и мифы	2002	9 000,00р.

Выдача кніг : таблица					
	Код выдачи	Код клиента	Код кнігі	Дата выдачи	Адзнака аб звароце
▶	1	4	203	19.02.2009	<input checked="" type="checkbox"/>
	2	5	17	20.02.2009	<input checked="" type="checkbox"/>
	3	3	31	21.02.2009	<input checked="" type="checkbox"/>
	4	2	67	24.02.2009	<input checked="" type="checkbox"/>
	5	5	465	13.03.2009	<input type="checkbox"/>
	6	1	203	15.03.2009	<input type="checkbox"/>
	7	1	105	24.04.2009	<input type="checkbox"/>



3. Створыце базу даных «Экзамены» (файл Mark.mdb), якая ўтрымлівае чатыры табліцы: «Настаўнікі», «Вучні», «Прадметы», «Экзамен». Узоры афармлення табліц прыведзены на наступных рэсунках:

Настаўнікі : таблица

	КодНастаўніка	Прозвішча	Імя	Імя па бацьку	Адрес
+	1	Абрыкосаў	Андрэй	Андрэевіч	г. Пінск, вул. Сідарэнкі, д. 5, кв. 10
▶	2	Ламусава	Вера	Васільёўна	в. Сноба, Пінскі р-н, д. 16
+	3	Пястроў	Ігар	Віктаравіч	г. Пінск, вул. Міра, д. 34, кв. 1
*	(Счетчик)				

Запись: [◀◀] [◀] [▶] [▶▶] [*] из 3

Вучні : таблица

	КодВучня	Прозвішча	Імя	Імя па бацьку	Клас	Адрес
▶	1	Акуліч	Святлана	Ігараўна	11 А	г. Пінск, вул. Прыгожая, д. 20
+	2	Ананіч	Андрэй	Іванавіч	11 А	г. Пінск, вул. Народная, д. 16, кв. 2
+	3	Біруроў	Павел	Сідаравіч	11 Б	г. Пінск, вул. Пляханава, д. 5, кв. 32
+	4	Брусава	Ганна	Іванаўна	11 Б	г. Пінск, вул. Васняцова, д. 20, кв. 65
+	5	Булгакава	Кіра	Янаўна	11 А	г. Пінск, вул. Васняцова, д. 2, кв. 16
*	(Счетчик)					

Запись: [◀◀] [◀] [▶] [▶▶] [*] из 5

Прадметы : таблица

	КодНастаўніка	КодПрадмета	Назва
▶		1	1 Беларуская літаратура
		1	2 Беларуская мова
		2	3 Геаграфія
		3	4 Біялогія
*	(Счетчик)		

Запись: [◀◀] [◀] [▶] [▶▶] [*] из 4

Экзамен : таблица

	КодВучня	КодПрадмета	Адзнака	Дата здачи
		1	9	06.06.2009
		1	8	08.06.2009
		1	7	10.06.2009
		1	8	12.06.2009
*	(Счетчик)			

Запись: [◀◀] [◀] [▶] [▶▶] [*] из 5

§ 11. Звязванне табліц базы даных

Устанаўленне сувязей у MS Access дае магчымасць аўтаматычна злучаць даныя з розных табліц. Так забяспечваецца цэласнасць базы даных.

Працэс устанаўлення сувязей паміж табліцамі называюць **пабудаваннем Схемы даных**.

Для ўстанаўлення сувязей паміж дзвюма табліцамі неабходна вызначыць у кожнай з іх палі для звязвання. **Гэтыя палі могуць мець аднолькавыя або розныя імёны, але павінны змяшчаць даныя аднаго тыпу!**

Разгледзім сувязь паміж табліцамі «Супрацоўнікі» (рыс. 3.10) і «Ведамасць заработка платы» (рыс. 3.11) у базе даных Salary.mdb. Структура кожнай з гэxих табліц паказана на рысунку 3.12.

	Табельны нумар	Прозвішча	Імя	Імя па бацьку	Адрес
▶ +	1	Алтухоў	Павел	Сямёновіч	вул. Нікрасава, 23
▶ +	2	Арцёмкіна	Ала	Леанідаўна	вул. Пляханава, д. 3, кв. 5
▶ +	3	Белавус	Аляксандр	Аляксандравіч	пр. Партызанскі, д. 43, кв. 1
▶ +	4	Буцік	Алена	Яўгенаўна	вул. Бельскага, д. 34, кв. 67
▶ +	5	Вяроўка	Пётр	Алегавіч	вул. Пешаходная, д. 23
▶ +	6	Кровель	Ірына	Васільеўна	вул. Народная, д. 13, кв. 65
▶ +	7	Кішмяк	Сяргей	Уладзіміравіч	вул. Пляханава, д. 46, кв. 78
▶ +	8	Рогава	Зоя	Пятроўна	пас. Азёрны, д. 6

Запись: [◀◀] [◀] [▶] [▶▶] * из 8

Рыс. 3.10

	Код выдачи	Табельны нумар	Дата выдачи	Памер	Падатак
▶ +	1	1	1	04.09.2009	780 000,00р.
▶ +	2	2	2	04.09.2009	820 000,00р.
▶ +	3	3	3	05.09.2009	910 000,00р.
▶ +	4	4	4	04.09.2009	560 000,00р.
▶ +	5	5	5	06.09.2009	435 000,00р.
▶ +	6	6	6	04.09.2009	740 000,00р.
▶ +	7	7	7	06.09.2009	680 000,00р.
▶ +	8	8	8	05.09.2009	420 000,00р.

Запись: [◀◀] [◀] [▶] [▶▶] * из 8

Рыс. 3.11

Правообладатель Народная асвета

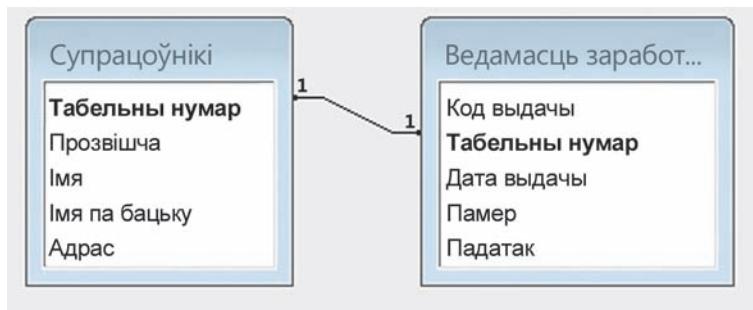


Рыс. 3.12

У табліцы «Супрацоўнікі» ключавым з’яўляецца поле **Табельны нумар**. Да-дзенае поле мае тып Лічыльнік і ўтрымлівае ўнікальныя значэнні для кожнага за-пису гэтай табліцы. Назавём гэта поле *першасным ключом*.

Калі ў табліцы «Ведамасць заработка» кожны супрацоўнік можа быць запісаны толькі адзін раз, то ключавым полем гэтай табліцы таксама можа з’яў-ляцца поле **Табельны нумар**. У табліцы «Ведамасць заработка» ключавое поле **Табельны нумар** будзем называць *знешнім ключом*.

У гэтым выпадку сувязь паміж першасным і знешнім ключамі, назы-ваюць сувяззю *адзін да аднаго*. Гэты тып сувязі прадстаўлены на рысун-ку 3.13.



Рыс. 3.13

Калі ў табліцы «Ведамасць заработка» адзін супрацоўнік можа быць запісаны некалькі разоў, то поле **Табельны нумар** ужо не будзе ўнікальным, па-колькі можа паўтарацца шматразова. Тады тып даных у гэтым полі можа пры-маць лікавыя значэнні, а ключавым полем з унікальнымі значэннямі можа быць вызначана поле **Код выдачи**.

У гэтым выпадку сувязь паміж аднайменнымі палямі **Табельны нумар** у абе-дзвюх табліцах называюць сувяззю *адзін да многіх*. Гэты тып сувязі паказаны на рисунку 3.14.



Рыс. 3.14

Прыклад. Пабудаваць Схему даных для базы даных «Бібліятэка», як паказана на рысунку 3.15.

Адкрыем многатаблічную базу даных «Бібліятэка», для якой будуць устанаўлівацца сувязі паміж табліцамі.

Пstryкнем мышшу па значку **Схема данных**  (гл. рыс. 3.2).

У акне **Добавление таблицы** вылучым першую табліцу, для якой устанаўліваецца сувязь. У нашым прыкладзе — гэта табліца «Кліенты».

Пstryкнем мышшу па кнопкы **Добавить**. На экране будзе паказана структура табліцы, якую мы дабавілі. Затым такім жа чынам дабавім у Схему даных астатнія табліцы для ўстанаўлення сувязей («Выдача кніг», «Кнігі») і пstryкнем мышшу па кнопкы **Закрыть**.

У акне табліцы «Кліенты» пstryкнем мышшу па полі, якое будзе выкарыстоўвацца для ўстанаўлення сувязі, напрыклад **Код кліента**, і перасунем яго на аднайменнае поле табліцы «Выдача кніг».

У акне **Изменение связей** пераканаемся, што звязаны неабходныя палі, і ўключым флажок **Обеспечение целостности данных**. У выніку выкананых дзеянняў у акне **Схема данных** два звязаныя палі злучацца лініяй, як паказана на рысунку 3.15.



Рыс. 3.15

Падобным чынам звязкам табліцы «Выдача кніг» і «Кнігі» па полі **Код кнігі**.

Для захавання створаных сувязей пstryкнем мышшу па кнопкы **Сохранить** і закрыем акно **Схема данных**. Пры закрыці акна **Схема данных** сувязі паміж табліцамі захоўваюцца.

Створаныя сувязі ў далейшым можна выдаліць у акне **Схема данных**, вылучыўши пstryчкай мышы лінію сувязі і націснуўши клавішу Delete (Выдаліць). Двойная пstryчка мышшу па лініі сувязі адкрывае акно **Изменение связей**, у якім сувязі могуць быць адредагаваны.

- ?**
- 1. Для чаго неабходна звязваць табліцы?
- 2. Якімі ўласцівасцямі павінны валодаць палі, якія звязваюцца?
- 3. Якія тыпы сувязей паміж палямі табліц існуюць?

Практиканне

Установіце сувязі паміж табліцамі баз даных, прапанаваных настаўнікам:

- а) база даных «Бібліятэка» (файл Libgagu book.mdb) з табліцамі «Кліенты», «Выдача кніг», «Кнігі»;
- б) база даных «Экзамены» (файл Mark.mdb) з табліцамі «Наставнікі», «Вучні», «Прадметы», «Экзамен».





§ 12. Стварэнне і запаўненне формы

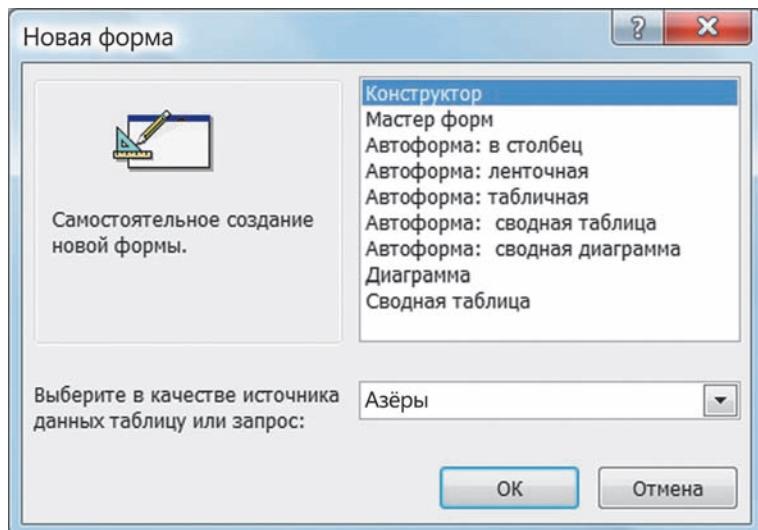
Вы ўжо ведаецце, што карыстальнік уводзіць даныя табліцы непасрэдна ў яе ячэйкі. Калі яго не задавальняе таблічны спосаб размяшчэння даных, ён можа падрыхтаваць **форму**. У форме карыстальнік размяшчае даныя на экране камп'ютара ў парадку, зручным яму для работы. У створанай форме можна ўводзіць, рэдагаваць і праглядаць даныя.

Формы можна ствараць на аснове гатовай табліцы або запыту. Разгледзім, як ствараюцца формы на аснове табліц.

СКБД Access пропануе тры спосабы стварэння форм: **Создание формы в режиме мастера**, **Создание формы в режиме конструктора** і **Автоформа**.

Пры падрыхтоўцы формы неабходна ў акне **База данных** выкананаць каманду **Формы → Создать** або націснуць кнопкі **Создать**.

У акне **Новая форма**, што адкрыеца, пропануюцца два асноўныя спосабы **Конструктор** і **Мастер форм**, а таксама **Автоформы: в столбец**, **ленточная**, **табличная** і інш. (рыс. 3.16). Хуткае стварэнне форм забяспечваюць **Мастер форм** і **Автоформы**.



Рыс. 3.16

Разгледзім на прыкладзе работу з **Мастером форм**.

Прыклад 1. Стварыць з дапамогай **Мастера форм** па даных табліцы «Гароды_Магілёўскай_вобласці» стужачную форму са стылем **Сонцестояние**, якая змяшчае даныя, паказаныя на рэсунку 3.17.

Правообладатель Народная асвета

Гарады_Mагілёўскай_вобласці

Гарады вобласці	Насельніцтва (тыс. чал.)	Заснаванне горада
Бабруйск	219	1387
Быхаў	16,1	1430
Горкі	34	1544
Клімавічы	14,5	1581
Крычаў	26,8	1136

Запись: | < < | 2 | > > | * | из 11 | < < | III | > > |

Рыс. 3.17

Создание форм

Выберите поля для формы.
Допускается выбор нескольких таблиц или запросов.

Таблицы и запросы

Таблица: Гарады_Mагілёўскай_вобласці

Доступные поля:

Выбранные поля:

Асноўныя рэкі

Гарады вобласці
Насельніцтва
Заснаванне горада

Отмена < Назад Далее > Готово

Рыс. 3.18

Правообладатель Народная асвета

Для стварэння формы з дапамогай **Мастера форм** двойчы пstryкнем мышшу па радку **Создание форм с помощью мастера** ў акне **База данных**. **Мастер форм** у некалькі этапаў адкрые акно **Создание форм**, у якім мы зададзім імя табліцы для формы і выберем доступныя палі, як паказана на рэсунку 3.18, а затым вызначым зневідимыя выгляд формы: выберем табліцу «Гарады_Магілёўскай_вобласці» і трэ палі **Гарады вобласці**, **Насельніцтва (тыс. чал.)**, **Заснаванне горада**, зневідимыя выгляд формы — **Ленточны**.

У далейшым у акне, што адкрыеца, выберем стыль формы **Солнцестояние**, зададзім імя формы і пstryкнем мышшу па кнопкы **Готово**.

У акне **Создание форм** можна адкрыць гатовую форму для ўводу даных.

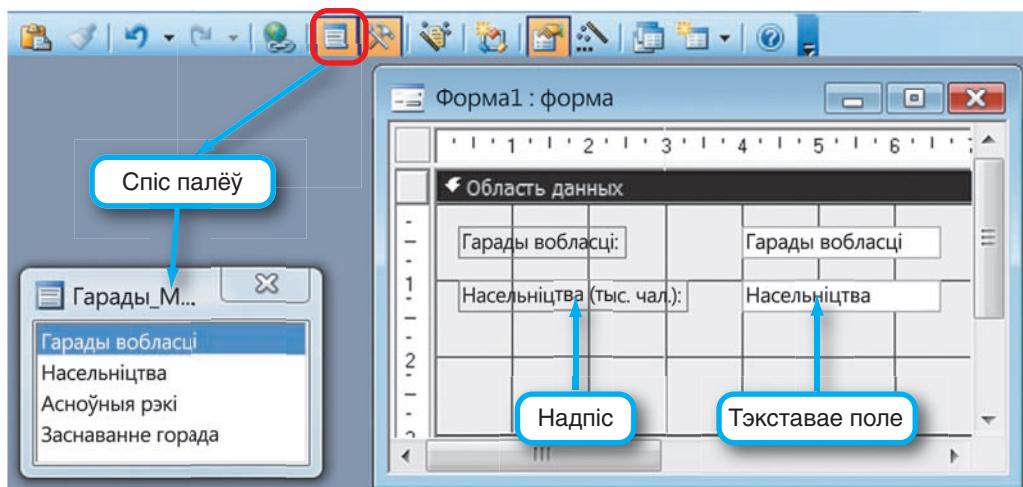
Заўважым, што, ствараючы форму, з табліцы можна пераносіць не ўсе палі. Пры запаўненні формы даныя аўтаматычна размяшчаюцца ў ячэйках табліцы, на аснове якой дадзеная форма стваралася.

Стварэнне формы з дапамогай **Конструктора** з'яўляецца больш складаным працэсам, чым яе падрыхтоўка з дапамогай **Мастера форм**.

Прыклад 2. На аснове табліцы «Гарады_Магілёўскай_вобласці» стварыць форму ў рэжыме **Конструктора**.

Для гэтага ў акне **База данных** выберем пункт **Формы**. Затым націснем кнопкы **Создать** меню гэтага акна. У акне **Новая форма** выберем пункт меню **Конструктор**, а таксама крыніцу даных — табліцу «Гарады_Магілёўскай_вобласці», на аснове якой будзеца форма.

У выніку выкананых дзеянняў адкрыоцца акно для канструявання формы і спецыяльнае акно са спісам палёў табліцы (рыс. 3.19).



Рыс. 3.19

Перанясём з дапамогай мышы са спіса палёў у **Область данных** неабходныя палі табліцы, напрыклад **Гарады вобласці**, **Насельніцтва (тыс. чал.)**, **Заснаванне горада**.

Пры пераносе кожнага поля ў **Область данных** у ім размяшчаюцца **Надпіс** (імя поля) і **Тэкставае поле** (даныя).

Скончым канструктуванне формы яе захаваннем.

Формы, створаныя з дапамогай **Мастера форм**, могуць быць зменены ў рэжыме **Конструктора**.



- Чаму карыстальнікі для ўводу даных аддаюць перавагу стварэнню форм замест выкарыстання табліц?
- Якія спосабы стварэння формы пропануе праграма Access?

Практыкаванне

Адкрыйце базу даных, пропанаваную настаўнікам. Стварыце формы па гато-вых табліцах, выкарыстаўшы **Мастер форм** або **Конструктор**:

а) для базы даных «Цеплавыя электрастанцыі Беларусі» (файл Ener.mdb), якая змяшчае табліцу «Электрастанцыі» са звесткамі аб буйных цеплавых электрастанцыях нашай краіны;

Табліца «Электрастанцыі»

Электрастанцыі : таблица			
Электрастанцыя	Магутнасць (тыс. кВт)	Паліва	Удзельны расход паліва (г/кВт*г)
Лукомльская ГРЭС	2500	мазут, прыродны газ	318
Бярозаўская ГРЭС	950	мазут	360
Мінская ТЭЦ-4	900	мазут, прыродны газ	235
Наваполацкая ТЭЦ	505	мазут	220
Мінская ТЭЦ-3	395	прыродны газ, мазут	245
Гомельская ТЭЦ-2	360	прыродны газ, мазут	236
Marijëuskaja TЭЦ-2	345	мазут	254
Светлагорская ТЭЦ	270	спадарожны газ	313

Форма «Электрастанцыі»

Электрастанцыі		Магутнасць (тыс. кВт)
Электрастанцыя		
Лукомльская ГРЭС		2500
Бярозаўская ГРЭС		950
Мінская ТЭЦ-4		900
Наваполацкая ТЭЦ		505

б) для базы даных «Чырвоная кніга Беларусі» (файл тіг.mdb), якая змяшчае табліцу «Чырвоная кніга Беларусі» са звесткамі аб рэдкіх відах жывёльнага свету нашай краіны.

Табліца «Чырвоная кніга Беларусі»

Чырвоная кніга Беларусі : таблица					
Код	Клас	Атрад	Сямейства	Від	Відарыс
1	Птушкі	Совападобныя	Сіпухавыя	Сіпуха	Bitmap Image
2	Млекакормячыя	Парнакапытныя	Пустарогія	Еўрапейскі зубр	Bitmap Image
3	Млекакормячыя	Драпежныя	Куніцавыя	Барсук	Bitmap Image
4	Млекакормячыя	Драпежныя	Кацінья	Рысь еўрапейская	Bitmap Image
5	Птушкі	Буслападобныя	Буслінья	Чорны бусел	Bitmap Image

Запісъ: [Назад] [Падваж] 2 [Далей] [Наступ] из 5

Форма «Млекакормячыя і птушкі»

Млекакормячыя і птушкі : форма

Від: Сіпуха

Відарыс:



Запісъ: [Назад] [Падваж] 1 [Далей] [Наступ] из 6

§ 13. Пошук даных з дапамогай запытаў

Пошук інфармацыі ў базах даных выконваецца праз запыты. З дапамогай **запыту** СКБД Access выбірае і адлюстроўвае наборы запісаў з табліц базы даных, якія задавальняюць пэўныя ўмовы.

Запыт можа фарміравацца на аснове адной або некалькіх звязаных табліц або запытаў, пабудаваных раней.

СКБД Access падтрымлівае стварэнне запытаў з дапамогай **Мастера і Конструктора**. **Мастэр** забяспечвае магчымасць стварэння простых запытаў.

Разгледзім, як ствараюцца запыты на аснове табліц. Выкарыстаем для гэтага табліцу базы даных «Акварыумныя рыбкі», паказаную на рэсунку 3.20.

Акварыумныя рыбкі : таблица

Назва рыбкі	Месца паходжання	Даўжыня (см)
Гамбузія	Паўднёвая Амерыка	5
Пляціецыліус чорны	Мексіка	4
Зялёны мечаносец	Мексіка	11
Тэтра-фон-рыба	Паўднёвая Амерыка	2
Бархатны гірардынус	Паўднёвая Амерыка	5
Макрапод	Кітай	9
Гурамі жамчужны	Індыйя	10
Ляліус	Індыйя	5
Неон	Паўднёвая Амерыка	4

Запись: [◀] [◀] 1 [▶] [▶] * из 9

Рыс. 3.20

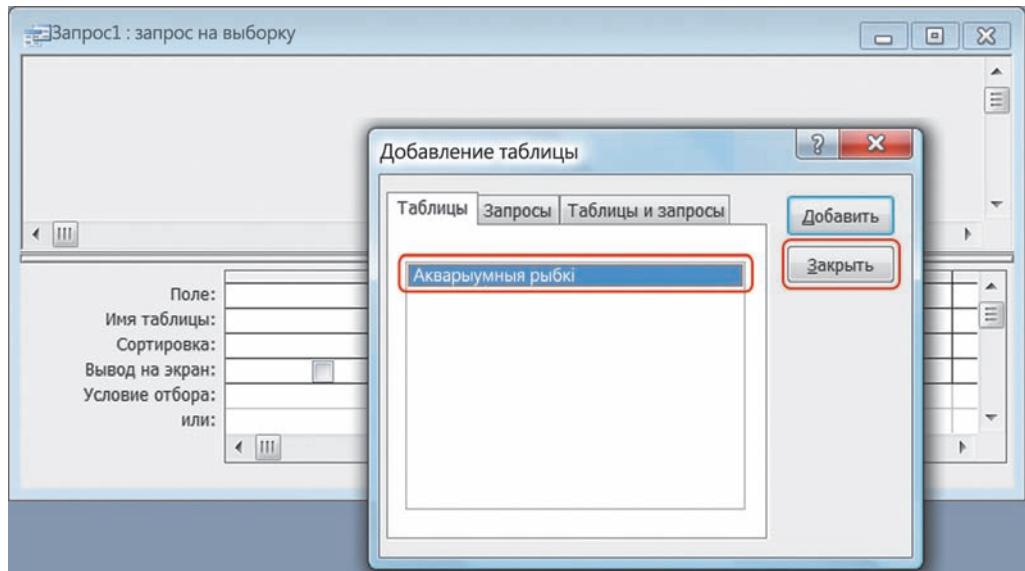
Прыклад 1. Стварыць з дапамогай **Конструктора** прости запыт, па якім будуць прадстаўлены звесткі аб назвах рыбак і месцах іх паходжання.

Пстрыкнем мышшу па кнопкы **Запросы** ў акне **База данных**.

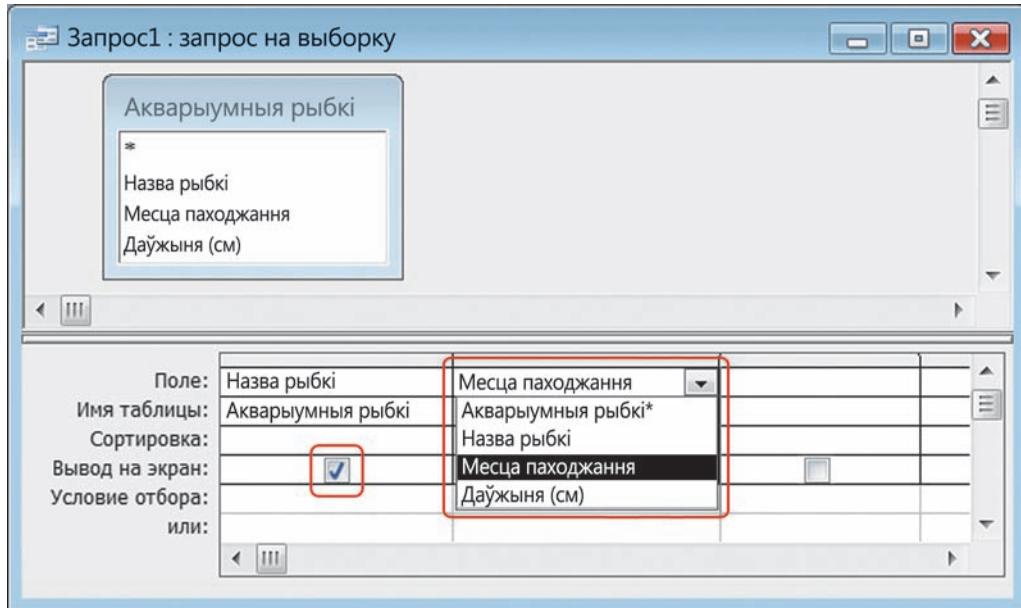
Двойчы пстрыкнем мышшу па радку **Создание запроса в режиме конструктора** або выканаем каманду **Создать → Конструктор**.

У выніку адкрыюцца два акны, як паказана на рысунку 3.21.

У акне **Добавление таблицы** выберам табліцу з іменем «Акварыумныя рыбкі» і пстрыкнем па кнопкы **Добавить**, а затым — па кнопкы **Закрыть**.



Рыс. 3.21



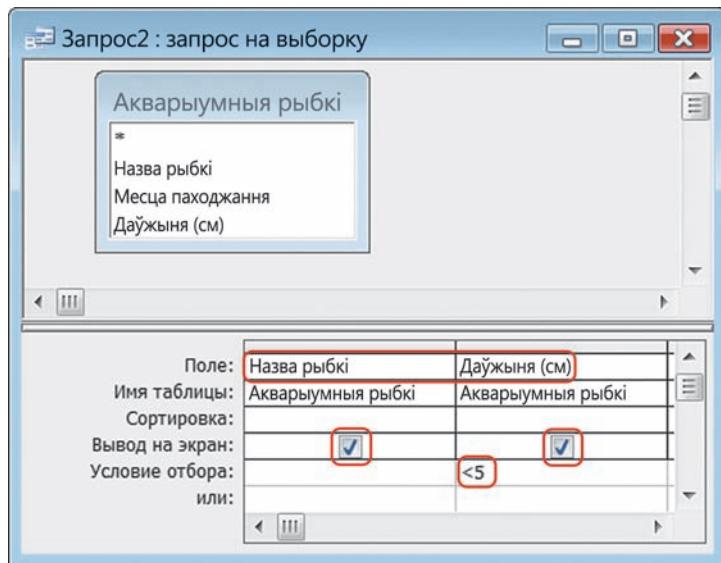
Рыс. 3.22

У акне **Запрос на выборку** (рыс. 3.22), рухаючыся злева направа, у выпадающих спісках радка **Поле** выберам палі для дабаўлення ў запыт: **Назва рыбкі**, **Месца паходжання**.

Устаноўка флажка ў кожным адабраным слупку радка **Вывод на экран** дазволіць вывесці патрэбныя слупкі. Пасля заканчэння канструйвання запыту захаваем яго. Рэзультатам выканання запыту будзе новая часовая табліца (рыс. 3.23).

Запрос1 : запрос на выборку	
▶ Гамбузія	Паўднёвая Амерыка
▶ Пляціпецыліус чорны	Мексіка
▶ Зялёны мечаносец	Мексіка
▶ Тэтра-фон-рыя	Паўднёвая Амерыка
▶ Бархатны гірардынус	Паўднёвая Амерыка
▶ Макрапод	Кітай
▶ Гурамі жамчужны	Індыя
▶ Ляліус	Індыя
▶ Неон	Паўднёвая Амерыка

Рыс. 3.23



Рыс. 3.24

Прыклад 2. Стварыць запыт, па якім будуць прадстаўлены звесткі аб рыбках, даўжыня якіх меншая за 5 см.

Адкрыем у рэжыме **Конструктора** новае акно для запыту. У радку **Условие отбора** апішам запыт, як паказана на рэсунку 3.24.

Запрос2 : запрос на выборку		
	Назва рыбкі	Даўжыня (см)
▶	Пляціпецыліус чорны	4
	Тетра-фон-рыа	2
	Неон	4

Рыс. 3.25

табліц з крыніцамі даных трэба пазначыць будуць выкарыстоўвацца ў запыце.

Рэзультатам выканання запыту будзе новая часовая табліца (рыс. 3.25).

Найбольш складаным з'яўляецца падрыхтоўка запытаў адразу па некалькіх звязаных табліцах. Правцэс стварэння такіх запытаў нагадвае фарміраванне запыту па адной табліцы. Аднак на этапе адбору

адразу некалькі табліц, даныя з якіх

- 1. Для чаго выкарыстоўваюцца запыты?
- 2. Як будуецца прости запыт з дапамогай **Конструктора**?
- 3. Якія магчымасці пралануе карыстальніку **Конструктор** для стварэння запыту на выбарку з умовамі?

Практыкаванні

1. Адкрыйце табліцу базы даных «Акварыумныя рыбкі» і стварыце запыты, паказаны ў прыкладах 1 і 2 гэтага параграфа.

2. Адкрыйце табліцу базы даных «Цеплавыя электрастанцыі Беларусі» і стварыце па ёй:

а) прости запыт, які змяшчае запісы палёў Электрастанцыя, Магутнасць (тыс. кВт), Паліва;

б) запыт, які змяшчае запісы з палямі Электрастанцыя і Магутнасць (тыс. кВт) < «400 тыс. кВт»;

в) запыт з запісамі Электрастанцыя і Паліва = «мазут»;

г) запыт з запісамі Электрастанцыя і Паліва = «мазут» или Паліва = «спадарожны газ».



3. Адкрыйце базу даных «Бібліятэка» і стварыце з дапамогай Конструктора запыт, паказаны на рэсунку.

Прагледзьце табліцу, пабудаваную па гэтым запыце.

Поле:	Прозвішча	Імя	Назва кнігі	Дата выдачы	Адзнака аб звароце
Імя таблицы:	Кліенты	Кліенты	Кнігі	Выдача кніг	Выдача кніг
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:					
или:					
	<input type="button" value=" <"/>	<input type="button" value=" >"/>			

§ 14. Сартаванне запісаў у табліцы

Для эфектыўнага пошуку інфармацыі ў вялікіх табліцах базы даных, у якіх могуць захоўвацца сотні і тысячи запісаў, у карыстальніка часта ўзнікае неабходнасць размясціць запісы ў пэўнай паслядоўнасці, г. зн. адсартаваць іх.

Пад **сартаваннем** запісаў у табліцы базы даных разумеюць працэс іх упрадавання ў пэўнай паслядоўнасці па значенні аднаго з палёў.

У залежнасці ад тыпу даных, прызначаных для сартавання, усе запісы ў табліцы базы даных могуць сартавацца:

- па велічыні ліку, калі тып даных лікавы;
- па алфавіце, калі тып даных тэкставы (сімвальны);
- па дате і часе, калі даныя ў полі змяшчаюць значэнні даты і часу.

Для сартавання запісаў па даных канкрэтнага поля неабходна паставіць курсор у любы радок адпаведнага слупка і пstryкнучы мышшу па адной з кнопкі на панэлі **Стандартная**:



— сартаванне па нарастанні;



— сартаванне па спаданні.

Прадэманструем працэс сартавання запісаў у табліцы «Вулканы», фрагмент якой паказаны на рыйсунку 3.26.

Прыклад. У табліцы «Вулканы» змясціца запісы поля **Назва** ў алфавітным парадку.

Змесцім курсор мышы ў любым радку поля **Назва**. Пstryкнем мышшу па кнопкы **Сортировка по возрастанию** або выканаем паслядоўнасць камандаў: **Записи → Сортировка → Сортировка по возрастанию**.

Вынік сартавання для прапанаванага фрагмента паказаны на рыйсунку 3.27.

Вулканы : таблица					
	Назва	Кантынент	Месцазнаходжанне	Вышыня над узроўнем мора	Актыўнасць
▶	Этна	Еўропа	Сіцилія	3340	<input checked="" type="checkbox"/>
	Везувій	Еўропа	Аленіны	1277	<input type="checkbox"/>
	Гекла	Еўропа	Ісландыя	1491	<input checked="" type="checkbox"/>
	Меру	Афрыка	Танзанія	4567	<input checked="" type="checkbox"/>
	Керынчы	Азія	Хансю	3805	<input checked="" type="checkbox"/>
	Кракатау	Азія	Камчатка	813	<input type="checkbox"/>
	Семеру	Азія	Мінданао	3676	<input checked="" type="checkbox"/>

Рыс. 3.26

Вулканы : таблица					
	Назва	Кантынент	Месцазнаходжанне	Вышыня над узроўнем мора	Актыўнасць
▶	Везувій	Еўропа	Аленіны	1277	<input type="checkbox"/>
	Гекла	Еўропа	Ісландыя	1491	<input checked="" type="checkbox"/>
	Керынчы	Азія	Хансю	3805	<input checked="" type="checkbox"/>
	Кракатау	Азія	Камчатка	813	<input type="checkbox"/>
	Меру	Афрыка	Танзанія	4567	<input checked="" type="checkbox"/>
	Семеру	Азія	Мінданао	3676	<input checked="" type="checkbox"/>
	Этна	Еўропа	Сіцилія	3340	<input checked="" type="checkbox"/>

Рыс. 3.27



1. Для чаго выкарыстоўваецца сартаванне даных?
2. Які працэс называюць сартаваннем запісаў табліцы базы даных?
3. Што неабходна зрабіць для выканання сартавання запісаў у табліцы базы даных?

Практыкаванні

1. Адкрыйце табліцу «Кліенты» базы даных «Бібліятэка», прапанаваную настаўнікам, выканайце сартаванне запісаў гэтай табліцы, змясціўшы прозвішчы кліентаў у алфавітным парадку.
2. Адкрыйце базу даных «Гарады Магілёўскай вобласці» (MogilevCity.mdb), што змяшчае аднайменную табліцу. Выканайце сартаванне запісаў гэтай табліцы:
 - а) запісы поля **Гарады вобласці** размясціце ў алфавітным парадку;
 - б) запісы поля **Насельніцтва (тыс. чал.)** упарадкуйце па нарастанні.
3. Стварыце базу даных «Рэкі Беларусі», фрагмент табліцы якой паказаны на рэсунку.

Рэкі : таблица

Назва	Плошча басейна (кв. км)	Даўжыня (км)	Густата рачной сеткі (км/кв. км)
Леваць	21900	536	0,45
Дняпро	504000	2201	0,39
Вілія	25100	498	0,44
Заходні Буг	39400	772	0,42
Сож	42140	648	0,38
Прыпяць	121000	761	0,42
Бярэзіна	24500	613	0,35
Нёман	98200	937	0,47
► Захадняя Дзвіна	37900	1020	0,45

Выканайце сартаванне запісаў гэтай табліцы:

- а) запісы поля **Даўжыня (км)** упарадкуйце па нарастанні;
- б) запісы поля **Густата рачной сеткі (км/кв. км)** размясціце ў парадку спадання.

§ 15. Стварэнне справаздач

Справаздачы з'яўляюцца зручным і гібкім спосабам прагляду і раздрукоўкі выніковых звестак з базы даных.

Пад **справаздачай** разумеюць дакумент, змест якога фарміруеца па вызначанным запыце на аснове інфармацыі, размешчанай у базе даных.

У справаздачах даныя падаюцца ў зручным выглядзе. Яны могуць быць адсортаваны, асобым чынам згрупаваны, а таксама могуць змяшчаць выніковыя значэнні і г. д.

Правообладатель Народная асвета

СКБД Access пропонує кілька способів створення справа-
здач: **Автоотчет**, **Мастер отчетов** і **Конструктор**. Справа-
здачі з'яўляюцца самастойнимі аб'ектамі базы даних.

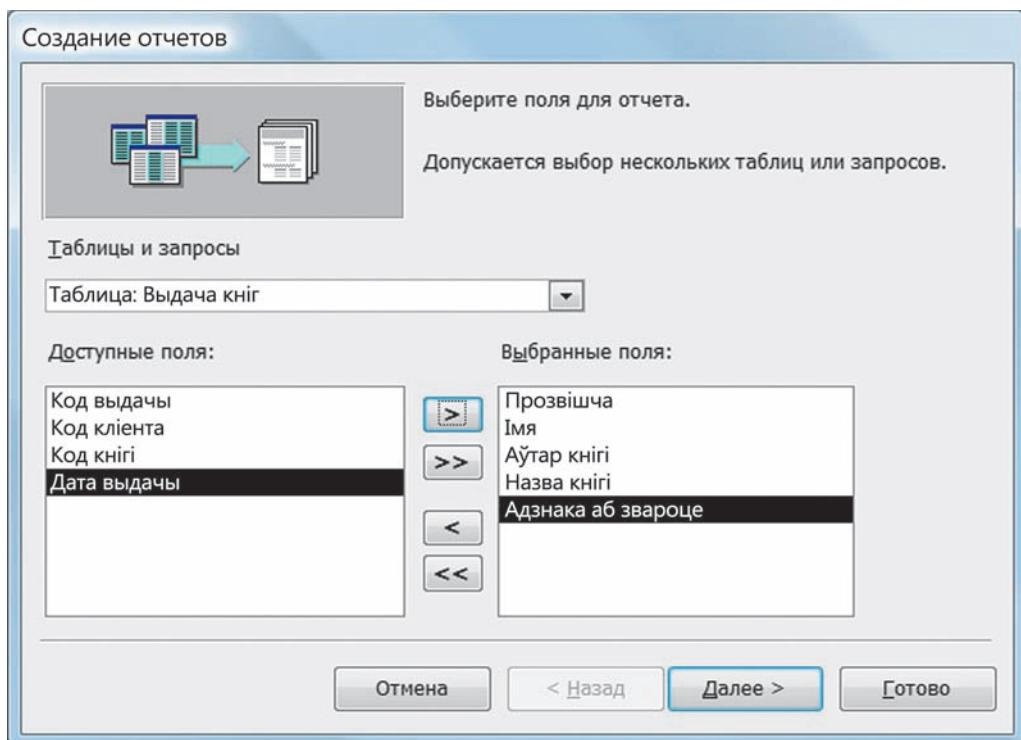
Пасля прагляду атрыманай справа-здачі яе можна захаваць або раздрукуваць.

Разгледзім на прыкладзе многатаблічнай базы даних «Бібліятэка», як з дапамогай **Мастера** ствараючыя справа-здачі.

Прыклад. Стварыць справа-здачу, якая змяшчае звесткі з трох табліц базы даних «Бібліятэка» (гл. § 10). З табліцы «Кліенты» выбраць імя і прозвішча кліента, з табліцы «Кнігі» — аўтара і назыву кнігі, а з табліцы «Выдача кніг» — адзнаку аб звароце кнігі.

Адкрыем базу даних «Бібліятэка» і двойчы пстрыкнем мышшу па радку **Создание отчета с помощью мастера** ў акне **База данных**.

У акне **Создание отчетов**, па чарзе актывізуючы назвы табліц базы даних у спіску **Таблицы и запросы**, перанясём патрэбныя палі з акна **Доступные поля** ў акно **Выбранные поля**, як паказана на рисунку 3.28.



Рыс. 3.28

Правообладатель Народная асвета

Прозвічча	Імя	Аўтар книгі	Назва книгі	Адзнака аб звароце
Смірноў	Аляксандар	А. Кристи	Собр. сочинений. Т. 10	<input checked="" type="checkbox"/>
Іваноў	Павел	В. Быкаў	Збор твораў. Т. 3	<input checked="" type="checkbox"/>
Цітоў	Аляксей	Я. Маўр	Палескія рабінзоны	<input checked="" type="checkbox"/>
Сімончык	Валянціна	Ж. Верн	Плавучый остров	<input checked="" type="checkbox"/>
Іваноў	Павел	Н. Кун	Легенды и мифы	<input type="checkbox"/>
Сідараў	Васіль	А. Кристи	Собр. сочинений. Т. 10	<input type="checkbox"/>
Сідараў	Васіль	Л. Толстой	Война и мир	<input type="checkbox"/>

Страница: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 <span style="border: 1px solid

2. Адкрыйце табліцу базы даных «Гарады Магілёўскай вобласці», прапанаваную настаўнікам, і стварыце справаздачу з дапамогай **Мастера**. Выгляд фрагмента справаздачы паказаны на рэсунку.

Гарады Магілёўскай вобласці	
Гарады вобласці	Бабруйск
Насельніцтва (тыс. чал.)	219
Заснаванне горада	1387
Гарады вобласці	Быхаў
Насельніцтва (тыс. чал.)	16,1
Заснаванне горада	1430

ІНФАРМАЦЫЙНЫЯ СІСТЭМЫ І ТЭХНАЛОГII

§ 16. Інфармацыйныя сістэмы

Чалавек сустракаеца з рознымі паводле свайго прызначэння сістэмамі ў по-быце, на вытворчасці, у адукцыі, медыцыне і іншых сферах дзейнасці.

Слова «сітэма» азначае *цэлае, складзенае з частак*.

Сістэмы існуюць у жывой і нежывой прыродзе. Прыкладам складанай сістэмы ў нежывой прыродзе з'яўляецца Сонечная сістэма, элементы якой — Сонца, планеты і іх спадарожнікі, мноства камет і астэродаў. Прыкладам біялагічнай сістэмы з'яўляюцца раслінныя клеткі.

Сістэмы складаюцца з розных элементаў: планет, рэк, раслін, жывёл, людзей, тэхнічных устроўстваў, дэталей механізмаў, інфармацыйных рэурсаў, матэматычных ураўненняў.

Мы часта сустракаем такія слова і словазлучэнні, як біясітэма, экалагічная сістэма, дзяржаўная сістэма, сістэма навучання, сістэма сацыяльнага забеспечэння, сістэма аховы здароўя і г. д.

У курсе інфарматыкі мы ўжо знаёміліся з рознымі сістэмамі: аперацыйнымі, файлавымі, сістэмамі кіравання базамі даных.

Даданае да паняцця «сістэма» слова «інфармацыйная» паказвае на яе асноўнае функцыянальнае прызначэнне — работа з інфармацыяй.

Інфармацыйная сістэма (ІС) — цэласная сістэма ўзаемазвязаных сродкаў і метадаў захоўвання, апрацоўкі, пошуку і распаўсюджвання інфармацыі, што абслугоўваеца і выкарыстоўваеца чалавекам.

Існуюць розныя класіфікацыі інфармацыйных сістэм: па функцыянальным прызначэнні, па вобласці прымянення, па ўзоруні аўтаматызацыі.

Па вобласці прымянення, напрыклад, вылучаюць наступныя ІС: бібліятэчныя, прававыя, эканамічныя, банкаўскія, маркетынгавыя, медыцынскія, метэаралагічныя і інш.

Сярод ІС існуе вялікі клас інфармацыйна-пошукавых (даведачных) сістэм (ІПС), якія прызначаны для пошуку інфармацыі. Пошук і адбор інфармацыі ў ІПС ажыццяўляецца карыстальнікам па зададзеных крытэрыях або ўмовах — запытах. Запыты карыстальніка ў ІПС будуюцца звычайна на натуральнай для чалавека мове, напрыклад беларускай, рускай, англійскай.

Многія ІПС у цяперашні час даступныя ў сетцы Інтэрнэт. Яны аператыўна забяспечваюць карыстальніку доступ да велізарнай колькасці электронных інфар-

мацыйных рэурсаў. Аглед інфармацыйных рэурсаў і інфармацыйных сістэм Беларусі размешчаны на сایце <http://infores.mpt.gov.by>. З некаторымі з іх Вы ўжо знаёміліся пры вывучэнні інфарматыкі.

Наогул, праблемам распрацоўкі і эфектуўнага выкарыстання нацыянальных інфармацыйных рэурсаў у нашай краіне надаецца вялікая ўвага на дзяржаўным узроўні. На стварэнне нацыянальнай аўтаматызаванай інфармацыйнай сістэмы, асноўнай задачай якой з'яўляеца фарміраванне адзінага нацыянальнага інфармацыйнага рэурса, накіравана дзяржаўная праграма «Электронная Беларусь». Для спрашчэння ўзаемадзеяння грамадзян з аўтаматызаванымі інфармацыйнымі сістэмамі дзяржаўных органаў укараняеца праграмны комплекс «Адноакно».

Нацыянальны прававы інтэрнэт-партал Рэспублікі Беларусь <http://pravo.by> забяспечвае доступ да **Эталонного банка данных правовой информации**. Ён уяўляе сабой аўтаматызованую поўнатэкставую інфармацыйна-пошукавую сістэму па заканадаўстве Рэспублікі Беларусь, якая дазваляе хутка знайсці неабходную інфармацыю (рыс. 4.1).

Рыс. 4.1

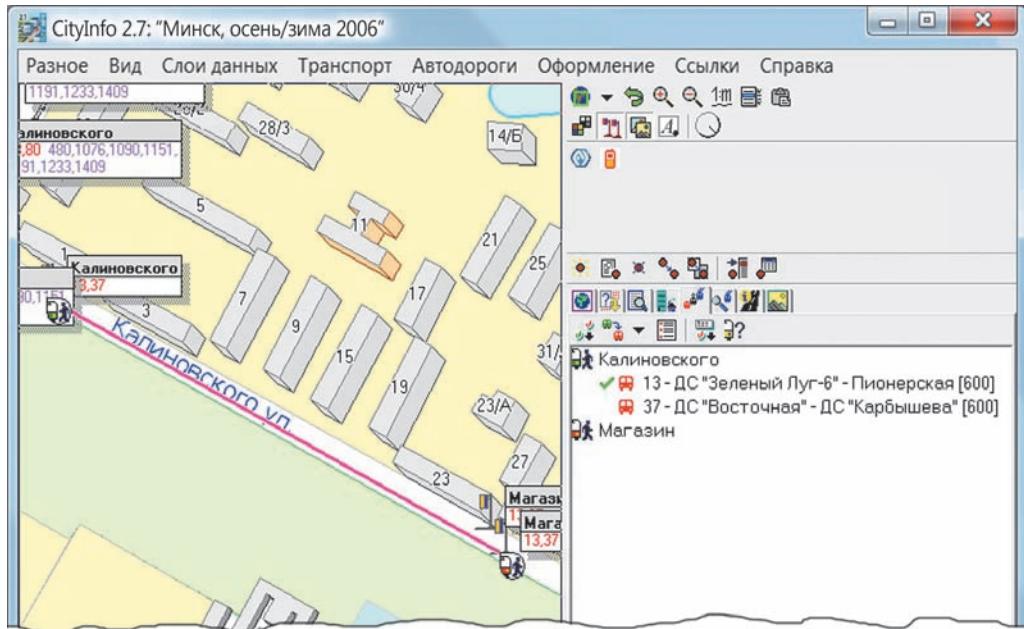
Правообладатель Народная асвета

Агульная колькасць дакументаў, даступных у гэтай сістэме, перавышае 120 тысяч. Тэксты прававых актаў дающца ў апошняй рэдакцыі, г.зн. з улікам усіх унесеных у іх змяненняў і дападненняў, што ўступілі ў сілу.

Важнае эканамічнае значэнне маюць **геаграфічныя сістэмы (ГІС)**, якія забяспечваюць збор, захоўванне, адлюстраванне і распаўсюджванне геаграфічных даных. Напрыклад, лічбавыя карты дэманструюць размяшчэнне аб'ектаў на мясцовасці для выяўлення адносін паміж імі, для даследавання змяненняў, што адбываюцца на вывучаемай тэрыторыі за вызначаны перыяд часу. Асноўная вобласці прыменення ГІС: экалогія і прыродакарыстанне, землеўпарядкованне, рэгіональнае планаванне, дэмографія і даследаванне працоўных рэсурсаў.

Для решэння задач у вобласці землеўпарядковання і земельнага кадастра прызначана **Земельно-информационная система Республики Беларусь** (<http://landgis.by>).

Адной з найбольш папулярных геаграфічных сістэм з'яўляецца сістэма CityInfo, якая прапануе электронную карту Мінска або іншых абласных гарадоў (рыс. 4.2) і забяспечвае пошук неабходнага аб'екта. Пасля ўводу ў акне **Адрес** назвы вуліцы і нумара дома можна ўбачыць на карце ў павялічным маштабе месца



Рыс. 4.2

Правообладатель Народная асвета

размяшчэння адшуканага сістэмай будынка. З дапамогай укладкі **Транспорт** можна даведацца аб неабходным відзе грамадскага транспарту і нумары маршрута для перамяшчэння з аднаго пункта горада ў другі. Для гэтага трэба пstryкнуць левай кнопкай мыши па двух пунктах на карце горада.

Усё большае распаўсядженне атрымліваюць сістэмы спадарожнікавай навігациі (напрыклад, GPRS), якія на аснове закладзеных у іх лічбавых карт дазваляюць вызначаць месцазнаходжанне аб'ектаў і могуць эфектуўна выкарыстоўвацца, напрыклад, для кіравання дарожным рухам.

Рашэнню актуальных экалагічных проблем дапамагаюць **біяінфармацыйныя сістэмы**, напрыклад **Красная книга Республики Беларусь** (<http://redbook.minpriroda.by>) (рыс. 4.3).

Красная книга Республики Беларусь змяшчае звесткі аб рэдкіх відах жывёл і дзікарослых раслін, што знаходзяцца пад пагрозай знікнення на тэрыторыі Беларусі. Яе рэсурсы з'яўляюцца агульнадаступнымі.



Рыс. 4.3

Правообладатель Народная асвета

Карыстальнік мае магчымасць выканату пошук неабходных звестак аб жывёлах або раслінах па каталоге або па алфавіце.

Інфарматыка разам з кібернетыкай, электронікай, біялогіяй займаецца распрацоўкай інтэлектуальных інфармацыйных сістэм, ствараемых на базе камп'ютара, для імітацыі рашэння чалавекам складаных інтэлектуальных задач.

Інтэлектуальная інфармацыйная сістэмы маюць здольнасць бачыць, чуць, гаварыць, думаць і нават адчуваць. У рамках даследавання ў вобласці стварэння такіх сістэм распрацоўваюцца натуральныя інтэрфейсы камп'ютара, будуюцца робаты і робататэхнічныя комплексы.

Прыкладам вядомай сістэмы штучнага інтэлекту з'яўляеца сістэма, якая імітуе зрокавае ўспрыманне. Гэта сістэма складаецца з камп'ютарных праграм і прылад распознавання вобразаў. Яна працуе ў двух рэжымах: навучання і распознавання. У рэжыме навучання чалавек або робат, што выконваюць ролю настаўніка, прад'яўляюць сістэме розныя аб'екты або прадметы і падаюць усе неабходныя звесткі аб іх. Затым у рэжыме распознавання чалавек паказвае сістэме новыя аб'екты, і яна павінна іх пазнаць або класіфікаваць.

Экспертныя сістэмы (ЭС) — гэта напрамак даследавання ў па стварэнні штучнага інтэлекту. Існуюць экспертныя сістэмы, якія перадаюць волыт больш падрыхтаваных карыстальнікам менш падрыхтаваным; інтэлектуальная навучальная, інфармацыйна-лагічныя і робататэхнічныя сістэмы; сістэмы-кансультанты і трэнажорныя сістэмы; сістэмы падтрымкі прыняцця рашэння.

Асноўныя вобласці прыменення ЭС: медыцина, электроніка, вылічальная тэхніка, геалогія, матэматыка, космас, сельская гаспадарка, кіраванне, фінансы, юрыспрудэнцыя і інш.

У медыцине створана некалькі соценъ экспернтых сістэм, што выкарыстоўваюцца для ўстанаўлення сувязі паміж парушэннямі дзейнасці арганізма чалавека і іх прычынамі. З дапамогай пашыранай базы ведаў пры атрыманні ўсіх магчымых звестак аб стане здароўя чалавека такія сістэмы могуць ставіць дыягназ на ўзроўні ўрача-кансультанта.

- ?
1. Якую сістэму называюць інфармацыйной сістэмай?
 2. Прывядзіце прыклады інфармацыйных сістэм.

§ 17. Інфармацыйныя тэхналогіі

Азначэнне інфармацыйных тэхналогій цесна звязана з паняццем «тэхнологія».

Тэрмін «тэхнологія» паходзіць ад грэчаскага слова *technē* — *навука аб уменні, майстэрстве, мастацтве* — і мае мноства значэнняў.

У вузкім сэнсе пад тэхналогіяй разумеюць працэс, вызначаемы сукупнасцю аперацый, прыёмаў, з дапамогай якіх функцыянуюць канкрэтныя прылады вытворчасці: механізмы, станкі, розная апаратура.

У больш шырокім сэнсе з дапамогай тэхналогій апісваюцца многія вытворчыя, эканамічныя, сацыяльныя, культурныя і іншыя працэсы, што адбываюцца ў грамадстве.

Паняцце «інфармацыйныя тэхналогіі» ўпершыню было ўжыта ў канцы 50-х гадоў ХХ ст. у Англіі і ЗША, аднак яго актыўнае выкарыстанне пачалося ў 80-я гады ХХ ст., калі пад уплывам новых тэхналогій у грамадстве пачалі шырока прымяняць тэрмін «інфармацыя».

У шырокім сэнсе пад **інфармацыйнымі тэхналогіямі (ІТ)** разумеюць комплекс узаемазвязаных навуковых, тэхналагічных, інжынерных дысцыплін, якія вывучаюць метады эфектыўнай арганізацыі працы людзей, занятых апрацоўкай і захоўваннем інфармацыі ва ўсіх сферах чалавечай дзейнасці: вытворчай, кірауніцкай, фінансавай, навуковай, сацыяльнай, культурнай.

У Законе Рэспублікі Беларусь ад 10 лістапада 2008 г. «Аб інфармацыі, інфарматызацыі і ахове інфармацыі» дадзена такое азначэнне інфармацыйнай тэхналогіі: **ІТ** — сукупнасць працэсаў, метадаў ажыццяўлення пошуку, атрымання, перадачы, збору, апрацоўкі, назапашвання, захоўвання, распаўсюджвання і (або) падачы інфармацыі, а таксама карыстання інфрамацый і аховы інфармацыі.

У сувязі з актыўным укараненнем вылічальнай тэхнікі і персанальных камп'ютараў у вузкім сэнсе пад **ІТ** звычайна разумеюць **камп'ютарныя інфармацыйныя тэхналогіі (КІТ)**.

Пад **камп'ютарнай інфармацыйнай тэхналогіяй** разумеюць працэс, які даваляе чалавеку ажыццяўляць апрацоўку, пошук, збор, захоўванне і перадачу інфармацыі з дапамогай камп'ютара.

Калі ў аснову класіфікацыі КІТ пакласці працэс апрацоўкі інфармацыі, прадстаўленай у рознай форме, то можна вылучыць **ІТ апрацоўкі тэкставай і графічнай інфармацыі**, **ІТ па работе з базамі даных**, **электроннымі табліцамі і інш.**

Трэба памятаць, што адна інфармацыйная тэхналогія можа ўключачь у сябе іншыя. Напрыклад, мультымедыйная тэхналогія ўключае ў сябе тэхналогіі апра-

цоўкі тэкставай і графічнай інфармацыі, тэхналогію стварэння анімацый, вэб-тэхналогію.

Інфармацыйную тэхналогію, якая апісвае працэс апрацоўкі тэкставай інфармацыі, можна паказаць у выглядзе іерархічнай структуры, вылучыўшы этапы, дзеянні і элементарныя аперацыі гэтага працэсу (рыс. 4.4).

Адрозненне паміж інфармацыйнымі сістэмамі і інфармацыйнымі тэхналогіямі з'яўляецца вельмі важным. Трэба дакладна разумець, што ў аснове інфармацыйной тэхналогіі ляжыць перш за ўсё *працэс*, выкананне якога пабудавана на спосабах дзеянасці і аперацыях, а ў аснове інфармацыйнай сістэмы ляжаць выкарыстаныя праграмныя і тэхнічныя *сродкі*.

Напрыклад, калі мы гаворым аб тэхналогіі апрацоўкі тэкставай інфармацыі, мы можам не прывязваць яе да канкрэтнай камп'ютарнай праграмы або сістэмы, паколькі працэс рэдагавання тэксту з'яўляецца агульным для любых камп'ютарных праграм, у якіх апрацоўваецца тэкставая інфармацыя.

Цяперашні час характарызуецца актыўным укараненнем інфармацыйных сістэм і тэхналогій у адукцыі.



Рыс. 4.4

Практычна ўсе вывучаныя Вамі камп'ютарныя тэхналогіі актыўна выкарыстоўваюцца ў адукцыі. Напрыклад, на аснове мультымедыйных тэхналогій ствараюцца вучэбныя камп'ютарныя презентацыі. Выкарыстанне аўдыё- і відэаматэрыялаў палягчае вывучэнне замежных моў. Тэхналогія апрацоўкі даных у электронных табліцах дапамагае не толькі выконваць складаныя разлікі, але і ствараць вучэбныя камп'ютарныя мадэлі з розных прадметных абласцей і вывучаць гэтыя віртуальныя мадэлі ў курсах фізікі, хіміі, біялогіі і г. д.

Рэгуляванне выкарыстання IC і IT у адукцыі, перспектывы і напрамкі іх развіцця знаходзяць сваё адлюстраванне ў праграмах «Комплекснай інфарматызацыі сістэмы адукцыі Рэспублікі Беларусь», якія распрацоўваюцца кожныя 4-5 гадоў.

Шырокое выкарыстанне ў беларускіх школах атрымалі электронныя сродкі навучання (ЭСН) па розных вучэбных прадметах.

Па сваім прызначэнні ЭСН можна падзяліць на навучальныя, тэставыя праграмы, вучэбныя інфармацыйна-пошукаўыя сістэмы, праграмныя сродкі для мадэліравання і дэманстрацыі, вучэбныя гульнёвыя праграмы, праграмы для вольнага часу і г. д.

Пералік рэкамендаваных ЭСН для адукцыйных устаноў размешчаны на сایце Галоўнага інфармацыйна-аналітычнага цэнтра Беларусі (адрас у сетцы Інтэрнэт — <http://www.giac.unibel.by>).

Штогод у нашай краіне праводзіцца Рэспубліканскі конкурс «Камп'ютар. Адукцыя. Інтэрнэт» па распрацоўцы ЭСН для агульнаадукцыйных, прафесійных і спецыяльных навучальных устаноў, у якім могуць прымаць удзел вучні і педагогі.

Яшчэ адным напрамкам выкарыстання IC і IT з'яўляецца прымяненне іх у кіраўніцкай дзейнасці ў сістэме адукцыі. Распрацаваныя і ўжытыя ў Беларусі аўтаматызаваныя сістэмы кіравання ўстановамі адукцыі змяшчаюць базы даных з разнастайнай інфармацыйай аб настаўніках, навучэнцах, бацьках і інш.

З дапамогай вэб-тэхналогій ва ўстановах адукцыі ствараюцца вэб-сайты. Сайт школы звычайна змяшчае гісторыю і навіны школы, звесткі аб настаўніках, вучнях, аб дасягненнях школьнага калектыву, а таксама вучэбна-метадычныя матэрыялы.

У Беларусі распрацаваны і выкарыстоўваецца банк даных адoranай і таленавітай моладзі краіны, які ўтрымлівае інфармацыю аб лаўрэатах і стыпендыятах спецыяльнага фонду Прэзідэнта Рэспублікі Беларусь па сацыяльнай падтрымцы адoranых вучняў і студэнтаў, аб пераможцах, прызёрах і ўладальніках дыпломаў міжнародных і рэспубліканскіх алімпіяд і конкурсаў і г. д.

Для аказання дапамогі навучэнцам у выбары прафесіі і навучальнай установы для прадаўжэння адукцыі распрацаваны і выкарыстоўваюцца сістэмы інфармацыйна-педагагічнай падтрымкі старшакласнікаў, напрыклад Абитуриент.by (www.abiturient.by).

The screenshot shows the homepage of the website abiturient.by. At the top, there's a navigation bar with links for 'Поиск по сайту...' (Search), 'Служебный вход' (Administrative login), and several subject categories: Математика (Mathematics), Русский язык (Russian language), Белорусский (Belarusian), Физика (Physics), and Химия (Chemistry). Below the navigation, there are three main sections: 1) 'Кем быть?' (What to become?), 2) 'Где учиться?' (Where to study?), and 3) 'Как поступить?' (How to enter). Each section has a brief description and a call-to-action button.

- ВАШЕМУ ВНИМАНИЮ**
- Главная
- Новости
- Каталог вузов
- Каталог специальностей
- Мониторинг
- Информация приемных комиссий вузов 2010 г.
- Полезно знать

НАШИ НОВОСТИ

06.04.10 [Началось профессионально-психологическое собеседование](#)

С 05.04.2010 в вузах, перечень которых утвержден Министерством образования, проводится профессионально-психологическое собеседование для граждан, предполагающих участвовать в конкурсе по специальностям высшего образования «Государственное управление», «Государственное управление и право», «Государственное управление и экономика», «Международные отношения», «Международное право», «Правоведение».

Рыс. 4.5

Гэта сістэма ажыццяўляе маніторынг уступнай кампаніі ў ВНУ Рэспублікі. Поўны спіс спецыяльнасцей кожнай ВНУ, па якіх праводзіцца набор абітурыентаў на ўсе формы навучання, дадзены на ўкладцы **Где учиться?** (рыс. 4.5) у раздзеле **План приема** адпаведнай ВНУ.

Даведацца пра вынікі рэпетыцыйнага і цэнтралізаванага тэсціравання можна на сайце Рэспубліканскага інстытута кантролю ведаў (www.rikz.unibel.by). Тут жа змешчаны звесткі аб метадычных дапаможніках для падрыхтоўкі абітурыентаў да тэсціравання.

На заканчэнне адзначым, што інтэнсіўнасць выкарыстання інфармацыйных сістэм і тэхналогій ва ўсіх сферах чалавечай дзеянасці з кожным годам няўхільна нарастает.



1. Якую тэхналогію называюць камп'ютарнай інфармацыйной тэхналогіяй?
2. Прывядзіце прыклады камп'ютарных інфармацыйных тэхналогій.
3. Якія інфармацыйныя тэхналогіі і сістэмы выкарыстоўваюцца ў Вашай навучальнай установе?

Структура HTML-документа

<pre><html> </html> <head> </head> <title> </title> <body> </body></pre>	<p>Пачатак і канец HTML-документа Галаўная частка: службовая інфармацыя Загаловак акна, у якім адлюстроўваецца дакумент Цела HTML-документа</p>
--	---

Фон і колер старонкі

<pre><body background="imf"> <body bgcolor="колер"> <body text="колер"></pre>	<p>Фонавая карцінка: <i>imf</i> — імя файла або URL Колер фону: колер — назва або код колеру Колер тэксту: колер — назва або код колеру</p>
---	---

Структура старонкі

<pre><p> <div> <p align=left right center> <div align=left right center>
 <hr> <hr size=h> <hr width=w> <hr align=left right center></pre>	<p>Абзац Раздел (можа змяшчаць некалькі абзашаў) Гарызантальнае выраўноўванне абзаша або раздзела: па левым краі па правым краі па цэнтры старонкі Пераход на новы радок Гарызантальная раздзяляльная лінія Таўшчыня лініі ў пікселях Даўжыня лініі ў пікселях або працэнтах ад шырыні акна Гарызантальнае выраўноўванне лініі</p>
---	--

Фарматаванне тэксту

<pre> <i> <u> <sup> <sub> <h1>... <h6> <pre></pre>	<p>Паўтлустае напісанне Курсіўнае напісанне Падкрэсленое напісанне Верхні індэкс Ніжні індэкс Загалоўкі (6 узроўняў: 1 — 6) Заданне шрыфту: шрыфт — назва шрыфту Памер шрыфту. Умоўныя адзінкі: 1 2 3 4 5 6 7 Адносныя адзінкі: -2 -1 0 +1 +2 +3 +4 Колер шрыфту: колер — назва або код колеру Аўтарскае фарматаванне. Тэкст у браўзеры адлюстроўваецца так, як запісаны ў HTML-кодзе</p>
---	---

Відарысы на старонцы

<pre> </pre>	<p>Спасылка на рысунак: <code>imf</code> — імя файла або URL Шырыня <code>w</code> і вышыня <code>h</code> відарыса ў пікселях Таўшчыня рамкі вакол відарыса ў пікселях</p> <p>Вертыкальнае выраўноўванне рысунка адносна суседніх аб'ектаў (тэксту): па верхнім краі па ніжнім краі па цэнтры</p> <p>Гарызантальнае выраўноўванне рысунка: злева ад тэксту справа ад тэксту</p> <p>Размяшчэнне падказкі: <code>txt</code> — тэкст паведамлення</p>
---	---

Гіперспасылкі

<pre> <body link="колер"> <body alink="колер"> <body vlink="колер"></pre>	<p>Спасылка на старонку, якая адкрываецца ў <code>tym</code> жа акне: <code>imf</code> — імя файла або URL Спасылка на старонку, якая адкрываецца ў іншым акне: <code>imf</code> — імя файла або URL Стварэнне закладкі з іменем <code>nm</code> Спасылка на закладку з іменем <code>nm</code> у <code>tym</code> жа дакуменце Спасылка на закладку з іменем <code>nm</code> у іншым дакуменце: <code>imf</code> — імя файла або URL Колер непрагледжанай спасылкі: колер — назва або код колеру Колер актыўнай спасылкі: колер — назва або код колеру Колер прагледжанай спасылкі: колер — назва або код колеру</p>
---	--

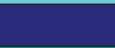
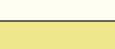
Спіскі

	Маркіраваны спісак
	Элемент спіска
<ul type=circle disc square>	Тып маркёра: акружнасць круг квадрат
	Нумараўаны спісак
<ol type=1 a A>	Спосаб нумарацыі: арабскія лічбы малыя лацінскія літары вялікія лацінскія літары
<ol start=n>	Пачатак нумарацыі з нумара n

Табліцы

<table>	Табліца
<table width=w>	Шырыня табліцы ў пікселях або працэнтах ад шырыні акна
<table align=left right center >	Гарызантальнае выраўноўванне табліцы адносна суседніх аб'ектаў (акна): па левым краі па правым краі па цэнтры
<table border=w>	Таўшчыня меж у пікселях
<table cellspacing=w>	Адлегласць паміж ячэйкамі ў пікселях
<table cellpadding=w>	Адлегласць паміж змесцівам ячэйкі і яе мяжой у пікселях
<tr>	Радок табліцы
<td>	Ячэйка табліцы
<td width=w>	Шырыня ячэйкі ў пікселях або ў працэнтах
<td align=left right center >	Гарызантальнае выраўноўванне змесціва: па левай мяжы па правай мяжы па цэнтры ячэйкі
<td valign=top bottom middle>	Вертыкальнае выраўноўванне змесціва: па верхній мяжы па ніжній мяжы па цэнтры
<td bgcolor="колер">	Колер ячэйкі
<td colspan=n>	Аб'яднанне n слупкоў
<td rowspan=n>	Аб'яднанне n радкоў

Назвы і шаснаццадковыя коды колераў

Назва	Код	Колер
aqua	#00FFFF	
black	#000000	
blue	#0000FF	
blueviolet	#8A2BE2	
brown	#A52A2A	
chocolate	#D2691E	
coral	#FF7F50	
crimson	#DC143C	
cyan	#00FFFF	
darkblue	#00008B	
darkcyan	#008B8B	
darkgray	#A9A9A9	
darkgreen	#006400	
darkmagenta	#8B008B	
darkorange	#FF8C00	
darkred	#8B0000	
gold	#FFD700	
gray	#808080	
green	#008000	
indigo	#4B0082	
ivory	#FFFFFF	
khaki	#F0E68C	
lavender	#E6E6FA	
lightblue	#ADD8E6	
lightcyan	#E0FFFF	
lightgreen	#90EE90	
lightgrey	#D3D3D3	
lightpink	#FFB6C1	

Назва	Код	Колер
lime	#00FF00	
magenta	#FF00FF	
maroon	#800000	
mediumblue	#0000CD	
navy	#000080	
olive	#808000	
orange	#FFA500	
orchid	#DA70D6	
palegreen	#98FB98	
peru	#CD853F	
pink	#FFC0CB	
plum	#DDA0DD	
purple	#800080	
red	#FF0000	
rosybrown	#BC8F8F	
salmon	#FA8072	
seagreen	#2E8B57	
sienna	#A0522D	
silver	#C0C0C0	
skyblue	#87CEEB	
tan	#D2B48C	
teal	#008080	
tomato	#FF6347	
violet	#EE82EE	
wheat	#F5DEB3	
white	#FFFFFF	
yellow	#FFFF00	
yellowgreen	#9ACD32	

Графічныя прымітывы модуля GraphABC

<code>SetPixel(x,y,color: integer);</code>	Зафарбоўвае адзін піксель з каардынатамі (x, y) колерам <code>color</code>
<code>MoveTo(x,y: integer);</code>	Перасоўвае нябачнае пяро да пункта з каардынатамі (x, y) ; працуе разам з функцыяй <code>LineTo(x,y)</code>
<code>LineTo(x,y: integer);</code>	Рысуе адрэзак ад бягучага месцазнаходжання пяра да пункта (x, y) ; каардынаты пяра пры гэтым становяцца роўнымі (x, y)
<code>Line(x1,y1,x2,y2: integer);</code>	Рысуе адрэзак з пачаткам у пункце $(x1, y1)$ і канцом у пункце $(x2, y2)$
<code>Circle(x,y,r: integer);</code>	Рысуе акружнасць з цэнтрам у пункце (x, y) і радыусам <code>r</code>
<code>Ellipse(x1,y1,x2,y2: integer);</code>	Рысуе эліпс, зададзены апісаным вакол яго прамавугольнікам з каардынатамі процілеглых вяршынь $(x1, y1)$ і $(x2, y2)$
<code>Rectangle(x1,y1, x2,y2: integer);</code>	Рысуе прамавугольнік, зададзены каардынатамі процілеглых вяршынь $(x1, y1)$ і $(x2, y2)$
<code>Arc(x,y,r,a1,a2: integer);</code>	Рысуе дугу акружнасці з цэнтрам у пункце (x, y) і радыусам <code>r</code> , змешчаную паміж двума праменямі, што ўтвараюць вуглы $a1$ і $a2$ з восцю OX ($a1$ і $a2$ — рэчыўныя, задаюцца ў градусах і адлічваюцца супраць гадзіннікаў стрэлкі)
<code>Pie(x,y,r,a1,a2: integer);</code>	Рысуе сектар круга, абмежаваны дугой (параметры працэдуры маюць той жа сэнс, што ў працэдуры <code>Arc</code>)
<code>FloodFill(x,y,color: integer);</code>	Залівае вобласць колерам <code>color</code> , пачынаючы з пункта (x, y)
<code>TextOut(x,y: integer; s: string);</code>	Выводзіць радок <code>s</code> у пазіцыю (x, y) (пункт (x, y) задае верхні левы вугал прамавугольніка, які будзе змяшчаць тэкст з радка <code>s</code>)

Дзеянні з графічным акном

<code>SetWindowSize(w,h: integer);</code>	Задае шырыню і вышыню графічнага акна
<code>SetWindowCaption(s: string);</code>	Задае загаловак графічнага акна
<code>SaveWindow(fname: string);</code>	Захоўвае змесціва графічнага акна ў файл з іменем <code>fname</code>
<code>LoadWindow(fname: string);</code>	Выводзіць у графічнае акно рысунак з файла з іменем <code>fname</code>
<code>ClearWindow;</code>	Ачышчае графічнае акно белым колерам
<code>ClearWindow(c: ColorType);</code>	Ачышчае графічнае акно колерам <code>c</code>
<code>Redraw;</code>	Ажыццяўляе перарысоўку акна

Канстанты стандартных колераў

clBlack	чорны	clBrown	карычневы
clPurple	фіялетавы	clBlue	сіні
clWhite	белы	clSkyBlue	блакітны
clMaroon	цёмна-чырвоны	clYellow	жоўты
clRed	чырвоны	clGray	шэры
clGreen	зялёны	clSilver	сярэбраны

Дзеянні з пяром і пэндзлем

SetPenColor(color: integer);	Задае колер пяра, які апісваецца параметрам color
SetPenWidth(w: integer);	Задае шырыню пяра, роўную w пікселям
SetPenStyle(ps: integer);	Задае стыль пяра, які апісваецца параметрам ps. Стылі пяра задаюцца іменаванымі канстантамі: psSolid, psDot, psClear, psDash, psDashDot, psDashDotDot
SetBrushColor(color: integer);	Задае колер пэндзля, які задаецца параметрам color
SetBrushPicture(fname: string);	Задае ў якасці ўзору для зафарбоўкі пэндзлем відарыс, што захоўваецца ў файле fname
SetBrushStyle(bs: integer);	Задае стыль пэндзля, які апісваецца параметрам bs. Стылі пэндзля задаюцца іменаванымі канстантамі: bsSolid, bsHorizontal, bsBDiagonal, bsCross, bsClear, bsDiagCross, bsVertical, bsFDiagonal

Дзеянні з рысункамі

LoadPicture(fname: string): integer;	Загружает рысунак з файла с именем fname в апертуру памяти и заменяет аписальнік рысунка ў цэлую зменную n (n:=LoadPicture(fname))
DrawPicture(n,x,y: integer);	Выводзіць рысунак з аписальнікам n у пазіцыю (x,y) графічнага акна
DrawPicture(n,x,y,w,h: integer);	Выводзіць рысунак з аписальнікам n у пазіцыю (x,y) графічнага акна і маштабуе яго памеры да шырыні w і вышыні h
SavePicture(n: integer; fname: string);	Захоўвае рысунак з аписальнікам n у файл з іменем fname (форматы BMP, JPG або GIF)
SetPictureSize(n,w,h: integer);	Задае памер рысунка з аписальнікам n, роўны w × h пікселяў
SetPictureTransparent (n: integer; b: boolean);	Падключочае (b=True) або адключочае (b=False) рэжым прозыстасці пры рысаванні відарыса з аписальнікам n

Ад аўтараў	3
Раздзел 1. Асновы вэб-канструявання	
§ 1. Уяўленне аб вэб-канструяванні	4
1.1. Інструменты і метады распрацоўкі вэб-сайтаў	—
1.2. Праектаванне сайта	6
§ 2. Выкарыстанне офісных дадаткаў для стварэння вэб-сторонак	8
2.1. Стварэнне вэб-сторонак у рэдактары MS Word	—
2.2. Захаванне презентацый PowerPoint у выглядзе вэб-сторонак	14
§ 3. Асновы мовы разметкі HTML	18
3.1. Стварэнне HTML-документа ў рэдактары Блокнот	—
3.2. Відарысы на вэб-сторонках	24
3.3. Стварэнне гіперспасылак	30
§ 4. Падрыхтоўка відарысаў для Інтэрнэта	35
§ 5. Вэб-канструяванне ў рэдактары FrontPage	41
5.1. Асноўныя элементы інтэрфейса	—
5.2. Работа ў рэдактары FrontPage	43
5.3. Выкарыстанне табліц	50
5.4. Распрацоўка вэб-сайта	54
5.5. Публікацыя сайта	61
§ 6. Стварэнне фрагментаў сайтаў па розных прадметных абласцях	62
Раздзел 2. Асновы алгарытмізацыі і праграмавання	
§ 7. Выкананне практычных заданняў з розных прадметных абласцей	65
7.1. Пабудаванне геаметрычных фігур	—
7.2. Выкарыстанне растральных відарысаў	70
7.3. Пабудаванне графікаў і дыяграм	79
§ 8. Выкананне практычных заданняў па тэмах вучэбных прадметаў	85
8.1. Астраномія	—
8.2. Геаграфія	90
8.3. Біялогія і экалогія	94
8.4. Фізіка	98
Раздзел 3. Апрацоўка інфармацыі ў сістэме кіравання базамі даных	
§ 9. Базы даных і сістэмы кіравання базамі даных	104
§ 10. Стварэнне табліцы базы даных	108
10.1. Праектаванне базы даных	—
10.2. Стварэнне структуры табліцы	111
§ 11. Звязванне табліц базы даных	117
§ 12. Стварэнне і запаўненне формы	121
§ 13. Пошук даных з дапамогай запытаў	125
§ 14. Сартаванне запісаў у табліцы	129
§ 15. Стварэнне справаздач	131
Раздзел 4. Інфармацыйныя сістэмы і тэхналогіі	
§ 16. Інфармацыйныя сістэмы	135
§ 17. Інфармацыйныя тэхналогіі	140
Дадаткі	144

Вучэбнае выданне

Забароўскі Георгій Аляксандравіч
Пупцаў Аляксандр Яўгенавіч

ІНФАРМАТЫКА

Вучэбны дапаможнік для 11 класа
агульнаадукацыйных устаноў
з беларускай мовай навучання

Заг. рэдакцыі *В. Г. Бехціна*. Рэдактар *Н. М. Алганава*. Маастацкі рэдактар *Л. А. Дашкевіч*.
Тэхнічны рэдактар *Г. А. Дудко*. Карактары *З. М. Грышиэлі*, *Т. М. Вядзвернікова*, *Д. Р. Лосік*, *В. С. Бабеня*, *Г. В. Алешка*.

Падпісана ў друк 12.07.2010. Фармат $70 \times 90^1/16$. Папера афсетная. Гарнітура літаратурная.
Афсетны друк. Ум. друк. арк. 11,12. Ул.-выд. арк. 9,1. Тыраж 21 520 экз. Заказ .

Выдавецкае рэспубліканскае ўнітарнае прадпрыемства «Народная асвета»

Міністэрства інфармацыі Рэспублікі Беларусь.

ЛІ 02330/0494083 ад 03.02.2009.

Пр. Пераможцаў, 11, 220004, Мінск.

ААТ «Паліграфкамбінат імя Я. Коласа»

ЛП № 02330/0150496 ад 11.03.2009.

Вул. Чырвоная, 23, 220600, Мінск.

Правообладатель Народная асвета

(Назва і нумар школы)

Навучальны год	Імя і прозвішча вучня	Стан вучэбнага дапаможніка пры атрыманні	Адзнака вучню за карыстанне вучэбным дапаможнікам
20 /			
20 /			
20 /			
20 /			
20 /			