

ИФАРМАТЫКА

У. М. Котаў
А. І. Лапо
Ю. А. Быкадораў
А. М. Вайцеховіч

11



ІНФАРМАТЫКА

Вучэбны дапаможнік для 11 класа
ўстаноў агульнай сярэдняй адукацыі
з беларускай мовай навучання
(з электроннымі дадаткамі)

*Данушчана
Міністэрствам адукацыі
Рэспублікі Беларусь*

Мінск «Народная асвета» 2021

Правообладатель Народная асвета

УДК 004(075.3=161.3)
ББК 32.81я721
И74

Пераклад з рускай мовы *К. І. Чэрнікавай*

Аўтары:

У. М. Котаў, А. І. Лапо, Ю. А. Быкадораў, А. М. Вайцеховіч

Рэцэнзенты:

кафедра інфармацыйных тэхналогій і мадэлявання эканамічных працэсаў установы адукацыі «Беларускі дзяржаўны аграрны тэхнічны ўніверсітэт» (кандыдат педагагічных навук, дацэнт, загадчык кафедры *А. Л. Салун*); настаўнік інфарматыкі кваліфікацыйнай катэгорыі «настаўнік-метадыст» дзяржаўнай установы адукацыі «Гімназія № 1 г. Брэста» *І. Ю. Гарбацэвіч*

Электронны дадатак «Інфармацыйныя тэхналогіі»
(базавы ўзровень), электронны дадатак для павышанага ўзроўню
змяшчаны на рэсурсе profil.adu.by

ISBN 978-985-03-3637-8

© Чэрнікава К. І., пераклад на беларускую мову,
2021

© Афармленне. УП «Народная асвета», 2021

Правообладатель Народная асвета

ЗМЕСТ

Ад аўтараў 4

Глава 1. УВОДЗІНЫ Ў АБ'ЕКТНА-ПАДЗЕЙНАЕ ПРАГРАМІРАВАННЕ

| | |
|---|----|
| § 1. Аб'ектна-падзейная мадэль работы праграмы | 6 |
| 1.1. Элементы кіравання ў дадатках з графічным інтэрфейсам | — |
| 1.2. Падзеі | 9 |
| § 2. Візуальнае асяроддзе распрацоўкі праграм | 10 |
| 2.1. Структура праекта | — |
| 2.2. Інтэрфейс асяроддзя праграмавання | 11 |
| 2.3. Работа з формай | 12 |
| § 3. Праектаванне інтэрфейса аконнага дадатку з выкарыстаннем элементаў кіравання | 15 |
| 3.1. Асноўныя элементы кіравання | — |
| 3.2. Элемент кіравання кнопка (Button) | 17 |
| 3.3. Элемент кіравання метка (Label) | 18 |
| 3.4. Элемент кіравання тэкставае поле (Edit) | 19 |
| 3.5*. Элементы кіравання флажок (CheckBox) і пераключальнік (RadioButton) | 22 |
| § 4. Элементы кіравання для работы з графікай | 27 |
| 4.1. Элемент кіравання для ўстаўкі малюнка (PictureBox) | — |
| 4.2. Пабудова графікаў функцый | 28 |
| 4.3. Пабудова дыяграм | 31 |
| 4.4. Анімацыя | 32 |
| § 5. Стварэнне дадаткаў | 36 |
| 5.1. Распрацоўка аконных дадаткаў | — |
| 5.2. Стандартныя дыялогі | 37 |
| 5.3. Стварэнне меню | 38 |
| 5.4. Стварэнне дадатку «Блакнот» | 40 |
| 5.5. Стварэнне дадатку «Графічны рэдактар» | 41 |
| 5.6. Стварэнне дадатку «Калькулятар» | 44 |

Глава 2. АСНОВЫ ВЭБ-КАНСТРУАВАННЯ

| | |
|--|----|
| § 6. Вэб-канструаванне. Асноўныя паняцці | 46 |
| 6.1. Вэб-сайт | — |
| 6.2. Мова гіпертэкставай разметкі дакумента HTML. Структура HTML-дакумента. Тэгі і атрыбуты. Гіперспасылкі | 47 |
| § 7. Стварэнне вэб-старонак | 51 |
| 7.1. Інструменты стварэння вэб-старонак | — |
| 7.2. Элементы афармлення вэб-старонак | 52 |
| 7.3. Тэкст на вэб-старонцы | 53 |
| 7.4. Гіперспасылкі на вэб-старонцы | 55 |
| § 8. Паняцце пра каскадныя табліцы стыляў | 57 |
| § 9. Мультымедыя на вэб-старонках | 64 |
| 9.1. Графіка на вэб-старонках | — |
| 9.2. Гук і відэа на вэб-старонках | 66 |
| § 10. Візуальнае вэб-канструаванне | 71 |
| § 11. Распрацоўка фрагментаў тэматычных сайтаў | 75 |

Глава 3. ІНФАРМАЦЫЙНЫЯ ТЭХНАЛОГІ Ў ГРАМАДСТВЕ

| | |
|---|-----|
| § 12. Інфармацыйныя сістэмы, тэхналогіі і рэсурсы | 78 |
| 12.1. Інфармацыйныя сістэмы | — |
| 12.2. Інфармацыйныя тэхналогіі | 81 |
| 12.3. Інфармацыйныя рэсурсы | 86 |
| § 13. Інфарматызацыя грамадства | 88 |
| § 14. Адукацыя і прафесійная дзейнасць у інфармацыйным грамадстве | 91 |
| 14.1. Адукацыя ў інфармацыйным грамадстве | — |
| 14.2. Прафесійная дзейнасць у інфармацыйным грамадстве | 93 |
| § 15. Кібербяспека і кіберустойлівасць | 95 |
| 15.1. Кібербяспека | — |
| 15.2. Уразлівасць інфармацыйнай інфраструктуры | 97 |
| 15.3. Забеспячэнне кібербяспекі. Кіберустойлівасць | 99 |
| Дадатак да главы 1 | 102 |
| Дадатак да главы 2 | 107 |

Ад аўтараў

Паважаныя адзінаццацікласнікі!

Мы жывём у час хуткіх змяненняў, калі інфарматыка становіцца найважнейшым складальнікам усёй сістэмы навуковага пазнання, вызначае шляхі фарміравання інфармацыйнага грамадства.

У сучасным свеце патрэбнай якасцю асобы становіцца высокі ўзровень яе інфармацыйнай культуры, а інфармацыя з’яўляецца адным з асноўных рэсурсаў, які вызначае развіццё краіны, яе будучыню.

Структурна ў інфарматыцы вылучаюць такія раздзелы, як тэарэтычная інфарматыка і прыкладная інфарматыка. У школьным курсе прадстаўлены абодва гэтыя раздзелы.

Тэарэтычная інфарматыка — фундаментальны складальнік вучэбнага прадмета. Вывучэнне агульных заканамернасцей працякання інфармацыйных працэсаў, асноўных паняццяў логікі, тэорыі алгарытмаў, моў праграмавання дазваляе развіваць у навучэнцаў лагічнае і алгарытмічнае мысленне, выходзяць інфармацыйную культуру, фарміруе навуковы светапогляд.

Прыкладная інфарматыка накіравана на ўжыванне паняццяў і вынікаў тэарэтычнай інфарматыкі да рашэння канкрэтных задач у канкрэтных прыкладных галінах. Прыкладная інфарматыка ўключае ў сябе архітэктuru камп’ютара і камп’ютарных сетак, камп’ютарнай графікі, камп’ютарнага мадэлявання, баз даных, інфармацыйных тэхналогій і інш. Вывучэнне прыкладной інфарматыкі накіравана на фарміраванне камп’ютарнай пісьменнасці — валоданне патрэбным наборам ведаў і навыкаў работы на камп’ютары для апрацоўкі, захоўвання, перадачы, атрымання і выкарыстання даных. Прыкладная інфарматыка — самы дынамічны складальнік інфарматыкі.

У друкаваным варыянце вучэбнага дапаможніка змяшчаюцца главы «Уводзіны ў аб’ектна-падзейнае праграмаванне», «Асновы вэб-канструявання» і «Інфармацыйныя тэхналогіі ў грамадстве». Глава «Камп’ютарнае мадэляванне» прадстаўлена ў электронным дадатку да вучэбнага дапаможніка. Акрамя таго, у электронным дадатку змяшчаецца альтэрнатыўны варыянт для вывучэння тэмы «Уводзіны ў аб’ектна-падзейнае праграмаванне» ў асяроддзі праграмавання Delphi, якое з’яўляецца прафесійным асяроддзем хуткай распрацоўкі дадаткаў з аконным інтэрфейсам. Дадзеная магчымасць дазволіць убачыць, што адно і тое ж заданне можа быць рэшана з выкарыстаннем розных інструментаў.

Матэрыялы вучэбнага дапаможніка падзелены на дзве калонкі. Колер фону дапаможа вам разабрацца ў прызначэнні змешчанай на ім інфармацыі.



— асноўныя матэрыялы, абавязковыя для вывучэння;



— прыклады, якія ілюструюць асноўныя матэрыялы;



— азначэнні асноўных паняццяў;



— гістарычныя звесткі, інфармацыя пра вучоных, якія ўнеслі ўклад у развіццё інфарматыкі, і іншыя цікавыя факты.

У вучэбным дапаможніку выкарыстоўваюцца наступныя ўмоўныя абазначэнні:



— пытанні і заданні для праверкі ведаў;



— раздзел «Практыкаванні» змяшчае заданні, пры выкананні якіх выкарыстоўваецца камп'ютар;



— раздзел «Практыкаванні» змяшчае заданні для выканання ў сшытку;



— раздзел «Практыкаванні» змяшчае заданні, пры выкананні якіх можа быць выкарыстана інфармацыя, змешчаная на Нацыянальным адукацыйным партале;

* — практыкаванне або прыклад для цікаўных.

У тэксце некаторых практыкаванняў вам будзе прапанавана адкрыць файл. Гэта азначае, што практыкаванне можна выканаць, выкарыстоўваючы файл, змешчаны на Нацыянальным адукацыйным партале <http://profil.adu.by> → катэгорыя «Інфарматыка/Інфарматыка» → курс «Інфармацыйныя тэхналогіі. 11 клас (базавы ўзровень)» або «Інфарматыка. 11 клас (павышаны ўзровень)». Зайсці на сайт і спампаваць файлы да практыкаванняў можна таксама з дапамогай матрычнага QR-кода:



Імя файла для спампоўвання змяшчае нумар параграфа і нумар практыкавання. Напрыклад, імя файла `pr10_3` азначае, што файл належыць да трэцяга практыкавання дзясятага параграфа.

Таксама на партале змешчаны файлы з праектамі, разгледжанымі ў прыкладах. Такія файлы маюць імя `pr3_1.zip` (праект для прыкладу 3.1).

Аўтары вучэбнага дапаможніка паспрабавалі зрабіць так, каб вы змаглі асвоіць патрэбныя сучаснаму чалавеку веды і навыкі ва ўмовах хуткага змянення інфармацыйных тэхналогій. У электронным дадатку да вучэбнага дапаможніка пададзены тыя тэмы 11-га класа, якія належаць да прыкладной інфарматыкі.



— дадатак «Інфармацыйныя тэхналогіі» змешчаны на электронным адукацыйным рэсурсе (<http://profil.adu.by>).



— матэрыял для павышанага ўзроўню змешчаны на электронным адукацыйным рэсурсе (<http://profil.adu.by>).

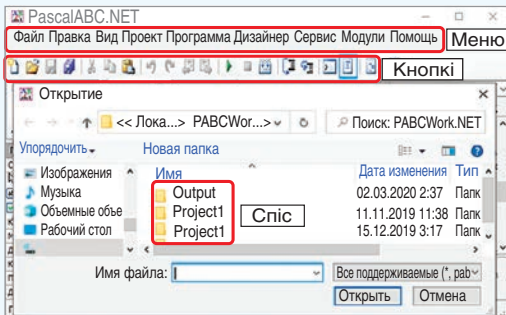
Глава 1 УВОДЗІНЫ Ў АБ'ЕКТНА-ПАДЗЕЙНАЕ ПРАГРАМАВАННЕ

§ 1. Аб'ектна-падзейная мадэль работы праграмы



Заснавальнікам RAD лічыцца супрацоўнік IBM, брытанскі кансультант па інфармацыйных тэхналогіях Джэймс Марцін (1933—2013), які ў пачатку 1990-х гг. сфармуляваў асноўныя прынцыпы RAD, засноўваючыся на ідэях Бары Войма і Скота Шульца.

Прыклад 1.1. Пасля загрузкі якога-небудзь рэдактара карыстальнік можа адкрыць файл для рэдагавання. Пры гэтым ён выбірае меню **Файл**, знаходзіць у спісе каманду **Адкрыць**, выбірае патрэбны файл, націскае кнопку **Адкрыць**. Як мы бачым, каб адкрыць файл, карыстальнік узаемадзейнічае з такімі элементамі кіравання, як меню, спіс, кнопка.



1.1. Элементы кіравання ў дадатках з графічным інтэрфейсам

Сучасныя праграмы, з якімі сёння працуюць карыстальнікі камп'ютара, адрозніваюцца ад тых, якія вы стваралі раней. Асноўнае адрозненне — узаемадзейнае карыстальніка з праграмай.

Праграмы, якія вы стваралі ў 7—10-м класах, узаемадзейнічалі з карыстальнікам з дапамогай тэкставага інтэрфейса (часта яго называюць інтэрфейсам каманднага радка). Пасля запуску праграмы вы ўводзілі даныя, праграма выконвалася, і вы бачылі вынік. І ўвод, і вывад даных ажыццяўляліся ў алфавітна-лічбавай форме.

Аперацыйныя сістэмы з графічным аконным інтэрфейсам (напрыклад, Windows) прадугледжваюць зносіны карыстальніка з праграмай з дапамогай элементаў кіравання. Да элементаў кіравання залічваюць: кнопкі, разнастайныя меню, тэкставыя паведамленні, спісы і інш. Пры рабоце праграмы карыстальнік выбірае які-небудзь элемент кіравання і выконвае над ім пэўнае дзеянне (прыклад 1.1). Калі такое дзеянне для выбранага элемента было вызначана, то праграма яго выконвае, інакш выдае паведамленне пра памылку.

Шмат якія сістэмы праграмавання дазваляюць ствараць праграмы з аконным інтэрфейсам. Такія праграмы называюць **аконнымі дадаткамі** (Windows Form Application).

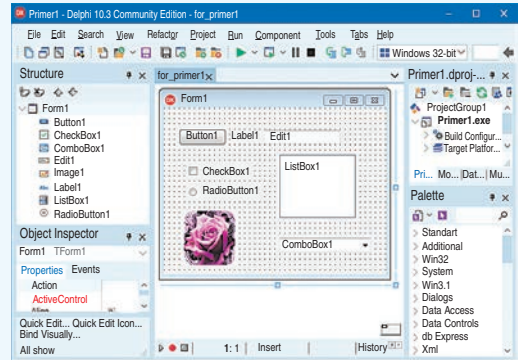
Праектаванне інтэрфейса акна праграмы можна выконваць з выкарыстаннем RAD-тэхналогіі (Rapid Application Development — **хуткая распрацоўка дадаткаў**). Тэхналогія RAD характэрная шмат для якіх сістэм праграмавання. Хуткая распрацоўка стала магчымай за кошт таго, што элементы кіравання былі візуалізаваны і сабраны ў спецыяльныя бібліятэкі — VCL (Visual Component Library — візуальная бібліятэка кампанентаў).

Розныя элементы кіравання можна перацягваць з палітры кампанентаў на форму з дапамогай мышы. Працэс стварэння інтэрфейса будучай праграмы ўяўляецца аналагам работы з якім-небудзь канструктарам. Праграмаванне ў RAD-асяроддзях з'яўляецца візуальным, паколькі код па стварэнні аб'екта не пішацца, а генеруецца асяроддзем. Задача праграміста — напісанне кода па кіраванні гатовымі кампанентамі.

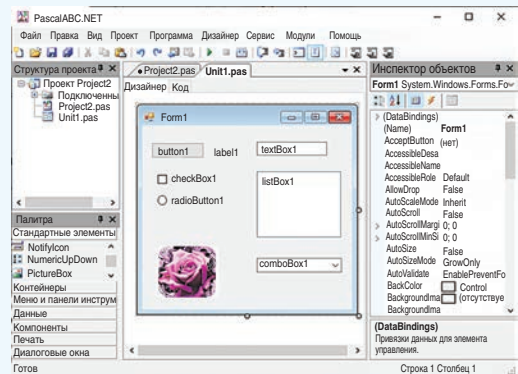
Візуальнае праграмаванне падтрымліваецца ў PascalABC і Delphi (код пішацца на мове Pascal), VisualBasic, C# і інш. (прыклад 1.2). Для навучання вучняў малодшых класаў выкарыстоўваецца візуальнае праграмаванне ў асяроддзі Скрэтч (Scratch).

Прыклад 1.2. Асяроддзі праграмавання, у якіх рэалізавана падтрымка падрыгмы візуальнага праграмавання.

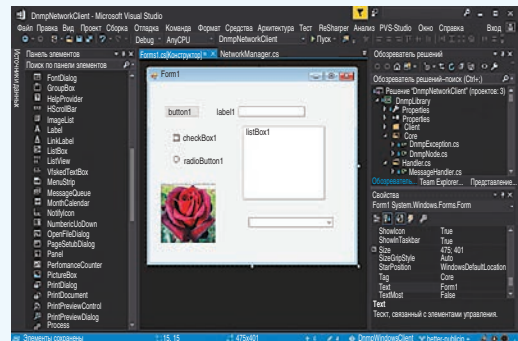
Delphi:



PascalABC:



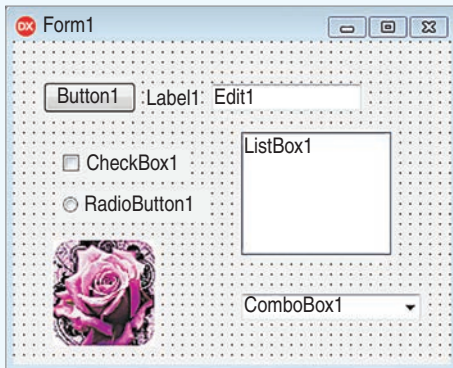
Visual Studio для мовы C#:



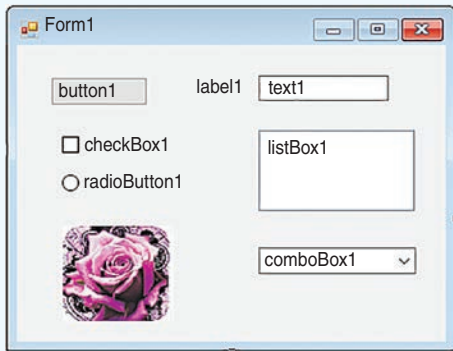
Прыклад 1.3. Асноўныя элементы інтэрфейса:

| Элемент кіравання | Імя |
|-----------------------|--------------------|
| Кнопка | Button |
| Надпіс | Label |
| Поле для ўводу тэксту | TextBox (Edit) |
| Флажок | CheckBox |
| Радыекнопка | RadioButton |
| Спіс | ListBox |
| Выпадаючы спіс | ComboBox |
| Малюнак | PictureBox (Image) |

Элементы кіравання на форме ў асяроддзі праграмавання Delphi:



Элементы кіравання на форме ў асяроддзі праграмавання PascalABC:



Шмат якія элементы кіравання ў розных асяроддзях маюць аднолькавыя ці сінанімічныя імёны (прыклад 1.3).

Ствараюцца аконныя дадаткі як праект і складаюцца з некалькіх файлаў. Выгляд акна будучага дадатку будзеца на форме. Для формы захоўваюцца два файлы — адзін змяшчае апісанне выгляду формы, іншы — апісанне дзеянняў пры выбары карыстальнікам таго ці іншага элемента кіравання. Галоўны файл праекта змяшчае апісанне яго структуры, а таксама каманды па стварэнні формы і запуску дадатку.

Усе элементы, змешчаныя на форме, і сама форма ўтвараюць сістэму аб’ектаў, якія ўзаемадзейнічаюць. Спосаб іх узаемадзеяння заснаваны на аб’ектна-арыентаваным праграмаванні.

Аб’ектна-арыентаванае праграмаванне (ААП) — тэхналогія стварэння праграм, заснаваная на выкарыстанні сістэмы аб’ектаў. Кожны аб’ект валодае наборам уласцівасцей, якія апісваюць яго стан, і метадаў, што характарызуюць яго паводзіны.

Аб’ект — сукупнасць даных і метадаў работы з імі.

Арганізацыя даных унутры аб’екта схавана ад карыстальніка. Даныя і спосабы іх чытання і запісу з’яўляюцца ўласцівасцямі аб’екта, іх можна змяняць. Метады — працэдуры і функцыі для апрацоўкі даных.

1.2. Падзеі

Арганізацыя ўзаемадзеяння паміж праграмай і карыстальнікам кіруецца **падзеямі**: карыстальнік можа націснуць на клавiшу мышы ці клавiатуры, увесці тэкст і інш.

Метад праграміравання, заснаваны на кіраванні падзеямі, называюць **падзейна-арыентаваным праграміраваннем**.

Кожная падзея звязана з нейкім аб'ектам, якому перадаецца кіраванне ў той момант часу, калі адбываецца падзея. Сярод асноўных падзей можна вылучыць тры катэгорыі: падзеі мышы, падзеі клавiатуры і сістэмныя падзеі (прыклады 1.4—1.6).

Працэдура (або функцыя), якую ініцыявала падзея, называецца **апрацоўшчыкам падзеі**.

Запушчаны на выкананне праект знаходзіцца ў рэжыме чакання, рэагуючы на падзеі, улiчаныя пры яго стварэнні, выкліканыя дзеяннямі карыстальніка ці ўзнікшымі ў самой праграме.

Аб'ектна-падзейная мадэль праграмы дапускае наступнае:

- стварэнне аб'ектаў з характэрнымі для іх уласцівасцямі і метадамі;
- апісанне падзей, пры якіх аб'ект можа выконваць алгарытм апрацоўкі даных.

Прыклад 1.4. Падзеі мышы ўзнікаюць у тым выпадку, калі карыстальнік выконвае якія-небудзь дзеянні з мышшу:

| | |
|-------------|---|
| Click | Націсканне левай кнопкі мышы |
| DoubleClick | Дваякая пстрычка левай кнопкай мышы |
| MouseDown | Націсканне на любую кнопку мышы. Параметры апрацоўшчыка падзеі дазваляюць вызначыць, якая з кнопак была націснута і ў якім пункце |
| MouseUp | Вызваленне кнопкі мышы, якая была націснута |
| MouseMove | Перамяшчэнне паказальніка мышы |

Прыклад 1.5. Падзеі клавiатуры адбываюцца пры націсканні клавiш на клавiатуры:

| | |
|----------|---|
| KeyPress | Націсканне клавiшы з тэкставым сімвалам |
| KeyDown | Націсканне любой клавiшы |
| KeyUp | Вызваленне клавiшы |

Прыклад 1.6. Сістэмныя падзеі кіруюцца функцыямі аперацыйнай сістэмы:

| | |
|--------|---|
| Paint | Узнікае, калі элемент неабходна перамаляваць |
| Resize | Адбываецца, калі памеры элемента змяняюцца |
| Enter | Узнікае, калі элемент кіравання становіцца актыўным |
| Leave | Адбываецца, калі элемент кіравання перастае быць актыўным |



1. Якія праграмы называюць аконнымі дадаткамі?
2. Што разумеюць пад падзейным праграміраваннем?
3. Якія тыпы падзей вы можаце назваць?



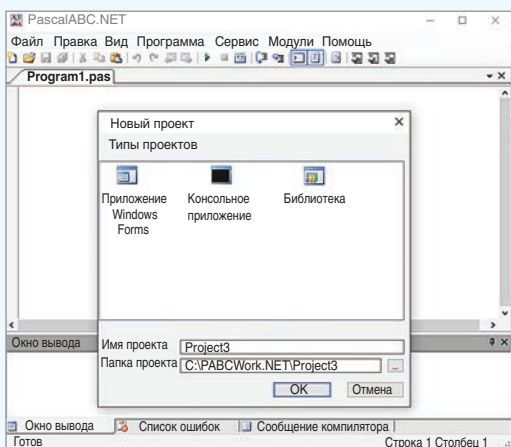
§ 2. Візуальнае асяроддзе распрацоўкі праграм

Работа па стварэнні аконных дадаткаў разглядаецца ў асяроддзі праграмавання PascalABC.Net.

Прыклад 2.1. Файлы праекта:

| Имя | Тип | Размер |
|----------------------------|-----------------------|--------|
| Project1.pabcproj 3 | PascalABC.NET Pr... | 2 КБ |
| Project1.pas | Файл "PAS" | 1 КБ |
| Unit1.fmabc 1 | Файл "FMABC" | 15 КБ |
| Unit1.Form1.inc | Free Pascal includ... | 1 КБ |
| Unit1.pas 2 | Файл PAS" | 1 КБ |


Прыклад 2.2. Стварэнне праекта ў PascalABC.Net:



Прыклад 2.3. Стварэнне і выкананне праекта:

1. Стварыць папку з імем Primer1.
 2. Стварыць прэкт.
 3. Запусціць праект на выкананне.
- Выгляд акона дадатку:



Для захавання змяненняў выкарыстоўваюць каманду **Сохранить** все (кнопка ).

2.1. Структура праекта

Пры стварэнні аконнага дадатку працуюць з праектам, які складаецца з некалькіх файлаў. У розных асяроддзях праграмавання праект можа складацца з рознай колькасці файлаў. Абавязковымі файламі з'яўляюцца наступныя:

- файл формы (1), які змяшчае апісанне выгляду акона дадатку;
- файл праграмнага модуля (2), які змяшчае апісанне функцый-апрацоўшчыкаў для аб'ектаў на форме;
- файл праекта (3), які дазваляе звязаць структурныя элементы праекта паміж сабой.

(Разгледзьце прыклад 2.1.)

Файлы аднаго праекта звычайна захоўваюцца ўнутры асобнай папкі. Пры кампіляцыі дадатку ствараецца файл з расшырэннем `.exe` і імем, якое супадае з імем праекта. Гэты файл запускае працуючы дадатак без загрузкі асяроддзя праграмавання. (Як скампіляваць дадатак, каб файл з расшырэннем `.exe` не выдаляўся пасля закрыцця акона, гл. у *Дадатку*, с. 102.)

Для стварэння праекта ў асяроддзі PascalABC.Net трэба выканаць каманды **Файл** → **Новый проект** → **Приложение Windows Form** (прыклад 2.2).

Пры стварэнні праекта файлы захоўваюцца аўтаматычна (прыклад 2.3).

2.2. Інтэрфейс асяроддзя праграмавання

Поўнае акно асяроддзя праграмавання PascalABC.Net пры стварэнні дадаткаў Windows Form можна паглядзець у *Дадатку* (с. 102).

Разгледзім асноўныя элементы.

Асноўнае меню і панэль хуткага доступу (прыклад 2.4) змяшчаюць каманды для кіравання праектам: захаванне, загрузка, выкананне і інш.

Форма (прыклад 2.5) служыць для візуальнага адлюстравання акна дадатку. У час праектавання дадатку на форме адлюстроўваецца кропкавая сетка, што дазваляе выраўноўваць кампаненты, што змяшчаюцца на форму.

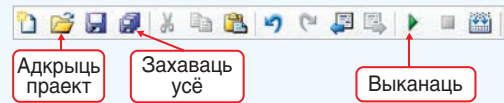
Інспектар аб'ектаў (прыклад 2.6) адлюстроўвае ўласцівасці (ці падзеі) выбранага аб'екта.

У левым слупку ўкладкі **Свойства** пералічаны ўсе ўласцівасці аб'екта, якімі карыстальнік можа кіраваць пры праектаванні дадатку. У правым слупку паказаны значэнні ўласцівасцей, якія могуць выбірацца са спіса ці ўводзіцца з клавіятуры.

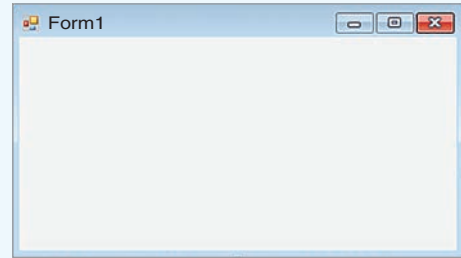
Укладка **События** змяшчае спіс падзей для аб'екта. Для кожнай падзеі можа быць вызначаны свой апрацоўшчык. Калі апрацоўшчык для падзеі вызначаны, насупраць падзеі будзе прапісана імя працэдуры (функцыі) апрацоўшчыка.

У ніжняй частцы інспектара аб'ектаў змешчана апісанне выбранай ўласцівасці ці апрацоўшчыка падзей.

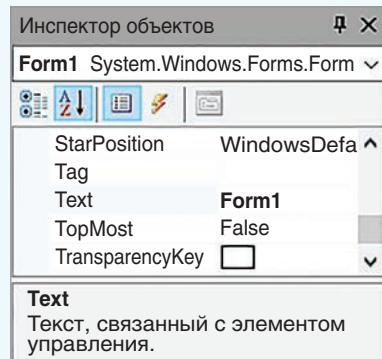
Прыклад 2.4. Меню і панэль хуткага доступу:



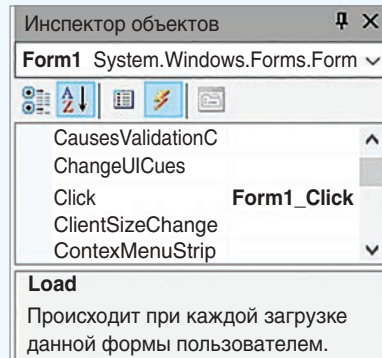
Прыклад 2.5. Форма:



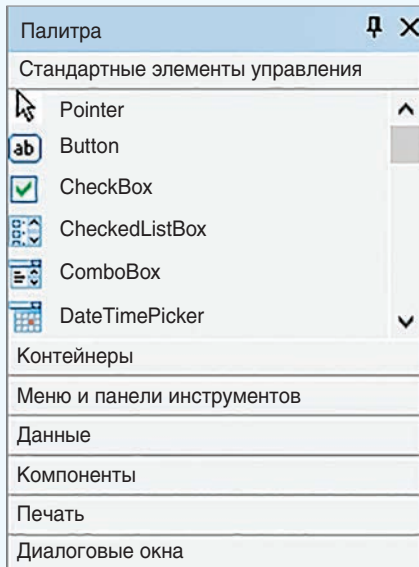
Прыклад 2.6. Інспектар аб'ектаў. Адлюстроўваюцца ўласцівасці формы:



Адлюстроўваюцца падзеі формы:



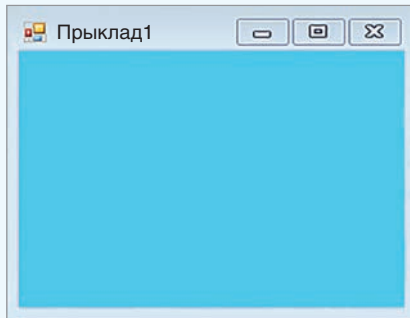
Прыклад 2.7. Палітра кампанентаў:



Прыклад 2.8. Змяненне ўласцівасцей формы ў інспектары аб'ектаў:

| Уласцівасць | Значэнне |
|-------------|---|
| Text | Прыклад 1 |
| BackColor | clAqua (выбраць са спіса на ўкладцы Интернет) |
| Size | 250; 250 |
| Location | 200; 200 |

Пасля змянення значэнняў уласцівасцей у інспектары аб'ектаў зменіцца выгляд формы:



Палітра кампанентаў (прыклад 2.7) змяшчае спіс візуальных кампанентаў, аб'яднаных у групы. Раскрыццё групы адбываецца па пстрычцы з назвай групы.

2.3. Работа з формай


Форма з'яўляецца аб'ектам і служыць для візуальнага адлюстравання акна дадатку. Як кожны аб'ект, форма валодае ўласцівасцямі (прыклад 2.8).

| Уласцівасць | Прызначэнне |
|-------------|--|
| Text | Заглавак формы адлюстроўваецца ў радку загатоўка акна пры запуску дадатку. Па змоўчанні — Form1 |
| BackColor | Колер формы. Можа быць выбраны адзін са стандартных (пералічаны ў спісе) ці зададзены ўручную трыма лікамі, якія адпавядаюць RGB |
| Size | Вышыня і шырыня формы. Можна выбраць два лікі праз «;» ці разгарнуць уласцівасць, націснуўшы значок  , і атрымаць магчымасць уводу значэнняў Width і Height |
| Location | Гарызантальная і вертыкальная каардынаты становішча верхняга левага вугла акна формы на экране. Можна выбраць два лікі праз «;» ці разгарнуць уласцівасць, націснуўшы значок  , і атрымаць магчымасць уводу значэнняў X і Y |
| (Name) | Імя (унутранае) формы. Выкарыстоўваецца ў праграмным кодзе для звароту да аб'екта. З'яўляецца ідэнтыфікатарам |

Для стварэння апрацоўшчыка падзей формы трэба ў інспектары аб'ектаў перайсці на ўкладку **События** (⚡), выбраць падзею. Працэдура генерыруецца аўтаматычна пры двайной пстрычцы мышшу ў пустым радку на супраць выбранай падзеі. Пасля гэтага асяроддзе пераключаецца на старонку, на якой пішацца код (прыклад 2.9).

Імя працэдуры-апрацоўшчыка складаецца з назвы кампанента, над якім адбываецца падзея, і назвы падзеі (`Form1_Click`).

Для кожнага аб'екта вызначаны апрацоўшчык па змоўчанні, які ствараецца пры двайной пстрычцы па аб'екце. Для формы такім апрацоўшчыкам будзе `Form1_Load` — падзея, якая адбываецца пры загрузцы формы.

Для пераключэння паміж акном праграмнага кода і канструктарам дызайну формы можна выкарыстаць укладкі **Дизайнер** і **Код** у верхняй частцы акна дадатку: .

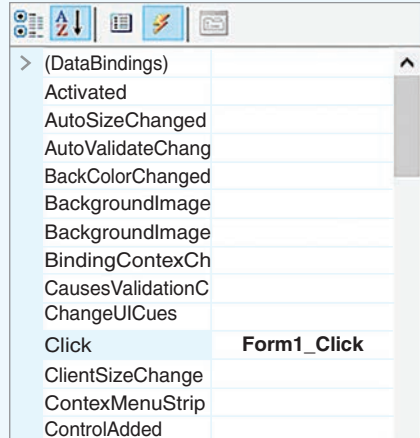
Пры стварэнні працэдуры-апрацоўшчыка ўласцівасці аб'ектаў можна змяняць праграмна. Для гэтага трэба звярнуцца да ўласцівасці па яе імені і прысвоіць новае значэнне. Напрыклад, для змянення колеру формы трэба запісаць наступную каманду:

```
BackColor := Color.Red;
```

Сістэма Pascal дазваляе спрасціць увод складаных імён у код праграмы. Пасля таго як вы набераце частку складанага імя, на экране з'явіцца спіс з усімі ўласцівасцямі і метадамі, якія належаць да гэтага аб'екта (прыклад 2.10).

Прыклад 2.9. Стварэнне апрацоўшчыка падзеі Click (клік левай клявішай мышы) для формы.

1. Выбар падзеі ў інспектары аб'ектаў:




2. Акно праграмнага кода з устаўленым апрацоўшчыкам:

```
Дизайнер Код
implementation
procedure Form1.Form1_click(sender: object; e: EventArgs);
begin
end;
end.
```

Прыклад 2.10. Падказка сістэмы пры ўводзе ўласцівасцей аб'екта:

```
procedure Form1.Form1_click(sender: obj
begin
  BackColor:=Color.r|
end;
end.
```



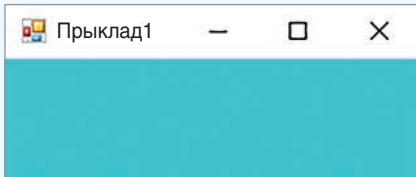
Калі вы ўведзяце першыя літары назвы ўласцівасці (метаду), то курсор перамясціцца ў спісе да тых уласцівасцей і метадаў, назвы якіх пачынаюцца на гэтыя літары. Пасля гэтага патрэбную ўласцівасць можна ўставіць у праграму пстрычкай мышы ці націсканнем клявішы Enter. Калі спіс не з'явіўся, яго можна выклікаць камбінацыяй клявіш Ctrl + прабел.

Прыклад 2.11. Код працэдуры-апрацоўшчыка:

```
procedure Form1.Form1_Click
(sender: Object; e: EventArgs);
begin
  BackColor := Color.Red;
end;
```

Каб пераканацца ў правільнасці работы праграмы, трэба запусціць праект і правесці, што пры выкананні пстрычкі мышшу па форме яе колер змяняецца на чырвоны.

Форма пасля запуску праекта:



Форма пасля пстрычкі мышы па ёй:



Прыклад 2.11. Стварыць апрацоўшчык падзеі для пстрычкі левай клавішай мышы па форме, у выніку якой колер формы павінен памянцца на чырвоны (працягнуць работу з прыкладам 2.8).

Этапы выканання задання

1. Перайсці на ўкладку **Events** у акне інспектара аб'ектаў.

2. Выканаць двайную пстрычку ў полі насупраць падзеі **OnClick**.

3. У акне рэдактара кода ў працэдуры

```
Form1.Form1_Click(sender: Object;
e: EventArgs);
упісаць каманду
BackColor := Color.Red;
```

Усе змяненні ўласцівасцей формы, якія выконвалі ў прыкладзе 2.8, можна запісаць праграма. Для гэтага ствараецца апрацоўшчык падзеі `Form1_Load`.

4. Захаваць змяненні ў праекце.



1. Якія элементы асяроддзя PascalABC.Net адлюстроўваюцца на экране пасля стварэння праекта?
2. Якія файлы ўваходзяць у склад дадатку, які ствараецца ў PascalABC.Net?
3. Для чаго прызначана форма?
4. Для чаго выкарыстоўваюць інспектар аб'ектаў?
5. Якія ўласцівасці формаў вы можаце назваць?
6. Як стварыць апрацоўшчык падзеі?



Практыкаванні

1 Унясіце змяненні ў праект з прыкладу 2.11 так, каб колер формы змяняўся выпадкова. Змяняць колер можна з дапамогай функцыі `FromArgb`. У гэтай функцыі чатыры параметры: празрыстасць (альфа-канал), інтэнсіўнасць чырвонага колеру, інтэнсіўнасць зялёнага колеру, інтэнсіўнасць сіняга колеру. Генерацыя выпадковых лікаў адбываецца наступным чынам. Спачатку ствараецца пераменная, якая з'яўляецца аб'ектам класа `Random` (каманда `var rnd: Random := new Random();`). Кожны новы выпадковы

лік можна атрымаць, звяртаючыся да метаду `next(x)`, дзе `x` задае паўінтэрвал $[0, x)$. Каманда змены колеру будзе выглядаць наступным чынам:

```
BackColor := Color.FromArgb(255, rnd.next(256), rnd.next(256), rnd.next(256));
```

2 Стварыце праект, у якім пры двайной пстрычцы мышы па форме яе памеры будуць павялічвацца на 5.

1. Стварыце і захавайце ў новай папцы праект.
2. Змяніце ўласцівасць `Text` формы на `Практыкаванне 2`.
3. Стварыце апрацоўшчык падзеі мышы `DbClick`.
4. Для змянення шырыні і вышыні формы можна выкарыстоўваць каманды:

```
Width := Width + 5;
Height := Height + 5;
```

5. Захавайце змены ў праекце.
6. Запусціце праект і правярце, як ён працуе.

3 Стварыце праект, у якім колер формы будзе змяняцца пры наведзенні на яе мышы, напрыклад з жоўтага на зялёны.

1. Змяніце ўласцівасць `Text` у формы на `Практыкаванне 3`.
2. Устанавіце жоўты колер формы.
3. Стварыце апрацоўшчыкі для дзвюх падзей мышы: `MouseEnter` і `MouseLeave`.
4. У кодзе падзеі `MouseEnter` устанавіце зялёны (Green) колер формы, а ў кодзе падзеі `MouseLeave` — жоўты (Yellow).
5. Захавайце змены ў праекце.
6. Запусціце праект і правярце, як ён працуе.

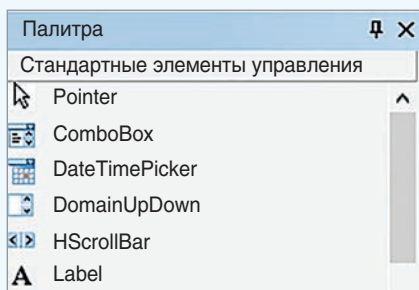
§ 3. Праектаванне інтэрфейса аконнага дадатку з выкарыстаннем элементаў кіравання

3.1. Асноўныя элементы кіравання

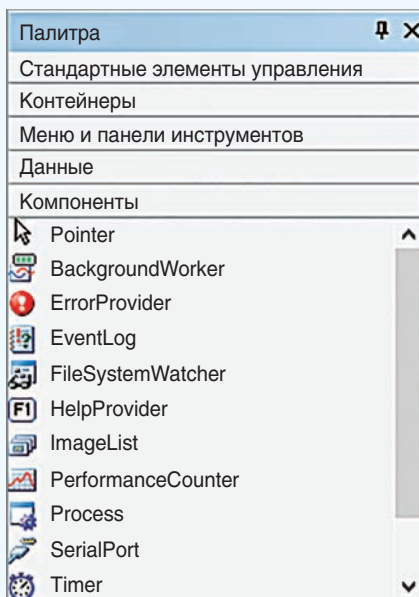
Элементамі кіравання называюцца аб'екты, якія выкарыстоўваюцца для адлюстравання даных ці арганізацыі ўзаемадзеяння паміж карыстальнікам і дадаткам з дапамогай мышы ці клавіятуры. Яны сабраны ў спецыяльныя бібліятэкі кампанентаў, якія АС выкарыстоўвае для забеспячэння аднастайнага інтэрфейсу

Для элементаў кіравання выкарыстоўваецца і іншая назва — віджэты. Слова ўжываецца прыкладна з 1920-х гг. у амерыканскім англійскім для абзначэння простага, але патрэбнай рэчы, маленькага вырабу. Адным з варыянтаў паходжання гэтага слова лічыцца словазліччэ «window gadget» (літар. «аконнае прыстасаванне»), якое таксама ўзнікла ў пачатку XX ст.

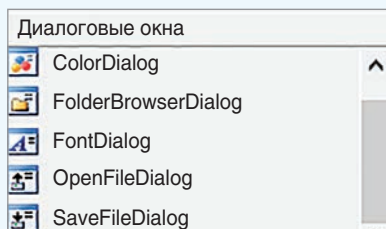
Прыклад 3.1. Палітра **Стандартные элементы управления:**



Прыклад 3.2. Палітра **Компоненты:**



Прыклад 3.3. Палітра **Диалоговые окна:**



прыкладных праграм. Найбольш распаўсюджанымі элементамі кіравання з'яўляюцца: кнопкі; палі, якія рэдагуюцца; спісы выбару; флажкі; пераключальнікі і г. д.

Кампаненты бібліятэкі змяшчаюцца на розных старонках палітры кампанентаў. Кожная старонка мае сваю назву. На старонцы **Стандартные элементы управления** (прыклад 3.1) змешчаны найбольш ужываемыя кампаненты:

- Кнопка (Button)
- Надпіс (Label)
- Поле для ўводу тэксту (TextBox)
- Флажок (CheckBox)
- Радыекнопка (RadioButton)
- Спіс (ListBox)
- Выпадаючы спіс (ComboBox)
- Малюнак (PictureBox)

Знешні выгляд гэтых кампанентаў на форме быў паказаны ў прыкладзе 1.4.

Іншыя старонкі палітры кампанентаў пададзены ў прыкладах 3.2 і 3.3.


Адным з кампанентаў палітры **Компоненты**, які найбольш часта выкарыстоўваецца, з'яўляецца кампанент Таймер (Timer).

Палітра **Меню и панели инструментов** мае кампаненты, патрэбныя для стварэння галоўнага меню праграмы ці кантэкставых меню для розных аб'ектаў, змешчаных на форму.

Палітры **Печать і Диалоговые окна** змяшчаюць кампаненты, якія забяспечваюць стандартныя дыялогі аперацыйнай сістэмы: адкрыццё і захаванне файла, выбар колеру, устаноўкі параметраў шрыфту, настройкі прынтара і кіраванне друкам.


Палітра **Данные** змяшчае кампаненты для работы з табліцамі баз даных.

3.2. Элемент кіравання кнопка (Button)

Кампанент **кнопка** належыць да элементаў кіравання. На панэлі кампанентаў **Стандартныя элементы управления** кнопка адлюстравана ў выглядзе , імя аб'екта — `button`. Кнопка, змешчаная на форму, атрымлівае імя `buttonN`, дзе `N` — нумар 1, 2, 3... (прыклад 3.4). Пры неабходнасці кнопку можна перамясціць у любое месца формы. Ключавыя вузлы дазваляць усталяваць патрэбны памер кнопкі.

Некаторыя ўласцівасці кнопкі пералічаны ў табліцы (прыклад 3.5).


Як бачна з табліцы, шмат якія ўласцівасці кнопкі супадаюць па імёнах і прызначэннях з уласцівасцямі формы, таму ў далейшым для кампанентаў будуць паказвацца толькі тыя ўласцівасці, якія адрозніваюцца ад ужо апісаных для іншых кампанентаў.

Асноўнай падзеяй кнопкі з'яўляецца `Click`. Для стварэння апрацоўшчыка падзеі `Click` для кнопкі можна паступіць гэтак жа, як і пры стварэнні аналагічнага апрацоўшчыка для формы: выбраць падзею на ўкладцы **События**  і выканаць двайную пстрычку ў полі на супраць падзеі `Click`. Можна проста выканаць двайную пстрычку па кнопцы. (Для формы асноўнай падзеяй з'яўляецца падзея `Load`, таму пры двайной пстрычцы па форме ствараецца апрацоўшчык падзеі `Load`.)

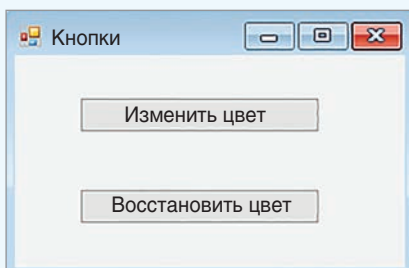
Прыклад 3.4. Кампанент *кнопка* на форме:



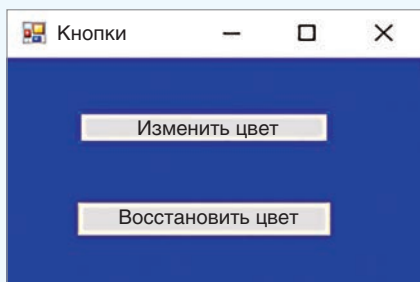
Прыклад 3.5. Уласцівасці кнопкі:

| Некаторыя ўласцівасці кнопкі Button | |
|-------------------------------------|--|
| Уласцівасць | Прызначэнне |
| Text | Загалолак (знешняе імя) кнопкі, тэкст, які адлюстроўваецца на кнопцы. Па змоўчанні — <code>button1</code> |
| Font | Уласцівасці шрыфту для подпісу загаловака. Уласцівасць <code>Font</code> з'яўляецца складанай, пра што сведчыць значок  , пры націсканні на які раскрываюцца ўсе ўласцівасці шрыфту |
| Height | Вышыня кнопкі |
| Weight | Шырыня кнопкі |
| Left | Гарызантальная каардыната становішча верхняга левага вугла кнопкі на форме |
| Top | Вертыкальная каардыната становішча верхняга левага вугла кнопкі на форме |
| (Name) | Імя (унутранае) кнопкі. Выкарыстоўваецца ў праграмным кодзе для звароту да аб'екта. З'яўляецца ідэнтыфікатарам |
| Enabled | Значэнне <code>True</code> гэтай уласцівасці забяспечвае даступнасць кнопкі для мышы ці клавіятуры |
| Visible | Значэнне <code>True</code> гэтай уласцівасці забяспечвае бачнасць кнопкі ў час выканання дадатку |


Прыклад 3.6. Знешні выгляд формы ў рэжыме канструктара дызайну:



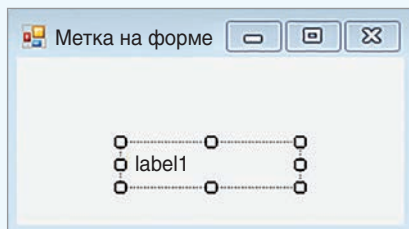
Знешні выгляд формы пры выкананні:



Уласцівасць кнопкі `BackColor` дазваляе мяняць яе колер. Для звароту да гэтай уласцівасці ў праграме выкарыстоўваюць запіс: `button1.BackColor`.

Уласцівасць кнопкі `BackgroundImage` дазваляе ўстаўіць на кнопку відарыс, які захоўваецца ў графічным файле. Для ўстаўкі можна выкарыстаць кнопку . Далей выбраць файл з маляўніцтвам.

Прыклад 3.7. Кампанент *метка* на форме:



Прыклад 3.6. Стварыць праект, змясціўшы на форме дзве кнопкі. Пры націсканні на адну з іх колер формы павінен змяніцца на сіні, а пры націсканні на другую — павінен аднавіцца зыходны колер.

Этапы выканання задання

1. Стварыць на форме дзве кнопкі.
2. Змяніць уласцівасць `Text` у кнопкі `button1` на **Изменить цвет**.
3. Змяніць уласцівасць `Text` у кнопкі `button2` на **Восстановить цвет**.
4. Стварыць апрацоўшчык падзеі `Click` для кнопкі `button1` і змяніць колер формы. Каманда

```
BackColor := Color.Blue;
```

5. Стварыць апрацоўшчык падзеі `Click` для кнопкі `button2` і змяніць колер формы на зыходны (назва колеру формы паказана ў полі `Color` інспектара аб'ектаў). Каманда

```
BackColor := SystemColors.Control;
```

6. Захаваць змены ў праекце.

Назва колеру `SystemColors.Control` задае не нейкі пэўны колер. Гэта колер элемента кіравання, зададзены колеравай схемай `Windows`. Таму ён не абавязкова будзе шэрым.


3.3. Элемент кіравання *метка* (`Label`)

Кампанент *метка* прызначаны для адлюстравання тэксту на форме. На панэлі кампанентаў **Стандартныя элементы управления** *метка* адлюстравана ў выглядзе **A** `Label`, імя аб'екта — `label`. *Метка*, змешчаная на форму, атрымлівае імя `labelN`, дзе `N` — нумар 1, 2, 3... (прыклад 3.7).


Некаторыя ўласцівасці меткі, адрозныя ад уласцівасцей кнопкі, пералічаны ў табліцы (прыклад 3.8). Асноўнай падзеяй для меткі з’яўляецца Click.

Прыклад 3.9. Стварыць праект, у якім апісана магчымасць выконваць наступныя дзеянні: пасля запуску праграмы ў акне з імем «**Работаем с кнопкой и меткой**» пры пстрычцы мышы па кнопцы «**Приветствие**» з’яўляецца паведамленне «**Прывітанне, свет!**».


Этапы выканання задання

1. Змяніць уласцівасць Text формы на «**Работаем с кнопкой и меткой**».
2. Дабавіць на форму кнопку button1.
3. Змяніць уласцівасць Text кнопкі на «**Приветствие**».
4. Дабавіць на форму метку label1.
5. Змяніць уласцівасці шрыфту для кампанента label1. Націснуць кнопку  у полі Font (колер шрыфту — сіні, памер — 20, стыль — тлусты курсіў).
6. Ачысціць поле Text у меткі.
7. Устанавіць значэнне true ва ўласцівасці меткі Autosize.
8. У апрацоўшчык падзеі Click для кнопкі button1 упісаць каманду label1.Text := 'Прывітанне, свет!';

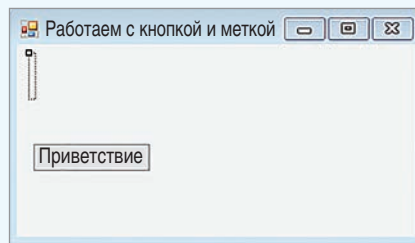
3.4. Элемент кіравання *тэкставае поле (Edit)*

Тэкставае поле — кампанент, прызначаны для ўводу і вываду тэкставай інфармацыі. На панэлі кампанентаў **Стандартныя элементы управлення** тэкставае поле адлюстравана ў выглядзе , імя аб’екта — TextBox.

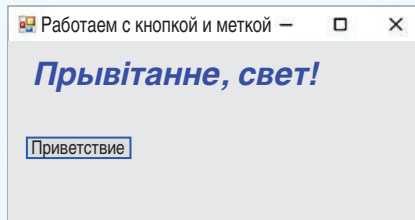
Прыклад 3.8. Уласцівасці меткі:

| Уласцівасць | Прызначэнне |
|-------------|---|
| Text | Адлюстроўвае ўведзены тэкст на форме |
| BackColor | Устанаўлівае колер фону меткі, які па змоўчанні супадае з колерам формы. Фон меткі можна зрабіць празрыстым, задаўшы ўласцівасці BackColor значэнне Color.Transparent |
| AutoSize | Значэнне true гэтай уласцівасці прыводзіць да аўтаматычнага змянення памераў меткі ў адпаведнасці з даўжынёй тэксту |
| TextAlign | Выраўноўванне тэксту адносна меж меткі:  |

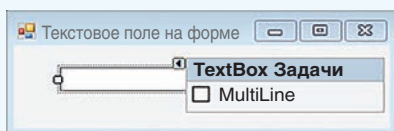
Прыклад 3.9. Форма на этапе канструявання:



Дадатак у час работы:



Прыклад 3.10. Кампанент *тэкставае поле* на форме:



Прыклад 3.11. Уласцівасці тэкставага поля:

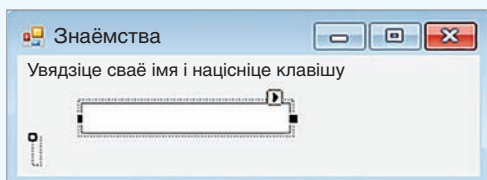
| Уласцівасць | Прызначэнне |
|-------------|--|
| BorderStyle | Вызначае мяжу вакол тэкставага поля: None (няма мяжы), FixedSingl ці Fixed3d (ёсць мяжа) |
| MaxLength | Абмяжоўвае колькасць сімвалаў, якія можна ўвесці ў TextBox |
| ReadOnly | Значэнне true забараняе рэдагаванне тэксту, які адлюстроўваецца ў TextBox |
| Text | Змяшчае тэкст, які ўводзіцца ці выводзіцца |

Тэкставае поле часта называюць тэкставым рэдактарам, паколькі яно забяспечана такімі функцыямі, як:

- капіраванне вылучанага тэксту ў буфер абмену (Ctrl+C);
- выразанне вылучанага тэксту ў буфер абмену (Ctrl+X);
- устаўка тэксту з буфера абмену ў пазіцыю курсора (Ctrl+V);
- адмена апошняй каманды рэдагавання (Ctrl+Z).

Уласцівасць Multiline вызначае, якім будзе рэдактар — аднарадковым ці шматрадковым.

Прыклад 3.12. Форма на этапе канстрування:



Тэкставае поле, змешчанае на форме, атрымлівае імя TextBoxN, дзе N — нумар 1, 2, 3... (прыклад 3.10).

У адрозненне ад раней разгледжаных кампанентаў, уласцівасць Text у тэкставага поля па змоўчанні пустая (у іншых — супадае з імем кампанента). Некаторыя ўласцівасці кампанента TextBox прыведзены ў тэблцы (прыклад 3.11).

Значэнне ўласцівасці Text кампанента *тэкставае поле* можа змяняцца праграма ці пры ўводзе з клавіятуры. Асноўнай падзеяй для TextBox з'яўляецца TextChanged, якая адбываецца пры змяненні кампанента. Найбольш часта праграмуюць падзею KeyPress, якая дазваляе вызначыць, што за клавіша была націснута.

Прыклад 3.12. Стварыць праект, у якім карыстальніка папросяць увесці яго імя, а потым, пасля націскання клавішы Enter, будзе выдадзена паведамленне «Імя, прыемна з Вамі пазнаёміцца!».

Этапы выканання задання

1. Змяніць уласцівасць Text у формы на «Знаёмства».
2. Змясціць на форме дзве меткі і тэкставае поле.
3. Змяніць уласцівасць Text у label1 на «Увядзіце сваё імя і націсніце клавішу Enter».
4. Ачысціць поле ўласцівасці Text у Label2.
5. Напісаць апрацоўшчык падзеі KeyPress для кампанента Edit1, які будзе правяраць націсканне клавішы ўводу (код клавішы Enter — 13). Калі клавіша націснута, то змяніць уласцівасць Text у label2:

```
if e.KeyChar = #13 then
    label2.Text := TextBox1.Text +
    ', прыемна з Вамі пазнаёміцца!';
```

Тэкставае поле `TextBox` выкарыстоўваецца таксама для ўводу і вываду лікаў. Пры гэтым неабходна выкарыстоўваць функцыі для ператварэння радкоў у лікі і лікаў у радкі. Гэтыя функцыі прыведзены ў табліцы:

| Назва функцыі | Дзеянне |
|------------------------|--|
| Увод з дапамогай Edit | |
| StrToInt | Ператварэнне радка ў цэлы лік |
| StrToFloat | Ператварэнне радка ў значэнне з плаваючай коскай |
| Вывад з дапамогай Edit | |
| IntToStr | Ператварэнне цэлага ліку ў радок |
| FloatToStr | Ператварэнне рэчаіснага ліку ў радок |

У рускамоўнай версіі Windows у якасці раздзяляльніка цэлай і дробавай часткі ліку па змоўчанні выкарыстоўваецца коска. Калі пры ўводзе лікаў у тэкставыя палі выкарыстаць кропку, то будзе ўзнікаць памылка ператварэння тыпаў.

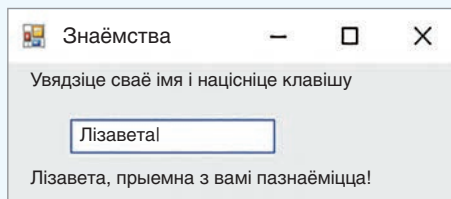
Прыклад 3.13. Стварыць праект, у якім карыстальнік зможа ўвесці лік, атрымаць яго значэнне ў квадраце і квадратны карань з гэтага ліку.

Этапы выканання задання

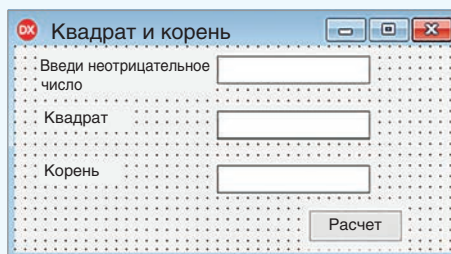
1. Змяніць уласцівасць `Text` у формы на «**Квадрат и корень**».
2. Змясціць на форме тры меткі, тры тэкставыя палі і кнопку.

Прыклад 3.12. Працяг.

Дадатак у час работы:



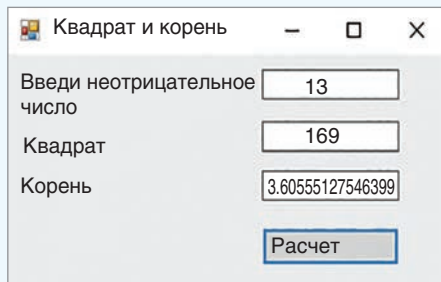
Прыклад 3.13. Форма на этапе канстрування:



Апрацоўшчык падзеі `OnClick` для `Button1`:

```
procedure Form1.button1_Click
(sender: Object; e: EventArgs);
var a, b: integer;
    c: real;
begin
    a := StrToInt(TextBox1.Text);
    b := a * a;
    c := sqrt(a);
    TextBox2.Text := IntToStr(b);
    TextBox3.Text := FloatToStr(c);
end;
```

Дадатак у час работы:



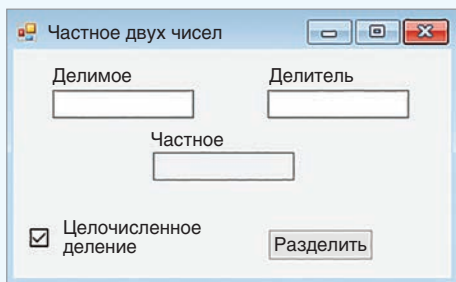
Прыклад 3.14. Кампаненты флажок і радыёкнопка на форме:



Прыклад 3.15. Уласцівасці кампанента флажок:

| Уласцівасць | Прызначэнне |
|-------------|--|
| Checked | Значэнне true у гэтай уласцівасці паказвае, што ўстаноўлена птушка — <input checked="" type="checkbox"/> , пустое акно індикатара — <input type="checkbox"/> адпавядае значэнню false |
| Text | Надпіс каля кампанента checkBox |
| RightToLeft | Вызначае, з якога боку кнопкі размяшчаецца надпіс: Yes (надпіс злева), No (надпіс справа) |
| CheckState | Вызначае стан кампанента: Unchecked — не вылучаны, Checked — вылучаны, Indeterminate (<input type="checkbox"/>) — пра-межкавы стан. Першыя два станы адпавядаюць уласцівасці Checked |

Прыклад 3.16. Форма на этапе канстрування:



3. Змяніць уласцівасць Text у Label1 на «Введите неотрицательное число».

4. Змяніць уласцівасць Text у Label2 на «Квадрат».

5. Змяніць уласцівасць Text у Label3 на «Корень».

6. Змяніць уласцівасць Text у Button1 на «Расчет».

7. Напісаць апрацоўшчык Click для кнопкі.

3.5*. Элементы кіравання флажок (CheckBox) і пераклячальнік (RadioButton)

Флажок выкарыстоўваецца ў дадатках для ўключэння ці выключэння якіх-небудзь опцый. На панэлі кампанентаў **Стандартныя элементы управления** флажок адлюстраваны ў выглядзе CheckBox, імя аб'екта — CheckBox. Флажок, змешчаны на форму, атрымлівае імя checkBoxN, дзе N — нумар 1, 2, 3... (прыклад 3.14). Некаторыя ўласцівасці кампанента checkBox прыведзены ў тэбліцы (прыклад 3.15).

Аналагічным чынам выкарыстоўваецца кампанент **Переключатель (радыёкнопка)**. На панэлі кампанентаў Standard радыёкнопка адлюстравана ў выглядзе RadioButton, імя аб'екта — RadioButton. Пераклячальнік, змешчаны на форму, атрымлівае імя radioButtonN, дзе N — нумар 1, 2, 3... (прыклад 3.14).

Звычайна радыёкнопкі ўтвараюць групы ўзаемазвязаных індикатараў, якія дазваляюць выбраць толькі адну з некалькіх узаемавыключальных альтэрнатыв. Пры змяшчэнні на фор-

ме некалькіх пераклучальнікаў уключаным павінен быць толькі адзін з іх (кантэйнер GroupBox).

Прыклад 3.16. Стварыць праект для вылічэння дзелі ад дзялення аднаго цэлага ліку на іншае. Лікі задаюцца ў тэкставых палях. Вынік вылічваецца пры націсканні на кнопку «Частное» і змяшчаецца ў трэцяе тэкставае поле. Вынік залежыць ад стану флажка.

Этапы выканання задання

1. Змясціць на форму тэкставыя палі (3), надпісы (3), флажок і кнопку.

2. Для кампанента textBox3 уставаць значэнне true для ўласцівасці ReadOnly.

3. Змяніць уласцівасць Text у кампанентаў label («Делимое», «Делитель», «Частное»).

4. Змяніць уласцівасць Text кампанента button1 на «Разделить».

5. Змяніць уласцівасць Text кампанента checkBox1 на «Целочисленное деление».

6. Напісаць апрацоўшчык падзеі Click для кампанента button1.

6.1. Правярыць, што палі кампанентаў textBox1 і textBox2 не пустыя. Інакш вывесці паведамленне «Адно з палёў не запоўнена».

6.2. Правярыць стан пераклучальніка checkBox. Калі ён уключаны, то выканаць цэлалікавае дзяленне, інакш звычайнае дзяленне.

6.3. Вывесці вынік.

7. Выканаць праграму для розных значэнняў. Правярыць работу дадатку, калі адно з палёў textBox1 або textBox2 (ці абодва палі) пустыя.

Прыклад 3.16. Працяг.

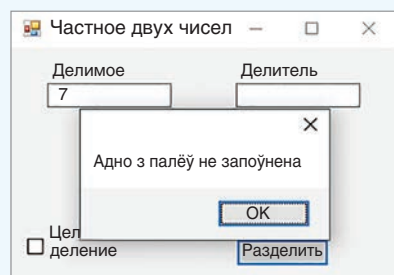
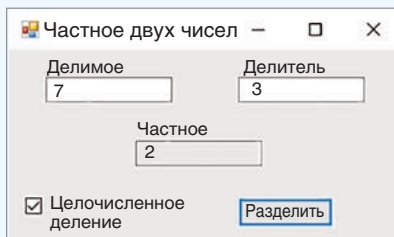
Апрацоўшчык падзеі OnClick для Button1:


```

procedure Form1.button1_Click
(sender: Object; e: EventArgs);
var a, b, c : integer;
    d : real;
begin
  if (TextBox1.Text <> '') and
    (TextBox2.Text <> '') then
    begin
      a := StrToInt(TextBox1.Text);
      b := StrToInt(TextBox2.Text);
      if CheckBox1.checked then
        begin
          c := a div b;
          TextBox3.Text := IntToStr(c);
        end
      else
        begin
          d := a / b;
          TextBox3.Text := FloatToStr(d);
        end;
      end
    else
      MessageBox.Show('Адно з палёў не запоўнена');
    end;
end;

```

Дадатак у час работы:



-  1. Якія кампаненты залічваюць да элементаў кіравання?
 2. Як змясціць кампанент на форму?
 3. Якія ўласцівасці кампанента button вы можаце назваць?
 4. Якая падзея з'яўляецца асноўнай для кампанента button?
 5. Для чаго прызначаны кампанент label?
 6. У якіх выпадках выкарыстоўваецца кампанент TextBox?
 7. Для чаго прызначаны кампаненты checkBox і radioButton?

  **Практыкаванні**

- 1 Адкрыце праект з прыкладу 3.9 і дапоўніце яго кнопкай «Ачысціць»¹. Кнопка «Ачысціць» павінна ачышчаць тэкст меткі (Уласцівасці Caption прысвоіць значэнне пустога радка: ""). Зрабіце выпадковым выбар колеру і памеру шрыфту ў меткі.
- 2 Адкрыце праект з прыкладу 3.12 і дабаўце на форму тры меткі і дзве кнопкі.
1. Змяніце ўласцівасці кампанентаў у адпаведнасці з указаннямі ў тэбліцы.

| Кампанент | Уласцівасць | Значэнне ўласцівасці |
|-----------|-------------|---|
| button1 | Text | Так |
| button1 | Visible | False |
| button2 | Text | Не |
| button2 | Visible | False |
| label3 | Text | Вы хочаце працаваць у IT? |
| label3 | Visible | False |
| label4 | Text | Выдатна! Паспехаў у вывучэнні інфарматыкі! Яна Вам спатрэбіцца! |
| label4 | Visible | False |
| label5 | Text | Іншыя прафесіі таксама патрабуюць ведаў інфарматыкі |
| label5 | Visible | False |

2. Дабаўце ў апрацоўшчык падзеі KeyPress каманду, якая робіць надпіс Label3 і кнопкі бачнымі.

¹ Перад змяненнем скапіруйце праект у новую папку

```

procedure Form1.textBox1_KeyPress(sender: Object;
e: KeyPressEventArgs);
begin
  if e.KeyChar = #13 then
    begin
      label2.Text := TextBox1.Text + ', прыемна з Вамі пазнаёміцца!';
      Label3.Visible := True;
      Button1.Visible := True;
      Button2.Visible := True;
    end;
  end;

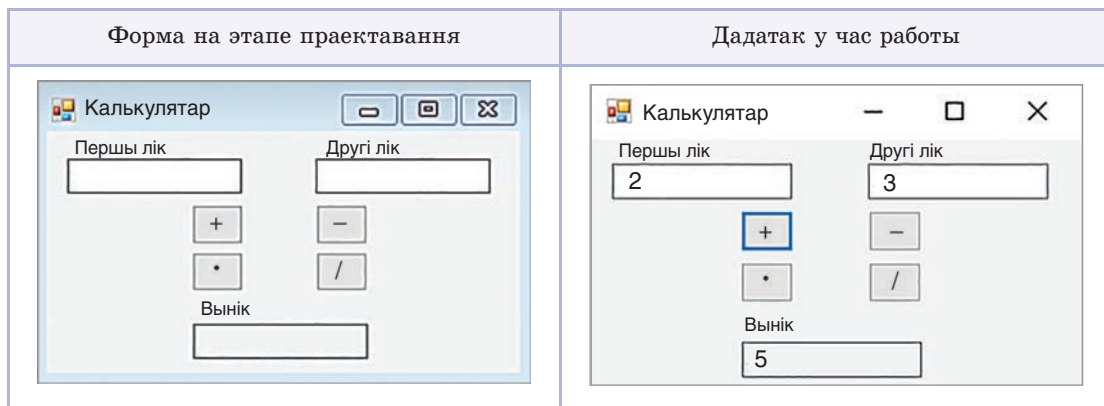
```

3. Напішыце апрацоўшчыкі Click для кнопак Button1 і Button2. Зрабіце бачнымі адпаведныя надпісы.

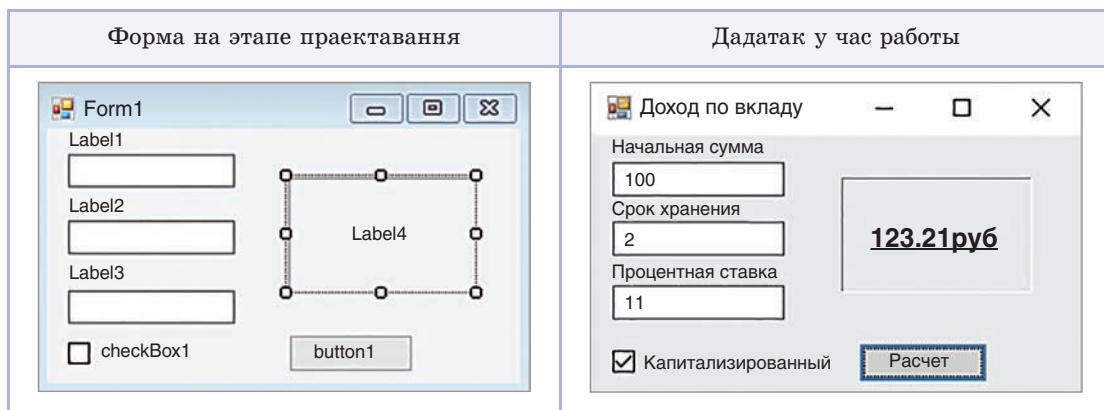
| Форма на этапе праектавання | Дадатак у час работы пасля запуску |
|---|--|
| | |
| Дадатак у час работы да адказу на пытанне | Дадатак у час работы пасля адказу на пытанне |
| | |

4*. Дабаўце ў дадатак яшчэ адно пытанне. Форму адказу выберыце самастойна.

3 Стварыце праект **Калькулятар**. Змясціце на форме тры палі TextBox і тры надпісы: «Першы лік», «Другі лік», «Вынік». Дабаўце кнопкі для вылічэння сумы, рознасці, здабытку і дзелі. Забараніце рэдагаванне ў полі з адказам. *Дабаўце праверку дзялення на нуль.



4 Стварыце праект, у якім вылічваецца даход па ўкладзе. Праграма павінна забяспечваць разлік грашовых сум для простых ці капіталізаваных укладаў. Калі ўклад просты, то працэнтная стаўка налічваецца ад зыходнай сумы, і кожны месяц яна аднолькавая, а калі капіталізаваны, то працэнтная стаўка налічваецца кожны месяц ад сумы ўкладу ў папярэднім месяцы.




Праверце, ці запоўнены палі з зыходнымі данымі. Калі не, то выведзіце адпаведнае паведамленне.

5 Зрабіце «кнопку, якая ўцякае», г. зн. пры наведзенні паказальніка мышы на кнопку яна павінна выпадаковым чынам памяняць месца.


6 Дабаўце ў практыкаванне 5 кнопку «**Домой**», якая павінна перамясціць «кнопку, якая ўцякае», у верхні левы вугал формы.

§ 4. Элементы кіравання для работы з графікай

4.1. Элемент кіравання для ўстаўкі малюнка (PictureBox)

Пры стварэнні дадаткаў нярэдка ўзнікае неабходнасць упрыгожыць іх графічным відарысам. У гэтым выпадку можна выкарыстаць кампанент **відарыс**. На панэлі кампанентаў **Стандартныя элементы управления** кампанент відарыс пададзены ў выглядзе , імя аб'екта PictureBox. Кампанент PictureBox, змешчаны на форме, атрымлівае імя PictureBox N, дзе N — нумар 1, 2, 3... (прыклад 4.1).

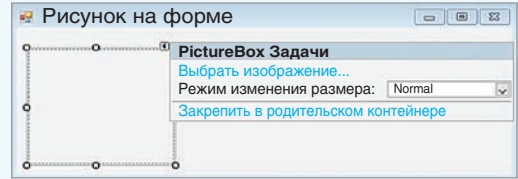
Кампанент уяўляе сабой кантэйнер, у які змяшчаецца відарыс. Некаторыя ўласцівасці кампанента PictureBox прыведзены ў табліцы (прыклад 4.2).

Выкарыстоўваючы ўласцівасць Image, можна выбраць і загрузіць відарыс на этапе праектавання дадатку. Відарыс можа быць выбраны ў кантэйнеры пры націсканні на кнопку  у правым верхнім вугле кампанента. У гэтым выпадку малюнак захоўваецца ў файле формы і для работы дадатку асобнага файла з малюнкам не патрабуецца.

Кампанент падтрымлівае ўстаўку малюнкаў у фарматах JPEG, PNG, BMP, GIF. Калі патрабуецца апрацоўка відарыса (любыя змяненні малюнка), то малюнак павінен быць захаваны ў фармаце BMP. Для малюнкаў фармату PNG ці GIF з празрыстым фонам пры загрузцы захоўваецца празрыстасць.

Малюнак можна загрузіць як фон формы. Для гэтага прызначана ўласцівасць формы BackgroundImage.

Прыклад 4.1. Кампанент *відарыс* на форме:

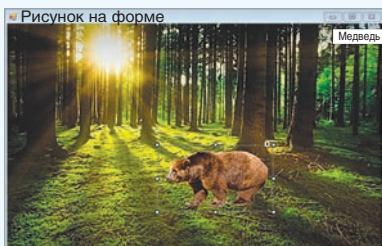


Прыклад 4.2. Некаторыя ўласцівасці кампанента *відарыс*.

| Уласцівасць | Прызначэнне |
|-------------|---|
| Image | Выкарыстоўваецца для адлюстравання відарысаў |
| SizeMode | Вызначае спосаб адлюстравання малюнка: <i>Normal</i> — левы верхні вугал малюнка сумешчаны з левым верхнім вуглом кантэйнера; <i>StretchImage</i> — малюнак упісваецца ў кантэйнер; <i>AutoSize</i> — памер падганяецца пад памер малюнка; <i>CenterImage</i> — малюнак будзе адцэнтраваны адносна кампанента; <i>Zoom</i> — пры змяненні памераў кантэйнера будучы захоўвацца прапарцыя малюнка |
| Margin | Вызначае прамежак паміж палямі відарыса і палямі іншага кампанента (значэнні All, Left, Top, Right, Bottom) |
| BackColor | Колер фону відарыса. Можа быць празрыстым |

Уласцівасць Image кампанента PictureBox валодае метадам Save, які выкарыстоўваецца для захавання відарыса. Метад Load кампанента PictureBox можа быць выкарыстаны для загрузкі відарыса пры адкрыцці дадатку.

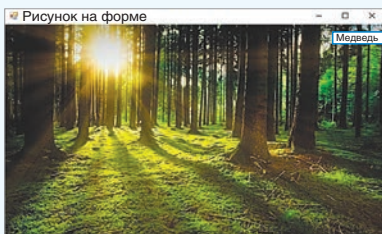
Прыклад 4.3. Форма на этапе канстрування:



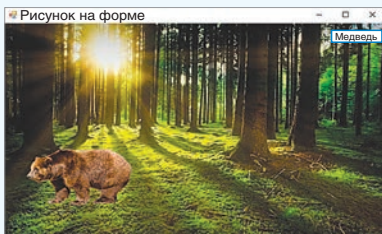
Апрацоўшчык падзеі Click для Button1:

```
procedure Form1.button1_Click
(sender: Object; e: EventArgs);
begin
    PictureBox2.Top := 220;
    var rnd: Random := new Random();
    PictureBox2.Left := rnd.
    Next(300);
    PictureBox2.Visible := True;
end;
```

Дадатак у час работы
Да націскання на кнопку:



Пасля націскання на кнопку:



Паколькі гарызантальнае становішча мядзведзя задаецца выпадковым чынам, то пры кожным націсканні на кнопку мядзведзь будзе прарысаваны ў новым месцы.

Прыклад 4.3. Стварыць праект, змясціць у ім фонавы відарыс на форме. Пры націсканні на кнопку паверх фонавага відарыса павінен з'явіцца іншы відарыс.

Этапы выканання задання

1. Устанавіць памеры формы Height = 450, Width = 670.

2. Загрузіць фонавы відарыс для формы. Задаць для ўласцівасці формы BackgroundImageLayout значэнне Stretch.

3. Змясціць на форму кампанент відарыс і кнопку.

4. Для кампанента PictureBox устанавіць значэнне для ўласцівасці Visible = False (відарыс нябачны пры запуску дадатку). Памеры Height = 120, Width = 200. Уласцівасць SizeMode = StretchImage.

5. Загрузіць відарыс у кампанент PictureBox. Відарыс можа быць фармату PNG, ці GIF з празрыстым фонам, ці фармату BMP з фонам аднароднага колеру. Уласцівасць BackColor = Transparent.

6. Напісаць апрацоўшчык падзеі OnClick для кампанента Button1.

Калі пры запуску дадатку відарыс мігціць, то спыніць мігаценне можна з дапамогай уключэння двайной буферызацыі:

```
DoubleBuffered := true;
```

Гэта каманда павінна быць прапісана ў апрацоўшчыку падзеі Load для формы.

4.2. Пабудова графікаў функцый

Прастора імянаў System.Drawing забяспечвае доступ да функцыянальных магчымасцей графічнага інтэрфейса Windows. Клас Graphics змяш-

чае метады для малявання графічных прымітываў.

Асноўная падзея для пабудовы відарысаў — Paint.

Калі ўзнікае неабходнасць маляваць па кропках, то для гэтага выкарыстоўваецца клас Bitmap (відарыс з пунктаў). Кожны пункт мае каардынаты X і Y . Сістэма каардынат такая ж, як і для графічнага акна PascalABC.Net — пункт з каардынатамі $(0, 0)$ змешчаны ў верхнім левым вугле, вось OY накіравана ўніз. Кожны пункт мае каардынаты X і Y . Каардынаты вымяраюцца ў пікселях. Найважнейшая ўласцівасць пікселя — яго колер. Для задання колеру ў PascalABC.Net можна выкарыстаць некалькі спосабаў (прыклад 4.4). Кропка адлюстроўваецца з дапамогай каманды `SetPixel(x1, y1, Color);`

Клас Graphics змяшчае вялікую колькасць уласцівасцей і метадаў, што дазваляюць будаваць відарысы. Шмат якія з метадаў Graphics супадаюць з працэдурамі, што выкарыстоўваліся ў бібліятэцы GraphABC асяроддзя праграмавання PascalABC.Net. Апісанне гэтых метадаў прыведзена ў дадатку.

Прыклад 4.5. Стварыць праект і пабудаваць графік функцыі $y = x \sin x$ на прамежку, зададзеным карыстальнікам.

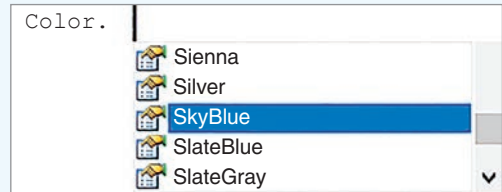
Этапы выканання задання

1. Змясціць на форму кампаненты: PictureBox, два кампаненты Label, два кампаненты TextBox і кампанент Button.

2. Змяніць уласцівасці Text у кампанентаў Label1, Label2 на x_0 і x_n адпаведна.

Прыклад 4.4. Спосабы задання колеру ў PascalABC.Net:

1. Заданне колеру з дапамогай канстант. Канстанты захоўваюць імя колеру (напрыклад, Color.SkyBlue — нябесна-сіні, Color.Red — чырвоны). Магчымыя значэнні з'яўляюцца ў выпадаючым спісе пасля ўводу ў праграму ключавога слова Color.

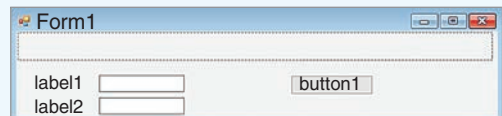


2. Для задання колеру можна выкарыстаць функцыю `Color.FromArgb(A, R, G, B)`, параметры якой задаюць празрыстасць (альфа-канал) і інтэнсіўнасць чырвонага, сіняга і зялёнага колераў адпаведна. Значэнні параметраў могуць змяняцца ад 0 до 255.

3. Заданне колеру з дапамогай шаснаццацярычнага лікаў. Пары лічбаў шаснаццацярычнага ліку задаюць празрыстасць і інтэнсіўнасць чырвонага, зялёнага і сіняга колераў адпаведна. Атрыманы шаснаццацярычны лік запісваецца як параметр функцыі `FromArgb`. Напрыклад, `Color.FromArgb($FF0000FF)` — сіні колер. Значэнні для іншых колераў: `$FFFFFF0000` — чырвоны, `$FF00FF00` — зялёны, `$FF000000` — чорны, `$FFFFFFFF` — белы.

Шаснаццацярычныя лікі можна перавесці ў дзесятковую сістэму злічэння і карыстацца гэтымі значэннямі. Напрыклад, `Color.FromArgb(4278190335)` — сіні колер.

Прыклад 4.5. Форма на этапе канструявання:



Прыклад 4.5. Працяг.

Апрацоўшчык падзеі Click для кампанента Button1:

```

procedure Form1.button1_Click
(sender: Object; e: EventArgs);
var x, y, h, k, x0, xn : real;
    x1, y1, n, c_x, c_y : integer;
    gr: Graphics;
    bm: Bitmap;
    p_c: Pen;
begin
//падрыхтоўка графічнай вобласці
для малявання па пікселях
bm := new Bitmap (PictureBox1.Width,
    PictureBox1.Height);
pictureBox1.Image := (Image) (bm);
gr := Graphics.FromImage
    (pictureBox1.Image);
//зафарбоўка графічнай вобласці
//белым колерам
gr.Clear (Color.White);
//колькасць пунктаў
n := 10000;
//канцы прамежку
x0 := StrToFloat (TextBox1.Text);
xn := StrToFloat (TextBox2.Text);
//цэнтр вобласці пабудовы
c_x := PictureBox1.Width div 2;
c_y := PictureBox1.Height div 2;
//маштабны каэфіцыент
k := PictureBox1.Width / (xn - x0);
//шаг
h := (xn - x0) / n;
x := x0;
//восі
p_c := new Pen (Color.Black, 1);
gr.DrawLine(p_c, 0, c_y, 2*c_x, c_y);
gr.DrawLine(p_c, c_x, 0, c_x, 2*c_y);
for var i := 1 to n do
begin
    y := x * sin(x);
    x1 := trunc(x * k) + c_x;
    y1 := trunc(-y * k) + c_y;
    //праверка пападання пункта
    //ў графічную вобласць
    if (y1 >= 0) and (y1 < 2*c_y) then
        bm.SetPixel(x1, y1, Color.Blue);
        x := x + h;
    end;
end;

```

3. Змяніць уласцівасці Text у кампанентаў Edit1 і Edit2 на -20 і 20 адпаведна.

4. Змяніць уласцівасці Text у кампанента Button1 на «Построить график».

5. Напісаць апрацоўшчык падзеі Click для кампанента Button1 і будаваць у ім графік функцыі па пунктах.

5.1. Намалюваць восі каардынат у выглядзе дзвюх перпендыкулярных ліній, якія перасякаюцца ў цэнтры кампанента PictureBox.

5.2. Каб атрымаць бачнасць сцэльнай лініі, колькасць пунктаў, якія ўтвараюць графік функцыі, павінна быць не меншай за 10 000 ($n = 10\,000$).

5.3. Шаг змянення значэння x вызначаецца як $h = \frac{x_n - x_0}{n}$.

5.4. Пры пабудове трэба ўлічваць маштаб: шырыня кампанента PictureBox павінна адпавядаць даўжыні задазенага прамежку. Тады маштабны каэфіцыент можна разлічыць па формуле

$$k = \frac{\text{PictureBox1.Width}}{x_n - x_0}.$$

5.5. Паколькі размяшчэнне восей каардынат на экране не супадае з размяшчэннем восей, прынятым у матэматыцы, то трэба ператварыць каардынаты: пункту (0; 0) павінен адпавядаць пункт у цэнтры кампанента PictureBox. Для гэтага атрыманае значэнне x трэба павялічыць на велічыню $c_x = \text{PictureBox1.Width} \text{ div } 2$, а значэнне y на $c_y = \text{PictureBox1.Height} \text{ div } 2$. Паколькі вось Y накіравана ўніз, а не ўверх, то ў значэння Y трэба яшчэ

памяняць знак на процілеглы. На канве будзе зафарбоўвацца пункт з каардынатамі $(x_{ekr} = x \cdot k + c_x, y_{ekr} = -y \cdot k + c_y)$.

5.6. Неабходна ўлічваць, што пры вылічэнні значэння x і y будуць рэчыўнымі, а значэнні графічных каардынат могуць быць толькі цэлымі. Таму перад прарысоўкай пункта трэба ператварыць рэчаісныя лікі ў цэлыя з дапамогай функцыі *trunc*.

5.7. Некаторыя пункты графіка пры пабудове могуць апынуцца за межамі графічнай вобласці, таму неабходна праверка значэння $y1$: значэнне павінна быць неадмоўным і меншым за вышыню графічнай вобласці.

4.3. Пабудова дыяграм

Асноўныя прынцыпы пабудовы гістаграм і кругавых дыяграм разбіраліся ў 10-м класе (прыклады 4.6 і 6.8). Выкарыстоўваючы аналагічныя метады Graphics, можна пабудаваць дыяграмы ў аконных дадатках, створаных у PascalABC.Net.

Прыклад 4.6. Стварыць праект і пабудаваць гістаграму па даных масіву з n элементаў ($n = 10$). Апісаць масіў з канстантнымі данымі ці дабавіць даныя ў масіў выпадковым чынам.

Этапы выканання задання

1. Змясціць на форму кампаненты: PictureBox і два кампаненты Button.

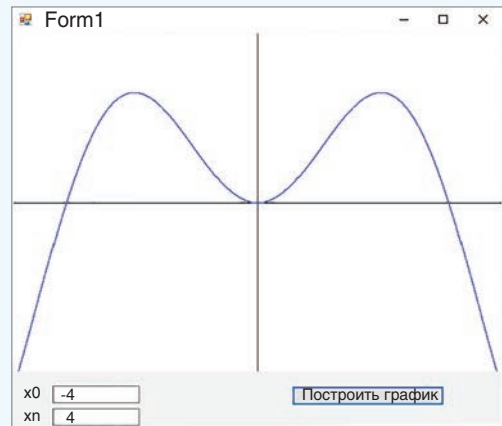
2. Змяніць уласцівасці Text у кампанента Button1 на «**Дыяграма с канстантнымі данымі**».

Прыклад 4.5. Працяг.

Дадатак у час работы:



Змяненне пачатковых значэнняў канцоў прамажку:



Прыклад 4.6. Форма на этапе канструявання:



Прыклад 4.6. Працяг.

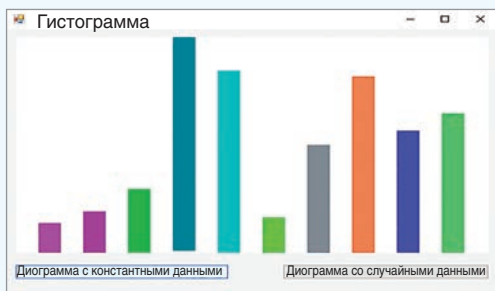
Апрацоўшчык падзеі Click для кампанента Button1:

```

procedure Form1.button1_Click
(sender: Object; e: EventArgs);
const a: array[1..10] of integer =
(10, 14, 22, 75, 63, 12, 37, 61, 42, 48);
n = 10;
var max, x, y1, y2, h, i, cr, cg,
cb: integer;
m: real;
gr: Graphics;
rnd: Random;
sb: SolidBrush;
begin
max := a[1];
for i := 2 to n do
if a[i] > max then
max := a[i];
h := trunc(PictureBox1.Width / (2*n+1));
m := PictureBox1.Height / max;
x := h;
//падрыхтоўка графічнай вобласці
для малявання прымітываў
gr := PictureBox1.CreateGraphics;
gr.Clear(Color.White);
rnd := new Random();
for i := 1 to n do
begin
cr := rnd.next(256);
cg := rnd.next(256);
cb := rnd.next(256);
sb := new SolidBrush(Color.FromArgb
(cr, cg, cb));
y1 := PictureBox1.Height - 1;
y2 := y1 - trunc(a[i] * m) - 1;
gr.FillRectangle(sb, x, y2, h, y1);
x := x + 2 * h;
end;
end;

```

Дадатак у час работы:



3. Змяніць уласцівасці Text у кампанента Button2 «Диаграмма со случайными данными».

4. Напісаць апрацоўшчык падзеі Click для кампанента Button1, у якім дыяграма будзе з дапамогай прамавугольнікаў.

4.1. Знайсці максімальны элемент у масіве *max*.

4.2. Разлічыць маштабны каэфіцыент:

$$m = \frac{PictureBox1.Height}{max}$$

4.3. У цыкле будаваць *n* прамавугольнікаў аднолькавай шырыні. Шырыня прамавугольніка

$$h = \frac{PictureBox1.Width}{2n + 1}$$

5. Апрацоўшчык для кампанента Button2 будзе адрознівацца ад апрацоўшчыка для кампанента Button1 толькі спосабам атрымання элементаў масіву.

5.1. Масіў павінен быць апісаны ў раздзеле **var: a: array[1..10] of integer;**

5.2. Элементы масіву са значэннямі ад 20 да 100 можна атрымаць наступным чынам:


```
rnd := new Random();
```

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

```
a[i] := rnd.next(80) + 20;
```

4.4. Анімацыя

Эфект анімацыі дасягаецца за кошт таго, што перад позіткам карыстальніка адбываецца хуткая змена відарысаў. Кожны з кадраў анімацыі застаецца на экране вельмі невялікі прамежак часу.

Для замеру інтэрвалаў часу можна выкарыстоўваць кампанент *таймер*. Ён змешчаны на панэлі **Компоненты** і паказаны ў выглядзе  Timer, імя аб'екта Timer. Кампанент Timer змяшчаецца ў асобную вобласць ніжэй за форму і атрымлівае імя TimerN, дзе N — нумар 1, 2, 3... (прыклад 4.7).

Некаторыя ўласцівасці кампанента Timer прыведзены ў табліцы:

| Уласцівасць | Прызначэнне |
|-------------|---|
| Enabled | Значэнне true абазначае, што таймер запушчаны |
| Interval | Час у мілісекундах, праз які адбываецца спрацоўванне таймера і выклік апрацоўшчыка Tick |

Кампанент мае адзіны апрацоўшчык — Tick, у якім апісваюцца дзеянні, што адбываюцца пасля заканчэння інтэрвалу спрацоўвання таймера.

Прыклад 4.8. Стварыць праект, у якім самалёт будзе пралятаць над горадам.

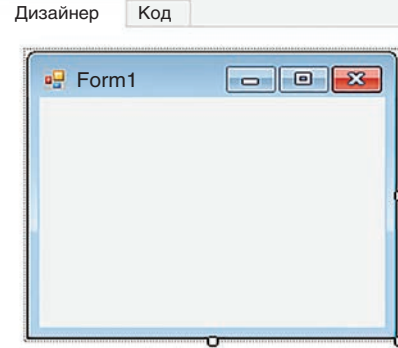
Этапы выканання задання

1. Змясціць на форму кампаненты PictureBox і Button, дабавіць кампанент Timer.

2. Загрузіць відарыс горада ў кампанент як фон формы (уласцівасць формы BackgroundImage).

3. Устанавіць празрысты колер фону для кампанента PictureBox1 (значэнне Transparent ва ўласцівасці BackColor). Устанавіць рэжым змянення памеру — AutoSize (уласцівасць SizeMode).

Прыклад 4.7. Кампанент *таймер*:



 timer1

Кампанент Timer не бачны пры рабоце дадатку, таму ён змяшчаецца ў вобласці, спецыяльна прызначанай для невізуальных кампанентаў.

Прыклад 4.8. Форма на этапе канстрування:



Апрацоўшчык падзеі Load для формы:

```

procedure Form1.Form1_Load (sender:
Object; e: EventArgs);
begin
    PictureBox1.Load ('plane.png');
    x := -PictureBox1.Width;
    y := 20;
    PictureBox1.Left := x;
    PictureBox1.Top := y;
end;

```

Прыклад 4.8. Працяг.

Апрацоўшчык падзеі Click для кампанента Button1:

```
procedure Form1.button1_Click
(sender: Object; e: EventArgs);
begin
  Timer1.Enabled := True;
end;
```

Апрацоўшчык падзеі Tick для кампанента Timer1:

```
procedure Form1.timer1_Tick
(sender: Object; e: EventArgs);
begin
  x := x + 1;
  PictureBox1.Left := x;
  if PictureBox1.Left > Width +
    + PictureBox1.Width then
    x := -PictureBox1.Width;
end;
```

Дадатак у час работы:



Калі пры запуску анімацыі ўзнікае мігаценне, то ўключыць двайную буферазацыю.

4. Напісаць апрацоўшчык падзеі Load для формы і апісаць пачатковае становішча самалёта, вызначыўшы каардынаты верхняга левага вугла PictureBox1 за межамі формы. Загрузіць у PictureBox1 відарыс з файла з малюнкам самалёта.

5. Змяніць уласцівасці Text у кампанента Button1 на «Паляцелі!».

6. Устаноўце значэнне False ва ўласцівасці таймера Enabled у інспектары аб'ектаў.

7. Устаноўце у інспектары аб'ектаў час спрацоўвання таймера роўным 10.

8. Напісаць апрацоўшчык падзеі Click для кампанента Button1, запусціць таймер.

9. У інспектары аб'ектаў устаноўце празрыстасць для кампанента PictureBox1.

10. Напісаць апрацоўшчык падзеі Tick і змяняць у ім значэнне ўласцівасці Left у кампанента PictureBox1. Калі самалёт выляцеў за мяжу, то вярнуць яго ў пачатковае становішча.

Двайная буферазацыя дазваляе зрабіць анімацыю больш плаўнай, паколькі ўсе аперацыі малявання спачатку выконваюцца ў памяці, а толькі потым на экране камп'ютара. Пасля завяршэння ўсіх аперацый малявання змесціва буфера капіруецца з памяці непасрэдна на звязаную з ім вобласць экрана.

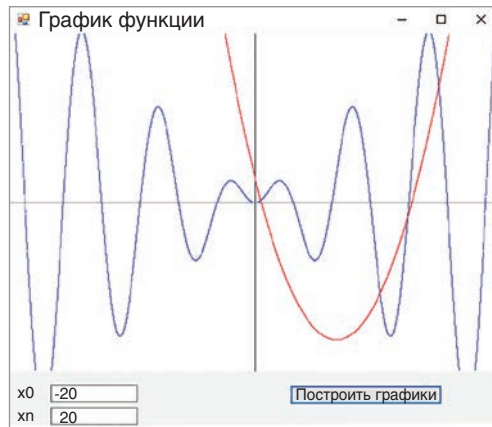


1. Які кампанент выкарыстоўваецца для размяшчэння відарысаў?
2. Якая ўласцівасць дазваляе загрузіць гатовы відарыс у кампанент PictureBox?
3. З дапамогай якой уласцівасці можна ўстанавіць празрысты фон для відарыса?
4. Які клас змяшчае метады для малявання графічных прымітываў?
5. Які метада канвы дазваляе зафарбаваць піксель?
6. Які кампанент выкарыстоўваецца для адліку часу?



Практыкаванні

- 1 Дабаўце ў праект з прыкладу 4.4. яшчэ адну жывёлу (напрыклад, вавёрку). Для змяшчэння вавёркі трэба дадавіць яшчэ адну кнопку. Месцазнаходжанне вызначыце выпадковым чынам у верхавіне якога-небудзь дрэва.
- 2 Дабаўце ў праект 4.5 пералічаныя магчымасці.
 1. Восі каардынат са стрэлкамі і подпісамі.
 2. Адзінкавы адрэзак на восі X .
 3. Графік функцыі $y = 0,3x^2 - 4x + 2$ у той жа сістэме каардынат чырвоным колерам.



Пабудуйце ў адной сістэме каардынат графікі наступных функцый. Кожную крывую малюйце сваім колерам. Для ўводу значэнняў параметраў a , b і c дабаўце на форму кампаненты Text:

1. $y = ax^2$, $y = b \cos x^2$;
 2. $y = ax^2 + bx + c$, $y = \frac{1}{x^2 + 3}$;
 3. $y = ax^3$, $y = \frac{x}{(x+3)^2}$, $y = bx \sin x$.
- 3 Змяніце праект з прыкладу 4.6 так, каб будавалася лінейная дыяграма (слупкі размешчаны гарызантальна).
 - 4* Змяніце праект з прыкладу 4.6 так, каб значэнні ў масіў можна было ўводзіць. Для гэтага лікі неабходна запісваць у кампанент TextBox, счытваць радок з лікаў і прабелаў, вылучаць лічбы і ператвараць іх у лікавыя значэнні.
 - 5 Дабаўце ў праект з прыкладу 4.8 кнопку «Стоп», пры націсканні на якую самалёт спыніцца. *Пасля спынення самалёта павінен з'явіцца парашут і апусціцца ўніз.
 - 6* Стварыце анімацыю руху Месяца вакол Зямлі. Для разліку каардынат верхняга левага вугла PictureBox1, які змяшчае Месяц, можна выкарыстаць параметрычнае ўраўненне акружнасці: $x = R \sin(t)$, $y = R \cos(t)$, дзе R — радыус, t — параметр, які змяняе сваё значэнне ад 0 да 2π .

§ 5. Стварэнне дадаткаў

Прыклад 5.1. Рэкамендацыі па стварэнні аконных дадаткаў.

1. У дадатку рэкамендуецца змясціць галоўнае меню і інструментальную панэль хуткіх кнопак, якія дубліруюць асноўныя раздзелы меню.

2. Пажадана, каб аб'екты дадатку валодалі кантэкставымі меню, якія з'яўляюцца пры націсканні правай клавшай мышы на аб'екце.

3. Для аб'ектаў рэкамендуецца прапісаць падказкі, якія ўсплываюць пры наведзенні паказальніка мышы на аб'ект.

4. Рэкамендуецца рэалізаваць радок стану, які выкарыстоўваецца для выдачы рознай інфармацыі.

5. Пры націсканні клавшы F1 павінен загрузацца файл даведкі.

6. У праграме пажадана рэалізаваць магчымасць настройкі і захавання настроек, каб пры наступным сеансе работы іх не давялося ўстанаўліваць нанова.

7. Калі вынік работы дадатку залежыць ад якіх-небудзь параметраў, абавязкова выберыце значэнні па змоўчанні. Яны дазваляць паскорыць узаемадзеянне карыстальніка з праграмай, а таксама з'яўляюцца прыкладам таго, у якім фармаце даныя трэба ўводзіць.

Магутным уплывам на псіхіку чалавека з'яўляецца колер, таму з ім трэба абыходзіцца вельмі асцярожна. Трэба імкнуцца выкарыстоўваць абмежаваны набор колераў і надаваць увагу іх правільнаму спалучэнню. Успрыманне колеру ў чалавека вельмі індывідуальнае, таму не трэба навязваць усім сваё бачанне колеру. Пажадана, каб асноўны колер формы быў нейтральным (напрыклад, у большасці дадаткаў Microsoft гэта светла-шэры колер).

5.1. Распрацоўка аконных дадаткаў

Стварэнне кожнага аконнага дадатку ажыццяўляецца, як правіла, у тры этапы:

1. Стварэнне інтэрфейса дадатку, г. зн. сродкаў узаемадзеяння карыстальніка з праграмай.

2. Распрацоўка сцэнарыя работы будучага дадатку. На гэтым этапе вызначаюць, якая інфармацыя будзе выводзіцца на экран, якія падзеі будуць адбывацца пры выкарыстанні розных кампанентаў, як дадатак павінен скончыць работу, якія вынікі і ў якім выглядзе захаваць і г. д.

3. Распрацоўка алгарытму рашэння пастаўленай задачы.

Большасць дадаткаў у аперацыйнай сістэме Windows выглядаюць і паводзяць сябе падобным чынам. Кампанія Microsoft прапанавала рэкамендацыі для распрацоўкі праграмнага забеспячэння¹, накіраваныя на тое, каб карыстальнік не марнаваў час на асваенне нюансаў карыстальніцкага інтэрфейса новай праграмы, а адразу пачаў прадукцыйна яе выкарыстоўваць. Гэтыя рэкамендацыі заснаваны на псіхафізіялагічных асаблівасцях чалавека і істотна палегчаць жыццё будучым карыстальнікам вашай праграмы.

Прывядзём некаторыя рэкамендацыі па распрацоўцы графічнага інтэрфейса аконных дадаткаў (прыклад 5.1).

¹ <https://docs.microsoft.com/ru-ru/windows/uwp/design/basics/design-and-ui-intro>

5.2. Стандартныя дыялогі

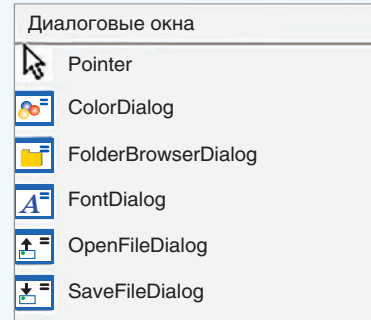
Практычна кожны дадатак Windows выкарыстоўвае стандартныя дыялогі, убудаваныя ў аперацыйную сістэму, для адкрыцця і захавання файлаў, выбару атрыбутаў шрыфту ці ўстаноўкі колеру, пошуку тэксту, друку. У бібліятэку VCL уключаны кампаненты, якія рэалізуюць адпаведныя вокны Windows. Яны змешчаны на панэлі **Дыялаговыя окна** (прыклад 5.2). У прыкладзе 5.3 прыведзены пералік кампанентаў для рэалізацыі стандартных дыялогаў.

Аб'екты на старонцы **Дыялаговыя окна** нябачныя ў час выканання, таму яны змяшчаюцца ў спецыяльнай вобласці пад формай (прыклад 5.4). Выгляд акна дыялогу залежыць ад версіі Windows.

Выклік і апрацоўка дыялогаў адбываецца праграма. Для ўсіх дыялогаў вызначаны метады ShowDialog() (прыклад 5.5). З дапамогай гэтага метаду адбываецца адкрыццё акна адпаведнага дыялогу. Ва ўласцівасцях кампанента-дыялогу запамінаецца выбар карыстальніка, які потым можна апрацаваць.

Дыялогі для адкрыцця і захавання файлаў выкарыстоўваюцца ў розных дадатках. Асноўная ўласцівасць кампанентаў OpenFileDialog і SaveDialog, у якой вяртаецца ў выглядзе радка імя файла, — гэта ўласцівасць FileName. Калі задаць дадзеную ўласцівасць на этапе канструявання ў акне інспектара аб'ектаў, то пры адкрыцці дыялогу

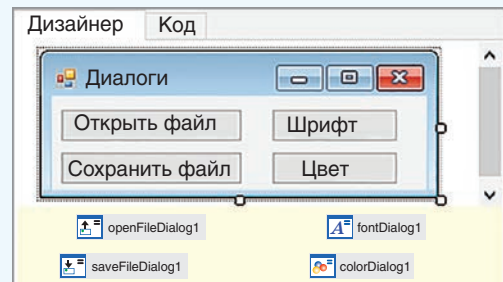
Прыклад 5.2. Панэль Дыялаговыя окна.



Прыклад 5.3. Спіс некаторых стандартных дыялогаў.

| Кампанент | Прызначэнне |
|----------------|--|
| OpenFileDialog | Стварэнне акна дыялогу « Открыть файл » |
| SaveFileDialog | Стварэнне акна дыялогу « Сохранить файл » |
| FontDialog | Стварэнне акна дыялогу « Шрифт » — выбар атрыбутаў шрыфту |
| ColorDialog | Стварэнне акна дыялогу « Цвет » — выбар колеру |

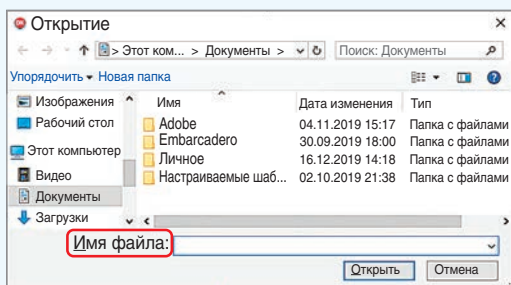
Прыклад 5.4. Дыялагавыя кампаненты і кнопкі для іх выкліку на форме:



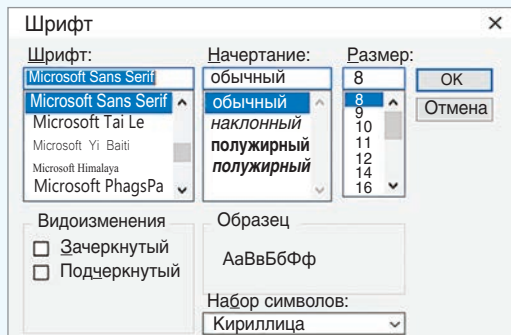
Прыклад 5.5. Стандартны зварот да дыялогу:

```
<імя дыялогу>. ShowDialog();
```

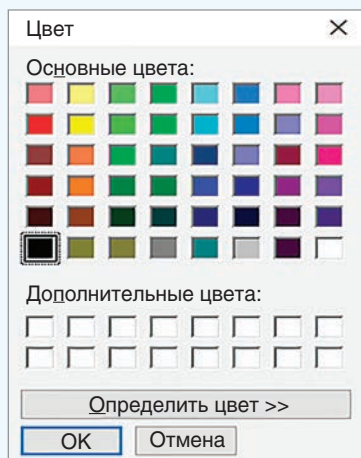
Прыклад 5.6. Стандартны дыялог для адкрыцця файла:



Прыклад 5.7. Стандартны дыялог для настроек шрыфту:



Прыклад 5.8. Стандартны дыялог для выбару колеру:



яно будзе з'яўляцца ў радку **Имя файла** (прыклад 5.6).

Для выкліку стандартнага акна ўстаноўкі атрыбутаў шрыфту можна выкарыстоўваць кампанент `FontDialog` (прыклад 5.7). У акне **Шрифт** карыстальнік можа выбраць імя шрыфту, яго стыль, памер. Асноўная ўласцівасць кампанента — `Font`.

Для выкліку стандартнага акна ўстаноўкі колеру выкарыстоўваецца кампанент `ColorDialog` (прыклад 5.8). У ім можна выбраць колер з базавай палітры. Асноўная ўласцівасць кампанента `ColorDialog` — `Color`. Гэта ўласцівасць адпавядае таму колеру, які карыстальнік выбраў у дыялогу.

5.3. Стварэнне меню

Практычна кожны дадатак павінен мець меню, якое дае зручны доступ да функцый праграмы. Існуе некалькі тыпаў меню:

- **галоўнае меню** с выпадаючымі спісамі раздзелаў;
- **каскадныя меню**, у якіх раздзелу першага меню належыць у адпаведнасць спіс падраздзелаў;
- **кантэкставыя меню**, якія з'яўляюцца пры націсканні правай клавішай мышы на аб'екце.

У `PascalABC.Net` меню ствараюцца кампанентамі `MenuStrip` (галоўнае меню) і `ContextMenuStrip` (кантэкставае меню), змешчанымі на панэлі **Меню и панели инструментов**. У час выканання праграмы самі кампаненты не бачны, таму змяшчаюцца ў спе-

цыяльнай вобласці пад формай (прыклад 5.9). На этапе выканання праграмы галоўнае меню будзе змешчана на сваё стандартнае месца — на версе формы, кантэкставае меню з’явіцца толькі пасля націскання правай кнопкі мышы па тым кампаненце, да якога яно належыць.

Для дабаўлення новых пунктаў меню трэба клікнуць левай клавішай мышы ў верхняй частцы формы (там, дзе звычайна змяшчаецца меню). Затым запоўніць ячэйкі, якія адпавядаюць пунктам меню (прыклад 5.10).

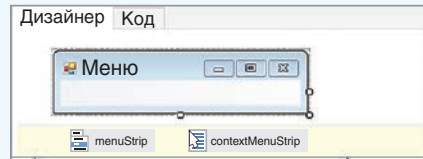
Кожны пункт меню з’яўляецца асобным аб’ектам. Спіс усіх кампанентаў, якія належаць да меню, можна ўбачыць у выпадаючым спісе ў інспектары аб’ектаў. Назвы пунктаў меню прапісваюцца ва ўласцівасці Text у акне інспектара аб’ектаў (прыклад 5.11).

Для кожнага пункта меню асноўнай падзеяй з’яўляецца падзея Click.

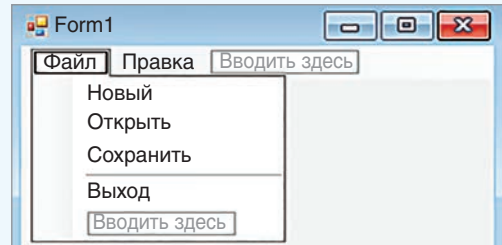
Стварэнне кантэкставых меню аналагічнае стварэнню галоўнага меню. Спачатку трэба выбраць кампанент на ніжняй панэлі, а затым запоўніць ячэйкі. Для таго каб пры пстрычцы правай кнопкай мышы на некаторым кампаненце з’яўлялася кантэкставае меню, трэба напісаць імя кантэкставага меню ва ўласцівасці ContextMenuStrip для выбранага кампанента (прыклад 5.12).

Напісанне апрацоўшчыкаў для меню і дыялогаў будзе разгледжана ў наступных пунктах.

Прыклад 5.9. Меню:

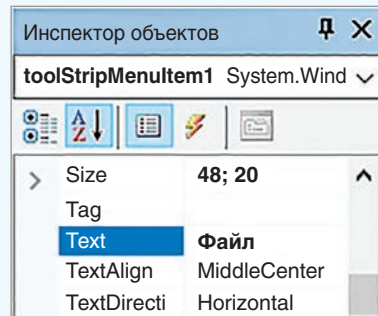


Прыклад 5.10. Рэдагаванне меню:

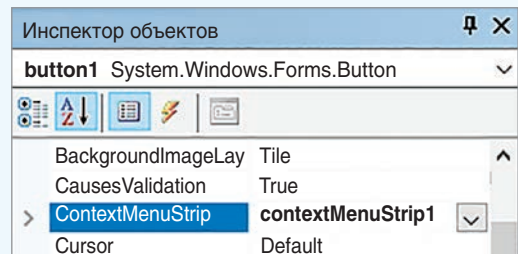


Калі ў якасці значэння ўласцівасці Caption увесці «—», то замест пункта меню з’явіцца раздзяляльнік.

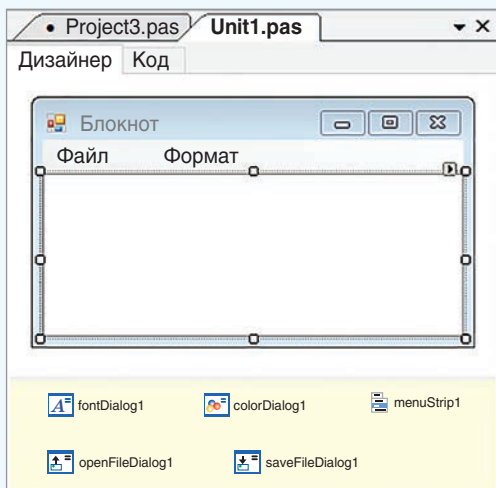
Прыклад 5.11. Назва пункта меню ў інспектары аб’ектаў:



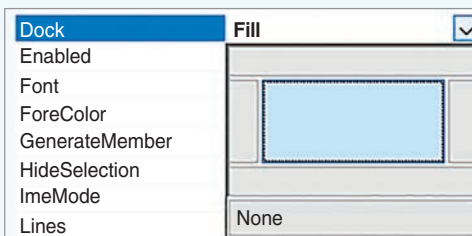
Прыклад 5.12. Кантэкставае меню для кампанента Button1:



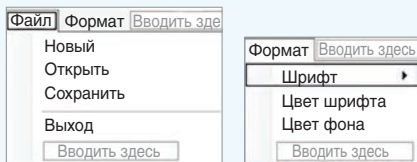
Прыклад 5.13. Форма на этапе канстрування:



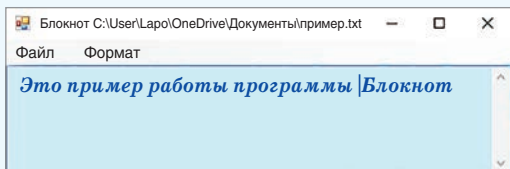
Прыклад 5.14. Настройка кліенцкай вобласці:



Прыклад 5.15. Структура меню:



Прыклад 5.16. Дадатак у час работы:



5.4. Стварэнне дадатку «Блакнот»

Праграма Блакнот павінна даваць магчымасць адкрыць і захаваць тэкставы файл, выбраць колер тэксту і колер фону.

Змясціць на форме (прыклад 5.13) наступныя кампаненты:

- рабочая вобласць для тэксту — `TextBox1`;
- дыялогі работы з файламі — `OpenFileDialog1`, `SaveFileDialog1`;
- дыялогі для настройкі знешняга выгляду дадатку — `FontDialog1`, `ColorDialog1`;
- галоўнае меню — `MenuStrip1`.

Кампанент `TextBox1` прызначаны для набору і рэдагавання тэксту. Тэкст можа набірацца ў некалькі радкоў, таму трэба ўстанавіць значэнне `true` для ўласцівасці `Multiline`. Для таго каб кампанент займаў усю кліенцкую частку формы, неабходна ўстанавіць ва ўласцівасці `Dock` значэнне `Fill` (прыклад 5.14). Устаноўце значэнне `Vertical` для ўласцівасці `ScrollBars` (вертыкальная паласа пракруткі).

Структура меню паказана ў прыкладзе 5.15. Для напісання апрацоўшчыкаў пунктаў меню трэба ў інспектары аб'ектаў выбраць адпаведны пункт меню, перайсці на ўкладку `Events` і выбраць падзею `Click`. Паколькі падзея `Click` з'яўляецца падзеяй па змоўчанні, то дваіны клік па пункце ў рэдактары меню створыць працедуру-апрацоўшчык.

Акно дадатку у час работы паказана ў прыкладзе 5.16.

Апрацоўшчыкі падзей для кожнага з пунктаў меню паказаны ў прыкладзе 5.17.

Для захавання і загрузкі файлаў апішам глабальную пераменную `F_N`:

```
var F_N: String;
```

Апрацоўшчык пункта меню **Новый** (`StripMenuItem4`) ачышчае радкі кампанента `TextBox1` ад уведзенага раней тэксту.

Апрацоўшчыкі пунктаў меню **Открыть** (`StripMenuItem5`) і **Сохранить** (`StripMenuItem6`) працуюць з файлам. Імя файла дабаўляецца да заглаўка акна.

Апрацоўшчык пункта меню **Выход** (`StripMenuItem8`) закрывае галоўную форму праекта.

Апрацоўшчык пункта меню **Шрифт** (`StripMenuItem9`) прыпісвае шрыфту, звязанаму з кампанентам `TextBox1`, уласцівасці, выбраныя карыстальнікам.

Апрацоўшчыкі пунктаў меню **Цвет текста** (`StripMenuItem10`) і **Цвет фона** (`StripMenuItem11`) устанаўліваюць для `TextBox1` колеры тэксту і фону, выбраныя карыстальнікам.

Для кампанента `TextBox` вызначаны наступныя дзеянні: Капіраваць (`Ctrl + C`), Выразаць (`Ctrl + X`), Устаўіць (`Ctrl + V`), Адмяніць (`Ctrl + Z`).

5.5. Стварэнне дадатку

«Графічны рэдактар»

Праграма «Графічны рэдактар» павінна даваць магчымасць адкрыць і захаваць файл, выбраць колер лініі і колер фону, устанавіць таўшчыню лініі. Рысаванне выконваецца выбраным колерам лініі пры націснутай левай клавiшы мышы. Клік правай клавiшай мышы ўнутры замкнутай вобласці выкарыстоўваецца для заліўкі абмежаванай вобласці выбраным колерам фону.

Прыклад 5.17. Апрацоўшчыкі падзей для пунктаў меню:

```
var F_N:string;

procedure
Form1.toolStripMenuItem4_Click
(sender: Object; e: EventArgs);
begin
//Файл – Новый
  TextBox1.Clear;
end;

procedure
Form1.toolStripMenuItem5_Click
(sender: Object; e: EventArgs);
begin
//Файл – Открыть
  openFileDialog1.ShowDialog();
  F_N := openFileDialog1.FileName;
  Text := 'Блокнот' + s;
  TextBox1.Lines := ReadAllLines(s);
end;

procedure
Form1.toolStripMenuItem6_Click
(sender: Object; e: EventArgs);
begin
//Файл – Сохранить
  saveFileDialog1.ShowDialog();
  F_N := saveFileDialog1.FileName;
  WriteAllLines(F_N,
    TextBox1.Lines);
  Text := 'Блокнот' + F_N;
end;

procedure
Form1.toolStripMenuItem8_Click
(sender: Object; e: EventArgs);
begin
//Файл – Выход
  close;
end;
```

Прыклад 5.17. Працяг.

```

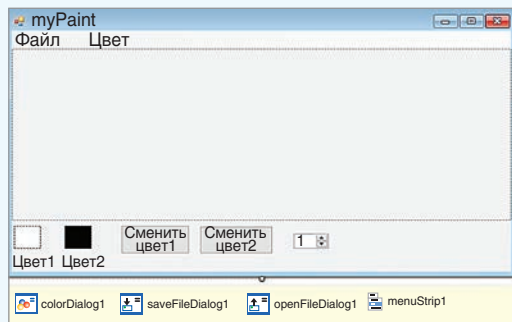
procedure Form1.
  toolStripMenuItem9_Click
  (sender: Object; e: EventArgs);
begin
  //Формат — Шрыфт
  fontDialog1.ShowDialog();
  TextBox1.Font := fontDialog1.
  Font;
end;

procedure Form1.
  toolStripMenuItem10_
  Click(sender: Object; e:
  EventArgs);
begin
  //Формат — Цвет текста
  colorDialog1.ShowDialog();
  TextBox1.ForeColor :=
  colorDialog1.Color;
end;

procedure Form1.
  toolStripMenuItem11_
  Click(sender: Object; e:
  EventArgs);
begin
  //Формат — Цвет фона
  colorDialog1.ShowDialog();
  TextBox1.BackColor :=
  colorDialog1.Color;
end;

```

Прыклад 5.18. Форма на этапе канструявання:



Спачатку спраектуем форму, змясціўшы на ёй наступныя кампаненты (прыклад 5.18):

- вобласць для малявання — PictureBox;
- кампаненты, якія адлюстроўваюць выбраны колер для малявання і колер фону — Panel1, Panel2;
- кнопкі для змены колеру;
- кампанент выбару колеру — ColorDialog1;
- кампанент для выбару таўшчыні лініі — numericUpDown1 (панэль кампанентаў **Стандартные элементы управления**);
- галоўнае меню — menuStrip1 і кампаненты для работы з файламі — OpenFileDialog1, SaveFileDialog1.

На этапе канструявання ўстанавіць значэнне ўласцівасці BackColor у кампанентаў Panel1 і Panel2 — Black і White адпаведна.

Ва ўласцівасцей Value і Minimum для кампанента numericUpDown1 устанавіць значэнне 1.

Структура меню паказана ў прыкладзе 5.19. Стварэнне малюнка і загрузка файла ў дадатак прыведзены ў прыкладзе 5.20.

У апрацоўшчыку падзеі Load для формы прапісаны першапачатковыя ўстаноўкі для стварэння графічнага аб'екта, зададзены параметры пэндзля і алоўка.

У апрацоўшчыку падзеі MouseDown для кампанента PaintBox1 задаём пераменнай m_d значэнне true — кнопка націснута. Тут жа запамінаем каардынаты пункта, паколькі ад гэтага пункта пачнём будаваць лінію.

У апрацоўшчыка `MouseDown` — значэнне пераменнай `m_d = false` — кнопка не націснута.

Для адсочвання траекторыі руху мышы па кампаненце `PaintBox1` ствараем апрацоўшчык падзеі `MouseMove`. Калі кнопка націснута, то можам будаваць лінію. Параметры `e.x`, `e.y` вяртаюць каардынаты пункта, у якім адбылося націсканне кнопкі.

Для перамяшчэння мышы неабходна выкарыстоўваць метада `DrawLine(x1, y1, e.x, e.y)` — рысаванне лініі, што злучае два пункты. Пасля прарысоўкі абнаўляем каардынаты.

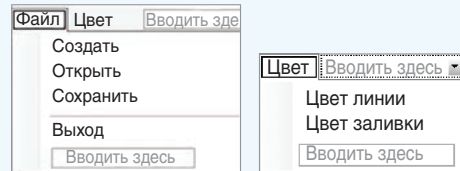
Таўшчыня лініі вызначаецца значэннем уласцівасці `Value` для кампанента `numericUpDown1`. Апрацоўшчык падзеі — `ValueChanged`.

Апрацоўшчыкі падзей для кампанентаў `OpenFileDialog1`, `SaveFileDialog1` выклікаюцца з адпаведных пунктаў меню і аналагічныя апрацоўшчыкам, апісаным для праграмы `Блакнот`. Для захавання і загрузкі файлаў неабходна апісаць глабальную радковую пераменную `FileName`. Дадатак можа захоўваць і загружаць файлы фармату `BMP`.

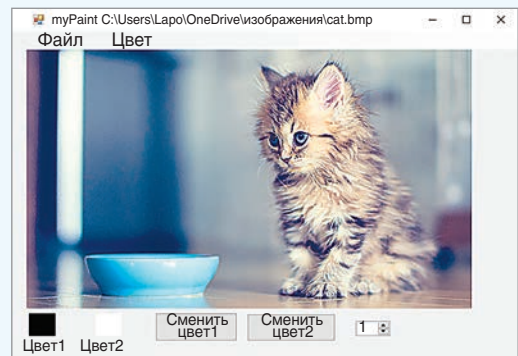
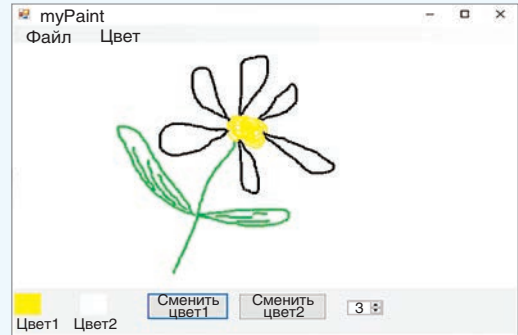
З дапамогай кампанента `ColorDialog1` можна выбраць колер лініі ці заліўкі. Пункты меню `Цвет` дазваляюць выбраць колер лініі ці заліўкі адпаведна.

У прыкладзе 5.21 прыведзена апісанне глабальных пераменных, якія выкарыстоўваюцца для стварэння дадатку «Графічны рэдактар». Апрацоўшчыкі ўсіх апісаных падзей прыведзены ў *Дадатку* (с. 105).

Прыклад 5.19. Структура меню:



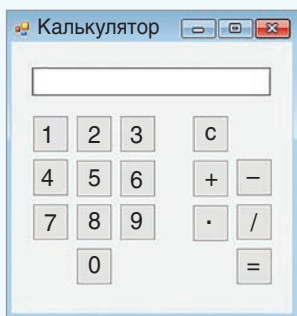
Прыклад 5.20. Дадатак у час работы:



Прыклад 5.21. Апісанне глабальных пераменных.

```
var gr: Graphics;
    bm: Bitmap;
    p_c: Pen;
    s_b: SolidBrush;
    c_f, c_b: Color;
    w: decimal;
    x1, y1, x2, y2: integer;
    m_d: boolean;
    F_N: string;
```

Прыклад 5.22. Форма на этапе канструявання:



Прыклад 5.23. Апрацоўшчыкі падзей:

```
//апісанне глабальных пераменных
var n1, n2: integer;
    znak: char;
procedure Form1.button1_Click
(sender: Object; e: EventArgs);
begin
    //прыпісванне лічбы да ліку
    TextBox1.Text := TextBox1.Text + '1';
end;
procedure Form1.button9_Click
(sender: Object; e: EventArgs);
begin
    //прыпісванне лічбы да ліку
    TextBox1.Text := TextBox1.Text1 + '2';
end;
```

Для іншых лічбавых кнопак неабходна змяніць толькі '1' на адпаведную лічбу.

```
procedure Form1.button12_Click
(sender: Object; e: EventArgs);
begin
    n1 := StrToInt(TextBox1.Text);
    //запамінанне знака аперцыі
    znak := '+';
    TextBox1.Clear;
end;
```

Для іншых кнопак са знакамі арыфметычных дзеянняў неабходна змяніць толькі '+' на адпаведны знак.

5.6. Стварэнне дадатку «Калькулятар»

Стварэнне калькулятара пачнём з канструявання формы. На ёй неабходна змясціць: поле TextBox для ўводу/вываду лікаў, 10 кнопак з лічбамі, 4 кнопкі з арыфметычнымі дзеяннямі, кнопку «=» і кнопку «C» — ачысціць (прыклад 5.22).

Пры націсканні на кнопку з лічбай праграма павінна дапісаць гэту лічбу да ліку ў полі TextBox. Пры націсканні на кнопку з арыфметычным дзеяннем неабходна запомніць лік, які ў дадзены момант знаходзіцца ў полі TextBox, і ачысціць поле для ўводу другога ліку. Лікі будзем захоўваць у дзвюх пераменных *n1*, *n2* тыпу *integer*. Знак аперцыі будзем захоўваць у пераменнай *znak* тыпу *char*. Пераменныя апісваюцца як глабальныя. Пры націсканні на кнопку «=» выконваецца арыфметычнае дзеянне і выводзіцца вынік.

Кнопкі могуць змяшчаць малюнак на паверхні (напрыклад, відарыса з лічбамі). Уласцівасць для змяшчэння малюнка — *BackgroundImage*.

Устанавіць значэнне *FixedSingle* для ўласцівасці *FormBorderStyle* формы. У гэтым выпадку мяжа формы не дазволіць змяняць яе памеры.

Коды працэдур-апрацоўшчыкаў прыведзены ў прыкладзе 5.23.

Для кожнай кнопкі на форме неабходна стварыць апрацоўшчык падзей *Click*.

Апрацоўшчыкі падзей для ўсіх лічбавых кнопак будуць ідэнтычныя.

Апрацоўшчыкі для кнопак арыфметычных дзеянняў будуць адрозні-

вацца толькі значэннем запамінальнай аперацыі.

Асноўныя вылічэнні адбываюцца ў апрацоўшчыка кнопкі «=». Ператварам у лік $n2$ значэнне поля Edit і выконваем арыфметычную аперацыю ў залежнасці ад значэння пераменнай *znak*. Пасля гэтага абнуляем пераменныя.

У апрацоўшчыка кнопкі «C» (ад англ. clear — ачысціць) адбываецца абнуленне пераменных і ачыстка поля Edit.

Створаны калькулятар мае вялікую колькасць абмежаванняў у сваёй рабоце, паколькі разлічаны на вылічэнні толькі з натуральнымі лікамі.

Прыклад 5.23. Працяг.

```
procedure Form1.button16_Click
(sender: Object; e: EventArgs);
begin
n2 := StrToInt (TextBox1.Text);
case znak of
'+': TextBox1.Text:= IntToStr (n1 + n2);
'-': TextBox1.Text:= IntToStr (n1 - n2);
'*': TextBox1.Text:= IntToStr (n1 * n2);
'/': TextBox1.Text:=IntToStr (n1 div n2);
end;
n1 := 0; n2 := 0;
znak := ' ';
end;
```

```
procedure Form1.button15_Click
(sender: Object; e: EventArgs);
begin
TextBox1.Clear;
n1 := 0; n2 := 0;
znak := ' ';
end;
```



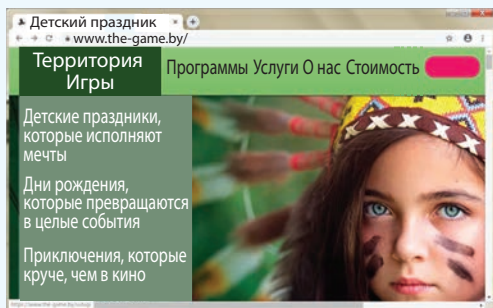
Практыкаванні

- 1 Дапоўніце праект Блакнот наступнымі магчымасцямі:
 1. Пункт меню **Сохранить файл** замяніце двума: **Сохранить** і **Сохранить как...**
 2. Дабаўце пункт меню **Переносить по словам**.
 3. Дабаўце магчымасць кіравання палосамі пракруткі: устанавіць гарызантальную, вертыкальную, абедзве.
 - 4*. Дабаўце наступныя дыялогі: **Параметры страницы, Печать, Поиск, Замена и соответствующие пункты в меню**.
 5. Дабаўце кантэкставае меню для кіравання кампанентам **Мето**. Каманды кантэкставага меню, якія дубліруюць каманды асноўнага меню, павінны выклікаць тыя ж апрацоўшчыкі.
 6. Прапануйце свае магчымасці.
- 2 Для праекта Графічны рэдактар дабаўце наступныя магчымасці:
 1. Рысаваць адрэзкі, авалы і прамавугольнікі.
 2. Прапануйце свае магчымасці.
- 3 Для праекта Калькулятар дабаўце наступныя магчымасці:
 1. Магчымасць работы з рэчаіснымі лікамі.
 2. Магчымасць вылічваць значэнні функцый: здабыванне квадратнага караня, трыганаметрычныя функцыі (для градусаў і радыян) і інш.
 - 3*. Магчымасць перакладу лікаў у іншыя сістэмы лічэння.

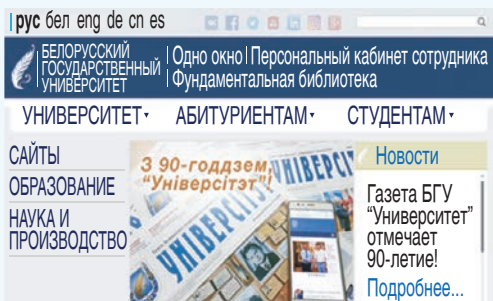
Глава 2 АСНОВЫ ВЭБ-КАНСТРУЯВАННЯ

§ 6. Вэб-канструяванне. Асноўныя паняцці

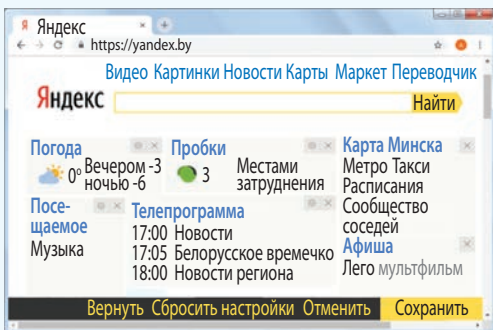
Прыклад 6.1. Сайт кампаніі, якая арганізуе дзіцячыя святы.



Прыклад 6.2. Сайт Беларускага дзяржаўнага ўніверсітэта.



Прыклад 6.3. Пошукавая сістэма Яндэкс.



6.1. Вэб-сайт

Вэб-сайт уяўляе сабой групу вэб-старонак, звязаных паміж сабой гіперспасылкамі і аб'яднаных агульнай тэмай і афармленнем.

У залежнасці ад зместу вэб-сайты могуць быць:

1. Прэзентацыйныя (сайты-візіткі) — рэклама пэўных акцый, мерапрыемстваў, падзей, прасоўванне пэўных тэхналогій, сэрвісаў ці паслуг (прыклад 6.1).

2. Карпарацыйныя — знаёмыя з кампаніяй, прадпрыемствам, арганізацыяй. Задача такіх сайтаў — стварэнне станоўчага іміджу, а таксама прасоўванне тавараў ці паслуг (прыклад 6.2).

3. Анлайн-сэрвісы — накіраваны на задавальненне штодзённых і побытавых патрэб звычайнага карыстальніка: пошук інфармацыі, работа з электроннай поштай і інш. (прыклад 6.3).

4. Электронныя магазіны — арыентаваны на атрыманне прыбытку ад продажу тавараў (прыклад 6.4).

Тыпы вэб-сайтаў у залежнасці ад тэхналогіі стварэння:

1. Статычныя — захоўваюцца на серверы і адлюстроўваюцца ў браўзеры ў адным і тым жа выглядзе (прыклад 6.5).

2. Дынамічныя — цалкам ці часткова генеруюцца на серверы ці ў браўзеры ў працэсе выканання запыту карыстальніка (прыклад 6.6).

Паводле ўзаемадзеяння карыстальніка з рэсурсамі вэб-старонак вэб-сайты можна падзяліць на:

1. Пасіўныя — інфармацыю на такіх сайтах можна толькі праглядаць (прыклад 6.5).

2. Інтэрактыўныя — карыстальнік мае магчымасць мяняць данымі з серверам, удзельнічаць у інтэрактыўным дыялогу (прыклад 6.6).

6.2. Мова гіпертэкставай разметкі дакумента HTML. Структура HTML-дакумента. Тэгі і атрыбуты. Гіперспасылкі

Вэб-старонка ўяўляе сабой гіпертэкставы дакумент, створаны на мове HTML (HyperText Markup Language — мова разметкі гіпертэксту).

Мова HTML апісвае структуру html-дакумента, дазваляе задаць базавыя параметры вэб-старонкі, уласцівасці і месцазнаходжанне аб'ектаў вэб-старонкі.

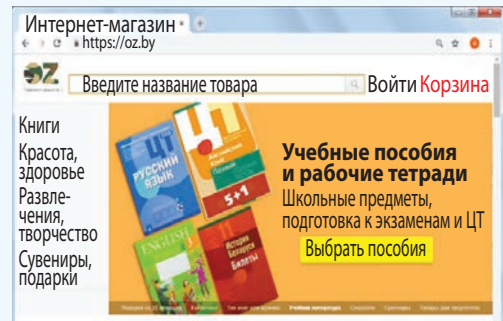
Html-дакумент — файл з расшырэннем **.html**. Такі файл можа быць адкрыты ў браўзеры як вэб-старонка.

Асноўныя кампаненты мовы HTML — тэгі, атрыбуты і іх значэнні.

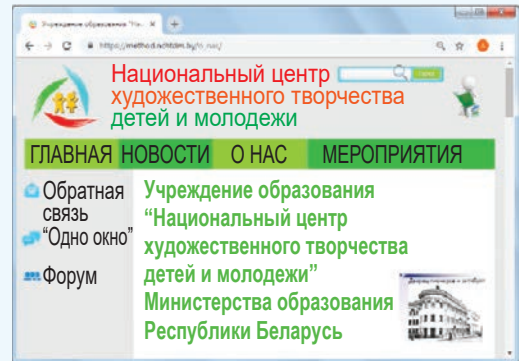
Тэгі — набор спецыяльных сімвалаў мовы HTML, якія ідэнтыфікуюць html-дакумент, вызначаюць раздзелы вэб-старонкі і становішча элементаў на вэб-старонцы.

Звычайна тэг — гэта кантэйнер з пары тэгаў (адкрываючага і закрываючага). Запісваюцца тэгі ў трохвугольных дужках: **<>**. Закрываючы тэг

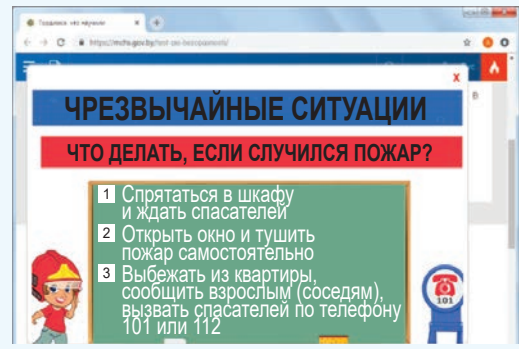
Прыклад 6.4. Сайт інтэрнэт-магазіна OZ.



Прыклад 6.5. Сайт Нацыянальнага цэнтра мастацкай творчасці дзяцей і моладзі.



Прыклад 6.6. Інтэрактыўная старонка на сайце МНС РБ з тэстам па бяспецы «Ганарымся, што навучылі».



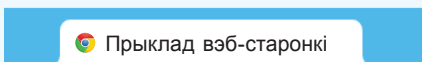
Мова HTML увесь час развіваецца. З 2014 г. рэкамендуецца выкарыстоўваць версію HTML5. Гэта версія змяшчае тэгі canvas, audio і video, якія дазваляюць дабаўляць графічныя, гукавыя і відэамаатэрыялы.

Прыклад 6.7. Структура HTML-дакумента.

```
<html>
<head> < ! -- Загаловак -->
  <title> Назва старонкі
</title>
  <meta charset = «utf-8»
  < ! -- устаноўка кодавай табліцы utf-8-->
</head>
<body> < ! -- цела -->
  <h1>Загаловак артыкула</h1>
  <p>Абзац артыкула</p>
</body>
</html>
```

Прыклад 6.8. Напісанне і адлюстраванне ў браўзеры тэга <title>.

```
<head>
  <title>Прыклад вэб-старонкі
</title>
</head>
```



Для павышэння якасці html-кода і забеспячэння супольнай работы над вэб-сайтамі прынята кіравацца пэўнымі правіламі афармлення і фармаціравання:

1. Выкарыстоўваць ніжні рэгістр, нягледзячы на тое што мова html неадчувальная да рэгістра.

2. Запісваць код так, каб радкі цалкам змяшчаліся на экране.

3. Вылучаць новы радок для кожнага адкрываючага тэга і ставіць водступы для кожнага даччынага (укладзенага) элемента.

4. Пры стварэнні вэб-старонак улічваць розніцу ў адлюстраванні html-кода ў розных браўзерах.

адрозніваецца ад адкрываючага толькі наяўнасцю сімвала /.

Код вэб-старонкі мае пэўную структуру:

1. Кантэйнер <html> </html> ідэнтыфікуе код як html-дакумент. У яго ўключаюцца кантэйнеры <head> </head> і <body> </body>.

2. Кантэйнер <head> змяшчае інфармацыю, якая не адлюстроўваецца на вэб-старонцы, — назву html-дакумента, тэгі для пошукавых машын і інш.

3. Кантэйнер <body> вызначае змест вэб-старонкі (кантэнт), які адлюстроўваецца браўзерам.

(Разгледзьце прыклад 6.7.)

Часта пры напісанні html-кода ўнікае неабходнасць устаўкі тлумачальных каментарыяў. Пачынаюцца яны з <!-- і заканчваюцца -->. Усё, што знаходзіцца ўнутры гэтага тэга, адлюстроўваецца на вэб-старонцы не будзе.

У загаложку html-дакумента змяшчаюцца:

1. Абавязковы кантэйнер <title> </title>, які вызначае назву вэб-старонкі і адлюстроўваецца на ўкладцы браўзера (прыклад 6.8).

2. Тэг <meta>, які ўстанаўлівае кадзіроўку дакумента.

Атрыбуты пашыраюць магчымасці тэгаў, задаюць уласцівасці аб'ектаў на вэб-старонцы. Напрыклад, можна выраўнаваць абзац ці задаць памер відарыса.





Значэнне атрыбута вызначае пэўныя ўласцівасці аб'ектаў на вэб-старонцы. Напрыклад, калі для тэга выкарыстоўваецца атрыбут выраўноўвання, то ён можа мець значэнне left або right.

Атрыбуты і іх значэнні запісваюцца ўнутры адкрываючага тэга (прыклад 6.9).

Асноўнай уласцівасцю гіпертэксту з'яўляецца здольнасць злучаць у адно цэлае інфармацыю, пададзеную ў лічбавай форме (тэкст, аўдыя- і відэафайлы, графіку, анімацыю, html-дакументы). Галоўным пры гэтым становіцца ўказанне на тое, што з чым і якім чынам павінна быць звязана. На вэб-старонках такія сувязі рэалізуюцца з дапамогай **гіперспасылак**. Гіперспасылка складаецца з дзвюх частак: паказальніка спасылкі і адраснай часткі спасылкі. Паказальнік спасылкі — гэта тое, што мы бачым на вэб-старонцы (тэкст ці малюнак). Тэкставы паказальнік звычайна вылучаецца сінім колерам і падкрэсліваннем.

Прыклад 6.9. Сінтаксіс мовы HTML.

```
<body bgcolor = "gray" text = "white">
Змесціва
</body>
```

-  — пара тэгаў;
-  — атрыбуты;
-  — значэнні атрыбутаў;
-  — месціва тэга-кантэйнера.

Пры афармленні гіперспасылак рэкамендуецца прытрымлівацца правілаў:

1. Спасылка павінна быць лёгка пазнавальнай.
2. Спасылка не павінна парушаць цэласнасць тэксту. Лішнія спасылкі замяняюць чытанню.
3. Па тэкставым паказальніку спасылкі павінна быць зразумела, куды яна вядзе.
- Тэкст пад спасылкай не павінен быць доўгім.



1. Па якіх прынцыпах можна падзяліць сайты на групы?
2. На якія групы можна падзяліць сайты ў залежнасці ад іх зместу?
3. Якая мова выкарыстоўваецца пры стварэнні вэб-старонак?
4. Якую структуру мае html-дакумент?
5. Што такое тэг, атрыбут?
6. Для чаго служаць гіперспасылкі?



Практыкаванні

- 1 Адкрыце ў браўзеры пералічаныя сайты.

1. edu.gov.by
2. camelotmebel.by
3. careers.epam.by
4. calc.by

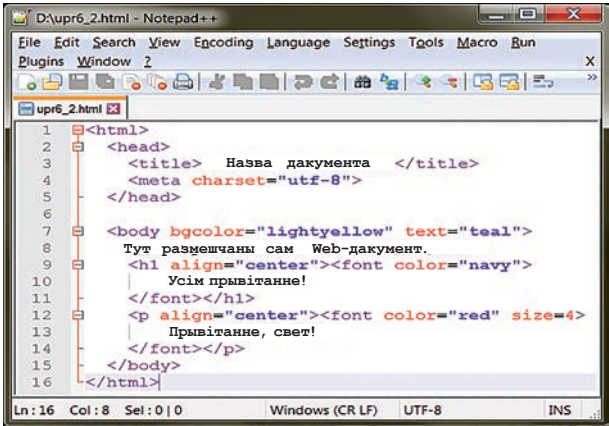
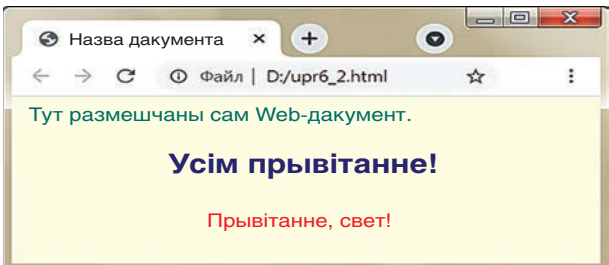
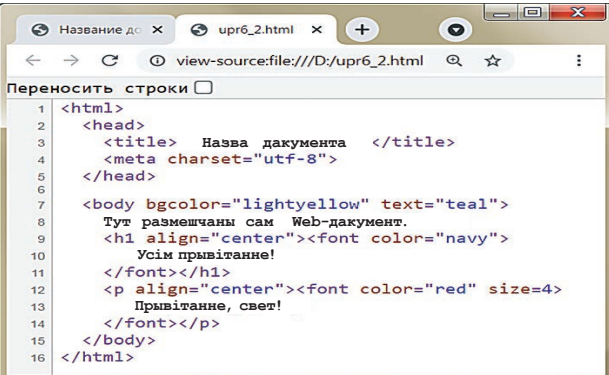
Да якога тыпу можна залічыць кожны з сайтаў у залежнасці ад іх зместу?

- 2 Адкрыце старонкі вэб-сайтаў, выкарыстоўваючы спасылкі:

1. <http://testy.by/quiz>
2. <https://stellarium-web.org>

Да якога тыпу можна залічыць дадзеныя сайты паводле арганізацыі ўзаемадзеяння карыстальніка з рэсурсамі вэб-старонак гэтых вэб-сайтаў?

3 Выканайце пералічаныя дзеянні. Якім колерам пры праглядзе кода старонкі ў браўзеры вылучаюцца тэгі, атрыбуты, значэнні атрыбутаў, змесціва тэгаў?

| Дзеянне | Вынік |
|--|--|
| <p>Адкрыцьце файл у рэдактары кода (напрыклад, Notepad++)</p> |  |
| <p>Захавайце гэты файл як html-дакумент і адкрыцьце яго ў браўзеры</p> |  |
| <p>У кантэкставым меню адкрытай у браўзеры вэб-старонкі выберыце Просмотр кода страницы ці выкарыстайце камбінацыю кlawіш Ctrl + U</p> |  |

4 Выпішыце ў сшытак тэгі, якія вызначаюць структуру html-дакумента з практыкавання 3.

§ 7. Стварэнне вэб-старонак

7.1. Інструменты стварэння вэб-старонак

Для стварэння вэб-старонак могуць выкарыстоўвацца розныя сродкі, ад самых простых да вельмі складаных. Выбар інструментаў залежыць ад таго, для каго і з якой мэтай ствараецца сайт.

У якасці інструментаў для стварэння вэб-старонак можна выкарыстоўваць:

1. Тэкставыя рэдактары (напрыклад, **Блакнот**). Стварэнне вэб-старонак у гэтым выпадку ўяўляе сабой працаёмкі працэс.




2. Рэдактары з падсветкай html-кода. Такія рэдактары звычайна маюць інтэлектуальны рэжым уводу (падказка пры ўводзе), што аптымізуе і спрашчае працу (прыклад 7.1).

3. Візуальныя вэб-рэдактары. З дапамогай такіх рэдактараў можна ствараць вэб-старонку, не ведаючы мовы HTML. Змест у працэсе рэдагавання выглядае максімальна падобным да канчатковай прадукцыі (прыклад 7.2).

4. Канструктары сайтаў — інтэрнэт-сэрвісы, якія даюць магчымасць хутка стварыць сайт з сучасным дызайнам без ведання мовы HTML і без выкарыстання спецыяльнага праграмачнага забеспячэння (прыклад 7.3).

Пасля напісання кода html-дакумента ў якім-небудзь рэдактары яго патрабуецца захаваць як файл з расшырэннем **.html**.

Прыклад 7.1. Рэдактары html-кода.

| | |
|---|--|
|  Notepad++ | Бясплатны рэдактар з адкрытым кодам для Windows |
|  Atom | Бясплатны кросплатформавы (Windows, MacOS, Linux) рэдактар |
|  Sublime Text | Кросплатформавы рэдактар. Мае бясплатную і платную версію |

Прыклад 7.2. Візуальныя вэб-рэдактары.

| | |
|--|---|
|  Web Page Maker | Бясплатная пробная версія. Прапануюцца разнастайныя шаблоны элементаў навігацыі |
|  Web Builder | Бясплатная пробная версія. Падтрымка шмат якіх фарматаў відэа |
|  Adobe Dreamviewer | Бясплатная пробная версія. Складана выкарыстоўваць без ведання HTML |

Прыклад 7.3. Канструктары сайтаў.

wix

Бясплатныя і платныя тарыфы. Канструктар увесь час абнаўляецца і прапануе новыя магчымасці.

ukit GROUP

Ёсць бясплатныя пробны перыяд. Шмат стыльных шаблонаў.

Прыклад 7.4. Элементы вэб-старонкі.
Варыянты размяшчэння:



Вёрсткай вэб-старонак называецца стварэнне такога html-кода, які дазваляе змяшчаць элементы вэб-старонкі ў патрэбных месцах дакумента і адлюстроўваць іх у акне браўзера згодна з распрацаваным макетам. Макет распрацоўваецца вэб-дызайнерам у графічным рэдактары.

Нярэдка пры стварэнні вэб-старонкі ўзнікае неабходнасць у выкарыстанні сімвалаў, якіх няма ў стандартнай камп'ютарнай клавіятуры або якія не падтрымлівае кадзіроўка html-дакумента. Такія сімвалы называюцца спецыяльнымі сімваламі HTML.

Каб змясціць спецыяльныя сімвалы на вэб-старонцы, неабходна задаць іх html-коды (см. *Дадатак да главы 2*, с. 107).

7.2. Элементы афармлення вэб-старонак

Любая вэб-старонка змяшчае пэўны набор стандартных элементаў. Безумоўна, гэты набор можа вар'іравацца ў залежнасці ад тэматычнай накіраванасці, а таксама ад мэта і задач сайта. Стварэнне такіх элементаў і праектаванне іх узаемнага размяшчэння з'яўляецца адной з галоўных задач вэб-распрацоўкі.

Асноўныя элементы афармлення вэб-старонак:

1. Загалавак (шапка сайта, header). Знаходзіцца ў верхняй частцы кожнай старонкі сайта. Можа быць выкананы як у тэкставым, так і ў графічным варыянце (часта гэта **лагатып**).

2. Асноўная частка. Займае большую частку вэб-старонкі. Тут змяшчаецца змястоўны інфармацыйны тэкст і ілюстрацыі — кантэнт (ад англ. *content* — змест).

3. Элементы навігацыі (меню). Асноўнае меню звычайна змяшчаецца пад шапкай, а дапаможнае — у левай частцы старонкі. Уяўляе сабой сукупнасць гіперспасылкаў.

4. Ніжні калантытул (падвал, footer). Знаходзіцца на кожнай старонцы сайта. Звычайна тут змяшчаецца інфармацыя пра распрацоўшчыка, кантактная інфармацыя, іконкі сацыяльных сетак і г. д.

5. Бакавыя панэлі. Вертыкальныя палосы справа і/ці злева ад асноўнай часткі. У іх размяшчаюцца, напрыклад, спасылкі, рэкламныя блокі, форма галасавання і г. д.

(Разгледзьце прыклад 7.4.)

7.3. Тэкст на вэб-старонцы

Як правіла, аснову кантэнту вэб-старонак складае тэкставая інфармацыя. Таму вельмі важна навучыцца ўводзіць і фармаціраваць блокі тэксту на старонцы. Для работы з тэкстам у мове HTML ёсць вялікая колькасць тэгаў.

Звычайна тэкст падзяляюць на абзацы. Гэта палягчае чытанне вялікага тэксту. У мове HTML для стварэння абзаца выкарыстоўваецца кантэйнер `<p>`. Пры праглядзе html-старонкі ў браўзеры абзацы аддзяляюцца невялікімі інтэрваламі (прыклад 7.5).

Для выраўноўвання тэксту ў абзацы тэг `<p>` падтрымлівае атрыбут `align` (прыклад 7.6). Ён можа прымаць адно з чатырох значэнняў:

- `left` — выраўноўванне тэксту па левым краі (па змоўчанні);
- `center` — выраўноўванне тэксту па цэнтры;
- `right` — выраўноўванне тэксту па правым краі;
- `justify` — выраўноўванне тэксту па шырыні.

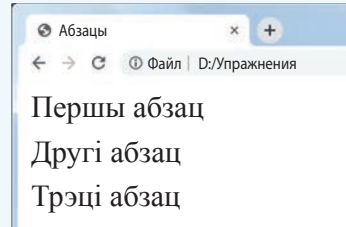
Для вылучэння раздзелаў тэксту на вэб-старонцы ўжываюцца загаловкі. У мове HTML існуе шэсць узроўняў загатоўкаў. Самым верхнім узроўнем з'яўляецца ўзровень 1 — тэг `<h1>`, а самым ніжнім — узровень 6, тэг `<h6>`. Змесціва загатоўка першага ўзроўню адлюстроўваецца самым буйным шрыфтам тлустага напісання, а загатоўка апошняга, шостага ўзроўню — самым дробным (прыклад 7.7). У тэгах `<h1>...<h6>` аналагічна тэгу `<p>`

Прыклад 7.5. Выкарыстанне тэга `<p>`.

Html-код:

```
<p>Першы абзац.</p>
<p>Другі абзац.</p>
<p>Трэці абзац.</p>
```

Адлюстраванне ў браўзеры:

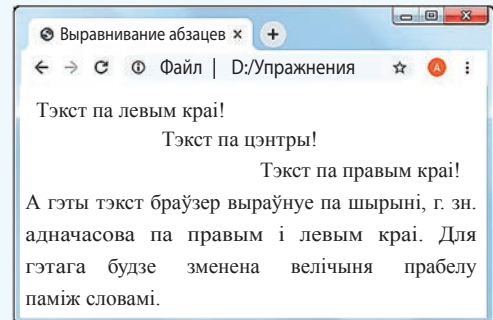


Прыклад 7.6. Выраўноўванне абзацаў у html-дакуменце.

Html-код:

```
<p align = "left">Тэкст па левым краі!
</p>
<p align = "center">Тэкст па цэнтры!</p>
<p align = "right">Тэкст па правым краі!</p>
<p align="justify">А гэты тэкст браўзер
выраўнуе па шырыні, г.зн. адначасова
па правым і левым краі. Для гэтага будзе
зменена велічыня прабелу паміж
словамі.</p>
```

Адлюстраванне ў браўзеры:

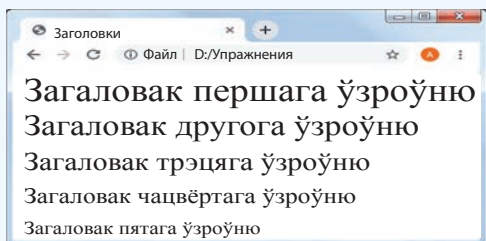


Прыклад 7.7. Загаловкі ў тэксце html-дакумента.

Html-код:

```
<h1>Загаловак першага ўзроўню</h1>
<h2>Загаловак другога ўзроўню</h2>
<h3>Загаловак трэцяга ўзроўню</h3>
<h4>Загаловак чацвёртага ўзроўню</h4>
<h5>Загаловак пятага ўзроўню</h5>
```

Адлюстраванне ў браўзеры:

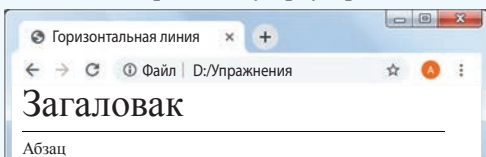


Прыклад 7.8. Гарызантальныя лініі ў html-дакуменце.

Html-код:

```
<h1>Загаловак</h1>
<hr>
<p>Абзац</p>
```

Адлюстраванне ў браўзеры:



Прыклад 7.9. Спосабы задання колеру ў html-дакуменце.

1. Па назве колеру:

```
<hr color = "red">
```

2. Шаснаццацярычным лікавым кодам:

```
<hr color = "#ff0000">
```

Запомніць лікавыя коды колераў складана. Для вызначэння кода патрэбнага колеру можна выкарыстоўваць табліцы (гл. *Дадатак да главы 2*, с. 107) ці інтэрнэт-сэрвісы (напрыклад, <https://colorscheme.ru/>).

выкарыстоўваюць атрыбут `align` для выраўноўвання тэксту.

Часам узнікае неабходнасць уставіць у тэкст html-дакумента перанос радка, не ствараючы пры гэтым абзаца. Напрыклад, пры разметцы вершаў ці тэкстаў песень. Для гэтага прадугледжаны няпарны тэг `
`, які паказвае на месца пераносу тэксту.

Для падзелу ці дадатковага вылучэння блокаў тэксту ў мове HTML існуе няпарны тэг `<hr>` — гарызантальная лінія (прыклад 7.8). Гарызантальная лінія расцягваецца на ўсю шырыню вэб-старонкі і мае адзін-два пікселі ў таўшчыню ў залежнасці ад браўзера, які выкарыстоўваецца.

Тэг `<hr>` падтрымлівае наступныя атрыбуты:

- `align` — вызначае выраўноўванне лініі;
- `size` — устанаўлівае таўшчыню лініі ў пікселях;
- `width` — вызначае шырыню лініі ў пікселях ці ў працэнтах у адносінах да шырыні акна браўзера;
- `color` — задае колер лініі.

Задаваць колер у html-дакуменце можна двума спосабамі: з дапамогай назвы (англійскія словы для назвы колеру) ці з дапамогай лікавага шаснаццацярычнага кода (прыклад 7.9). Перад кодам ставіцца знак `#`. У адпаведнасці з колеравай мадэллю RGB дзве першыя лічбы кода задаюць інтэнсіўнасць чырвонага (red) колеру, трэцяя і чацвёртая — зялёнага (green), дзве апошнія — сіняга (blue). Выкарыстанне кантэйнера `<div>` дз-

валяе згрупаваць розныя элементы вэб-старонкі ў блок.

У прыкладзе 7.10 у блок вылучана некалькі абзацаў. Атрыбут `align="center"` вызначае выраўноўванне па цэнтры для ўсіх абзацаў, аб'яднаных у блок.

7.4. Гіперспасылкі на вэб-старонцы

Гіперспасылкі ў html-дакуменце ствараюцца з дапамогай кантэйнера `<a>`. Унутры яго могуць быць: слова, група слоў ці відарыс.

Тэг `<a>` падтрымлівае атрыбуты:

- `href` — абавязковы атрыбут, які паказвае на абсалютны або адносны адрас спасылкі ці на імя закладкі для ўнутранай спасылкі;

- `target` — вызначае, дзе будзе адкрыты дакумент, на які ідзе спасылка (напрыклад, `_blank` — у новай укладцы ці акне).

Кантэйнер `<a>` можа быць выкарыстаны двума спосабамі:

1. Для спасылкі на іншы дакумент — лакальная спасылка (прыклад 7.11).

2. Для спасылкі на элемент унутры дакумента — унутраная спасылка.

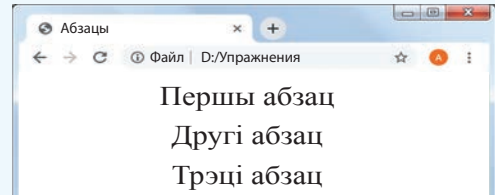
Вялікія дакументы чытаюцца лепш, калі яны маюць унутраныя спасылкі (прыклад 7.12). Значэннем атрыбута `href` у дадзеным выпадку будзе так званы якар — спасылка на ідэнтыфікатар аб'екта (`id`). `Id` — унікальная прымета аб'екта, якая дазваляе адрозніваць яго ад іншых аб'ектаў, г. зн. ідэнтыфікаваць. Імя для ідэнтыфікатара лепш выбіраць у адпаведнасці з прызначэннем спасылкі.

Прыклад 7.10. Блокі ў html-дакуменце.

Html-код:

```
<div align = "center">
  <p>Першы абзац</p>
  <p>Другі абзац</p>
  <p>Трэці абзац</p>
</div>
```

Адлюстраванне ў браўзеры:



Прыклад 7.11. Лакальныя спасылкі на вэб-старонку.

1. У адным каталозе:

```
<a href = "index.html">Домой</a>
```

2. Ва ўкладзеным каталозе:

```
<a href = "sait/index.html">Дадому</a>
```

3. У бацькоўскім каталозе:

```
<a href = "../index.html">Дадому</a>
```

4. У суседнім каталозе:

```
<a href = "../sait/index.html">Дадому</a>
```

Прыклад 7.12. Стварэнне ўнутранай спасылкі.

Створым спасылку ў канцы старонкі на яе пачатак.

Спачатку створым якар. Першы абзац, на які будзе ажыццёўлены пераход, атрымлівае ідэнтыфікатар `top`:

```
<p id = "top"> ... </p>
```

Імя спасылкі ў дадзеным выпадку пачынаецца з сімвала `#`:

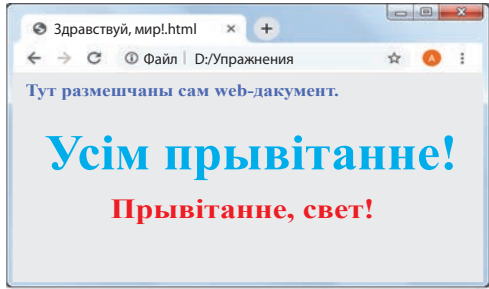
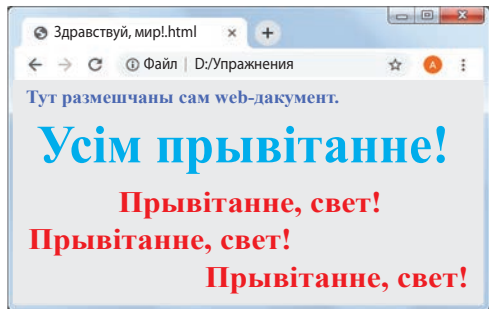
```
<a href = "#top">Наверх</a>
```

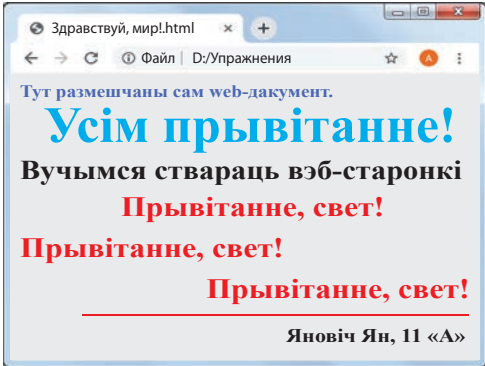

1. З дапамогай якіх прыкладных праграм можна ствараць вэб-старонкі?
2. Якое расшырэньне мае файл, што змяшчае код вэб-старонкі?
3. Які элемент вэб-старонкі займае яе большую частку?
4. Што такое шапка сайта? Што такое падвал сайта?
5. Які тэг выкарыстоўваецца для стварэння абзацаў? Як адлюстроўваюцца абзацы ў браўзеры?
6. Якія тэгі выкарыстоўваюцца для стварэння загатоўкаў? Які з гэтых тэгаў вызначае заглавак з самым буйным шрыфтам?
7. Чым, акрамя памеру, адрозніваецца адлюстраванне ў браўзеры тэксту загатоўкаў і тэксту абзацаў?
8. Якія атрыбуты падтрымліваюць тэгі абзацаў і загатоўкаў?
9. Як перайсці на новы радок унутры абзаца вэб-старонкі?
10. З дапамогай якога тэга можна размясціць на вэб-старонцы гарызантальную лінію? Якія атрыбуты падтрымлівае гэты тэг?
11. Які тэг вызначае гіперспасылку?
12. Якія спасылкі выкарыстоўваюцца ў html-дакуменце?



Практыкаванні

- 1 Адкрыце html-дакумент, створаны ў практыкаванні 2 пасля папярэдняга параграфа. Выканайце пералічаныя дзеянні.

| Дзеянне | Вынік |
|---|--|
| <p>Змяніце колер фону і асноўнага тэксту (значэнні атрыбутаў <code>b bgcolor</code> і <code>text</code> тэга <code>body</code>). Для задання новага колеру выкарыстоўвайце шаснаццацярычныя лікавыя коды (<code>#f5f5f5</code> — фон, <code>#4169e0</code> — тэкст)</p> |  |
| <p>Размясціце на вэб-старонцы некалькі абзацаў з розным выраўнаннем тэксту</p> |  |

| Дзеянне | Вынік |
|--|--|
| <p>Дабаўце на старонку заглавак другога ўзроўню і гарызантальную лінію. Колер загатоўка выберыце самастойна. Пад гарызантальнай лініяй (у ніжнім калантытуле) змясціце абзац з персанальнымі данымі (прозвішча, імя, клас)</p> |  |

- 2 Адкрыце файл index.html. Аформіце абзацы гэтай вэб-старонкі як спасылкі.
 - 1-ы абзац — спасылка на рэсурс <https://colorscheme.ru/>.
 - 2-і абзац — спасылка на вэб-старонку 2_2.html.
- 3 Дабаўце на старонку 2_2.html тэкст «Вярнуцца назад» і аформіце яго як перакрываваную спасылку на старонку index.html.
- 4 Стварыце на старонцы 2_2.html унутраную спасылку. Для гэтага:
 1. Стварыце закладку на пачатку старонкі.
 2. Дабаўце ў канцы старонкі тэг `
` і тэкст «Наверх» со спасылкай на закладку.

§ 8. Паняцце пра каскадныя табліцы стыляў

У адпаведнасці з канцэпцыяй сучаснай вэб-распрацоўкі html-код павінен змяшчаць толькі тэгі і кантэнт у іх. Для апісання знешняга выгляду сайта выкарыстоўваюцца стылявыя правілы спецыяльнай мовы разметкі стыляў — CSS.

CSS (англ. Cascading Style Sheets — каскадныя табліцы стыляў) — фармальная мова апісання знешняга выгляду html-дакумента.

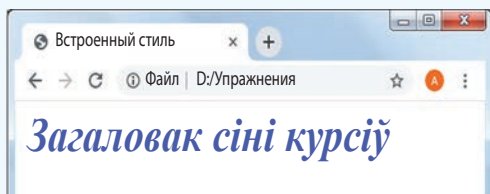


Хокан Віум Лі (нарадз. 27 ліпеня 1965 г., Нарвегія) — вучоны, спецыяліст у галіне інфарматыкі, у 1994 г. прапанаваў выкарыстоўваць каскадныя табліцы стыляў (CSS).

Прыклад 8.1. Убудаваныя стылі.
Стылявое апісанне:

```
<h1 style = "color: blue; font-style: italic">Загалавак сіні курсіў</h1>
```

Адлюстраванне ў браўзеры:



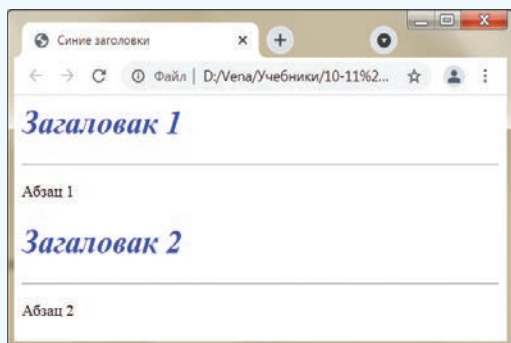
Прыклад 8.2. Табліцы стыляў.
Стылявое апісанне:

```
<style>
  h1 {color: blue;
      font-style: italic}
</style>
```

Разметка вэб-старонкі:

```
<body>
  <h1>Загалавак 1</h1>
  <hr>
  <p>Абзац 1</p>
  <h1>Загалавак 2</h1>
  <hr>
  <p>Абзац 2</p>
</body>
```

Адлюстраванне ў браўзеры:



CSS можа быць выкарыстаны як у дачыненні да асобнага тэга (элемента вэб-старонкі), так і адначасова да ўсіх ідэнтычных элементаў на ўсіх старонках сайта. CSS дапаўняе мову HTML і значна пашырае яе магчымасці.

Аддзяляючы стыль уяўлення дакументаў ад змесціва, CSS спрашчае стварэнне вэб-старонак. Наладка сайта становіцца больш хуткай і зручнай. З'яўляецца магчымым адначасовай работы з адным праектам адразу некалькім распрацоўшчыкам: дызайнеру, вярстальшчыку, праграмісту. За кошт гэтага павышаецца і хуткасць распрацоўкі сайта.

Спосабы падключэння стыляў CSS да html-дакумента:

1. Убудаваныя стылі — стылявое апісанне непасрэдна ў адкрываючым тэгу (прыклад 8.1). Дзейнічае толькі для гэтага тэга.

2. Табліцы стыляў — стылявое апісанне для ўсіх ідэнтычных элементаў вэб-старонкі (прыклад 8.2).

Задаецца з дапамогай парнага тэга `<style>`, які павінен знаходзіцца ў загалоўку дакумента. Стылі html-элементаў унутры тэга `<style>` задаюцца ў адпаведнасці з пэўным сінтаксісам.

3. Знешнія табліцы стыляў — стылявое апісанне html-элементаў у асобным файле (прыклад 8.3).

Знешнія табліцы стыляў — гэта файлы з расшырэннем `.css`, якія змяшчаюць стылявыя правілы, могуць быць створаны ў любым рэдактары

кода. Для падключэння знешніх табліц стыляў у загаловак html-дакумента змяшчаецца тэг `<link>` з абавязковымі атрыбутамі:

- `href` — спасылка на css-файл (напрыклад, `href = "my_styles.css"`);
- `rel = stylesheet` — адказвае за ўстаноўку ўзаемасувязі html-дакумента і css-файла;
- `type = text/css` — апісвае тып даных у табліцы стыляў.

Паняцце каскадных CSS заключаецца ў тым, што стылі, падключаныя да html-дакумента рознымі спосабамі, маюць розныя прыкрытэты.

Наяўнасць прыкрытэтаў азначае, што спачатку будуць выкарыстаны параметры фармаціравання, зададзеныя ў знешнім css-файле, а затым — прапісаныя ў загатоўку дакумента. У апошнюю чаргу фармаціраванне будзе дапоўнена або зменена ў адпаведнасці з параметрамі, абвешчанымі непасрэдна ў тэгах.

На вэб-старонцы пры дапамозе CSS можна змяніць:

- выгляд, памер і колер тэкставых элементаў;
- фонавыя колеры і відарысы;
- пазіцыю асобных элементаў у адносінах да іншых;
- водступы ўнутры элементаў і паміж імі;
- наяўнасць і знешні выгляд абводкі элементаў;
- бачнасць або нябачнасць пэўных фрагментаў на старонцы.

Прыклад 8.2. Працяг.

Сінтаксіс падключэння стыляў да html-элементаў у табліцах стыляў:

```
p {color: #ff0000; font-size: 14px}
■ — селектар;
■ — уласцівасць CSS;
■ — значэнні ўласцівасці CSS.
```

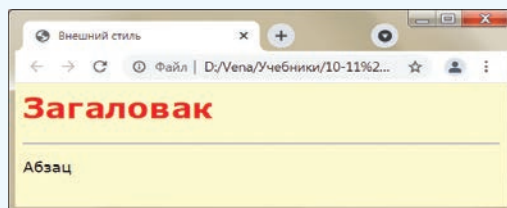
Прыклад 8.3. Знешнія табліцы стыляў. Апісанне стыляў у css-файле:

```
h1
{color: red}
/*Заголовок чырвонага колеру*/
body
{background: #ffffcc; font-family:
Verdana}
/*колер фону #ffffcc, шрыфт Verdana*/
```

Падключэнне css-файла да html-дакумента:

```
<head>
<link rel = stylesheet type = text/css
href = "1.css">
</head>
```

Адлюстраванне ў браўзеры:



Пры выкарыстанні CSS варта кіравацца наступнымі правіламі:

1. Выкарыстоўваць атрыбут `style` для якога-небудзь элемента, толькі калі стыль гэтага элемента адрозніваецца ад стыляў іншых аналагічных элементаў сайта.
2. Тэг `<style>` са стылявым апісаннем выкарыстоўваць у тым выпадку, калі старонка павінна мець індывідуальны дызайн, адрозны ад іншых старонак сайта.
3. У большасці выпадкаў разумна выносіць стылявыя апісанні ў асобны css-файл.

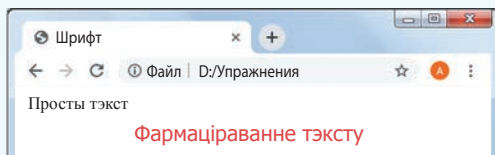
Прыклад 8.4. Выкарыстанне стыляў фармацавання шрыфту.

```
p {
    font-family: Tahoma;
    /* Тып шрыфту */
    font-weight: bold;
    /* Тлустасьць шрыфту */
    font-size: 18pt;
    /* Памер шрыфту */
    color: tomato;
    /* Колер тэксту */
    text-align: center;
    /* Выраўноўванне тэксту */
}
```

Скарочаны запіс:

```
p {
    font: bold 18pt Tahoma;
    color:tomato; text-align:center;
}
```

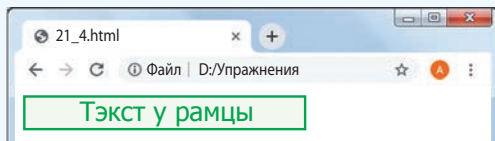
Адлюстраванне ў браўзеры:



Прыклад 8.5. Выкарыстанне стыляў для блока.

```
div {
    width:80%;
    /*Шырыня блока */
    background: #f0fff0;
    /*Колер фону*/
    border: 3px solid limegreen;
    /*Таўшчыня, стыль і колер мяжы */
    padding: 10px; /*Палі*/
    font: 18pt Tahoma;
    /*Памер і тып шрыфту */
    color: seagreen;
    /*Колер тэксту */
}
```

Адлюстраванне ў браўзеры:



З дапамогай CSS можна вызначыць стыль і від тэксту аналагічна выкарыстанню тэга ``, які задае ўласцівасці шрыфту. Пры гэтым стылі валодаюць вялікімі магчымасцямі і дазваляюць скараціць html-код. З атрыбутамі CSS, зададзенымі ўласцівасцямі шрыфту, можна пазнаёміцца ў *Дадатку да главы 2* (с. 110).

Стылі шрыфту задаюцца вялікім пералікам асобных уласцівасцей. Запіс, які вызначае ўласцівасці шрыфту, можна скараціць. У скарочаным запісе трэба быць асцярожным з парадкам указання уласцівасцей. Напрыклад, уласцівасць `size` павінна задавацца раней, чым уласцівасць `family` (прыклад 8.4).

Акрамя змянення параметраў шрыфтоў, можна кіраваць і ўласцівасцямі ўсяго тэксту. Значэнні ўласцівасцей прыведзены ў *Дадатку да главы 2* (с. 110).

З дапамогай CSS задаюцца ўласцівасці блокаў (прыклад 8.5).

Унутраныя водступы ў блоку (водступы ад знешняй мяжы блока да яго зместу) задае ўласцівасць `padding`. Гэтыя водступы часам называюць палямі.

Існуе некалькі спосабаў задання палёў:

1) `padding: 10 px` — аднолькавыя водступы з усіх бакоў;

2) `padding: 5 px 10 px` — водступы зверху і знізу `5px`, справа і злева `10px`;

3) `padding: 5 px 10 px 15 px` — водступы зверху `5px`, злева і справа `10px`, знізу `15px`;

4) `padding: 5 px 10 px 15 px 20 px` — розныя водступы з усіх бакоў, злева направа: верхні, правы, ніжні, левы.

Знешнія водступы блока (водступы ад знешняй мяжы блока да меж старонкі або да суседніх элементаў) задае ўласцівасць `margin`. Спосабы задання знешніх водступаў аналагічныя спосабам задання ўнутраных (прыклад 8.6). Калі неабходна вызначыць стыль для асобнага элемента вэб-старонкі або задаць розныя стылі для аднаго тэга, ужываюць класы (прыклад 8.7).

Сінтаксіс пры выкарыстанні класаў:

```
.Імя_класа {
  уласцівасць1: значэнне;
  уласцівасць2:
  значэнне; ... }
```

Каб паказаць у `html`-кодзе, што тэг выкарыстоўваецца з пэўным класам, да тэга дабаўляецца атрыбут `class="Імя_класа"`.

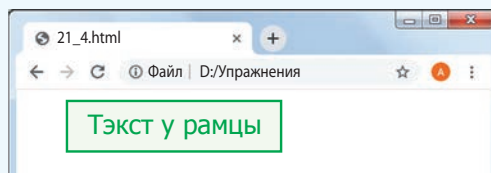
Класы гібкія, іх можна ствараць шмат і называць зразумелымі імёнамі. Імя класа можа змяшчаць у сабе літары, лікі, сімвал дэфіса (-) і падкрэслівання (_). Пачынацца імя павінна з лацінскай літары.

Створаны клас можна ўжываць у дачыненні да любых элементаў вэб-старонкі. З дапамогай класаў можна надаваць стыль не толькі цэлым загаловам і абзацам, але і асобным фрагментам старонкі, напрыклад словам. Для гэтага трэба заключыць слова ў кантэйнер `` і дадавіць атрыбут `class`.

Прыклад 8.6. Водступы ў блоку.

```
padding: 20px 10px;
margin: 10px 80px;
```

Адлюстраванне ў браўзеры:



Прыклад 8.7. Выкарыстанне класаў у CSS.

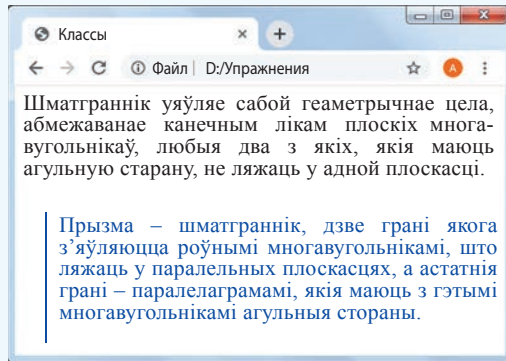
Стылявое апісанне:


```
p { /* Звычайны абзац */
  text-align: justify;
  /*Выраўноўванне тэксту па шырыні*/
}
.фигура { /* Абзац з класам figura */
  color: navy;
  /*Колер тэксту*/
  margin-left: 20px;
  /*Водступ злева*/
  border-left: 1px solid navy;
  /*Мяжа злева ад тэксту*/
  padding-left: 15px; /*Адлегласць ад лініі да тэксту*/
}
```

Указанне на выкарыстанне класа:

```
<p class = "figura">
```

Адлюстраванне ў браўзеры:



-  1. Што такое каскадныя табліцы стыляў?
2. Якімі спосабамі можна падключыць каскадныя табліцы стыляў да html-дакумента?
3. Якім чынам вызначаюцца прыярытэты пры падключэнні каскадных табліц стыляў рознымі спосабамі да аднаго html-дакумента?
4. Што можна змяняць на вэб-старонцы з дапамогай каскадных табліц стыляў?



Практыкаванні

1 Адкрыце файл 8.html. Аформіце абзацы гэтай вэб-старонкі, выкарыстоўваючы ўбудаваныя стылі.

- 1-ы абзац — `style = 'color: #c71585; font-style: italic'` (колер тэксту — лілова-чырвоны, напісанне — курсіў).
- 2-і абзац — `style = 'color: #c71585; font-size: 16px; text-align: justify'` (колер тэксту — лілова-чырвоны, памер шрыфту — 16 px, выраўноўванне тэксту — па шырыні).
- 3-і абзац — `style = 'color: #dc143c; font-family: Arial'` (колер тэксту — малінавы, шрыфт — Arial).
- 4-ы абзац — `style = 'color: teal; font-family: Verdana; font-size: 16px'` (колер тэксту — зеленавата-сіні, шрыфт — Verdana, памер шрыфту — 16 px).

Захавайце старонку пад новым імем.

2 Адкрыце файл 8.html. Аформіце вэб-старонку, выкарыстоўваючы табліцы стыляў.

```
<style>
p {color: teal; font-style: italic; text-align: justify}
h1 {font-family: Verdana; text-align: center}
</style>
```

Захавайце старонку пад новым імем.

3 Адкрыце файл 8.html. Аформіце вэб-старонку, выкарыстоўваючы знешнія табліцы стыляў (апісанне стыляў у файле). Параметры стыляў задайце самастойна. Захавайце старонку пад новым імем.

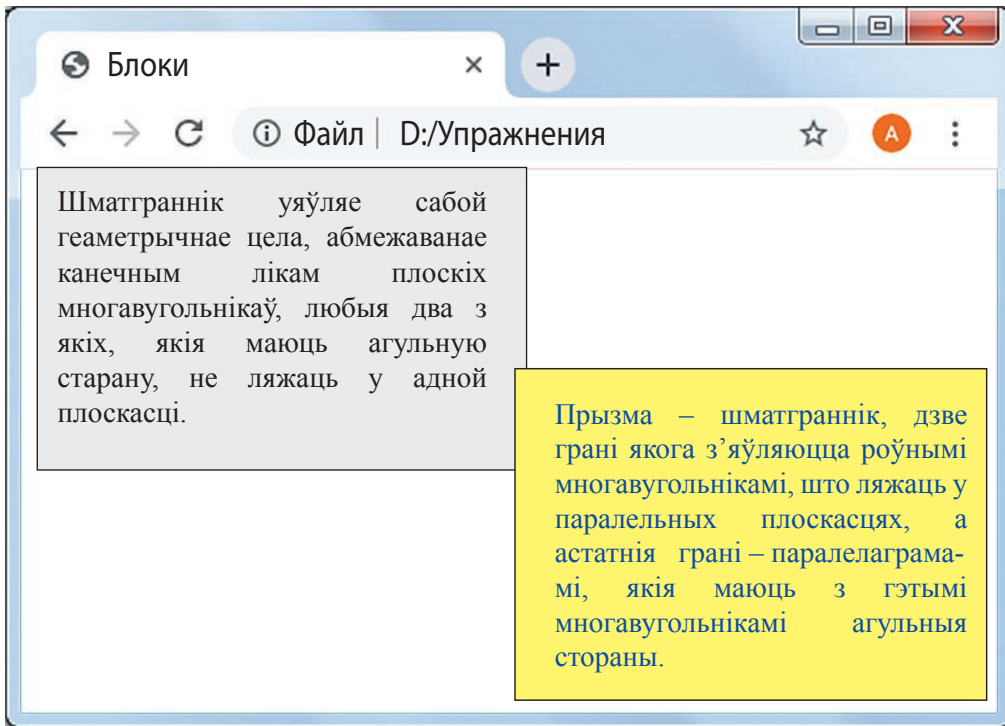
4 Адкрыце файл. Выкарыстайце ў дачыненні да блокаў тэксту рознае фармаціраванне, ужываючы класы.

```
1-ы блок:
div.b1 {
width: 300px; /* Шырыня */
background: #d3d3d3; /* Колер фону */
padding: 8px; /* Палі */
padding-right: 40px; /* Правае поле */
border: solid 1px black; /* Стыль, таўшчыня і колер мяжы */
float: left; /*Выраўноўванне блока па левым краі*/
}
```

2-і блок:

```
div.b2 {
width: 300px;
background: #f0e68c;
padding: 8px;
border: solid 2px black;
float: left;
position: relative; /* Размяшчэнне блока адносна першага блока */
top: 40px; /* Зрушэнне па вертыкалі ўніз */
left: -70px; /* Зрушэнне па гарызанталі ўлева */
}
```

Павінна атрымацца:



Выкарыстоўвайце клас для фармаціравання слова *Прызма* ў другім блоку.

Апісанне класа:

```
.termin {font-family: Verdana}
```

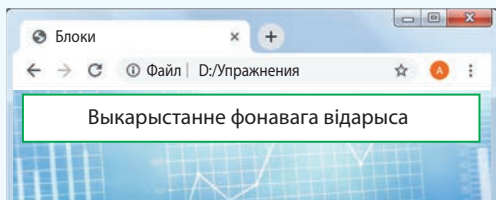
Выкарыстанне класа:

```
<span class = "termin">Прызма</span>
```

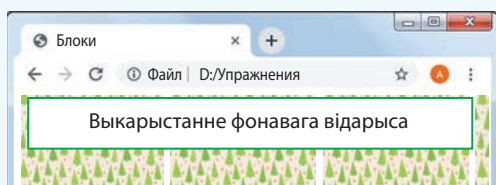

§ 9. Мультымедыя на вэб-старонках

Прыклад 9.1. Выкарыстанне фонавага відарыса.

Памер відарыса 1920 × 1200 рх:



Памер відарыса 225 × 225 рх:



Прыклад 9.2. Разметка фонавага відарыса.

Файл з відарысам знаходзіцца ў адной папцы з html-файлам:

```
<body background = "fon.jpg">
```

Файл з відарысам знаходзіцца ў папцы з імем img:

```
<body background="img/fon.jpg">
```

Прыклад 9.3. Дабаўленне ўласціваці width да відарыса.

```
img {
  width: 300px;
  /* Шырыня ў пікселях */
}
```

Пры выкарыстанні працэнтнага запісу шырыня відарыса вызначаецца ў залежнасці ад шырыні бацькоўскага элемента (напрыклад, блока). Калі бацькоўскі элемент не вызначаны, то ў яго якасці выступае акно браўзера.

```
img {
  width: 80%;
  /* Шырыня ў працэнтах */
}
```

9.1. Графіка на вэб-старонках

Выкарыстанне відарысаў на вэб-старонках дазваляе наглядна данесці да аўдыторыі патрэбную інфармацыю. Графіка дае магчымасць крыху ажывіць агульнае афармленне вэб-старонкі. Акрамя таго, для афармлення вэб-старонак ужываецца дзевяць графіка, г. зн. розныя графікі, дыяграмы і схемы.

На вэб-старонках відарысы актыўна выкарыстоўваюцца не толькі ў якасці ілюстрацый да тэксту, але і для фонавага афармлення (прыклады 9.1).

Калі фонавы відарыс меншы за памер вэб-старонкі, ён будзе размножаны па гарызанталі і вертыкалі.

Пры выкарыстанні відарыса ў якасці фону неабходна памятаць, што ён не павінен адцягваць увагу ад тэксту і павінен добра спалучацца з колеравай гамай вэб-старонкі.

Файлы з відарысамі, якія выкарыстоўваюцца на старонках сайта, прынята захоўваць у асобнай папцы.

Каб змясціць відарыс на вэб-старонку як фон, трэба ўстанавіць у тэгу <body> атрыбут background са значэннем, якое змяшчае поўнае імя файла з відарысам (прыклад 9.2).

Для дабаўлення ілюстрацыі на старонку выкарыстоўваецца няпарны тэг крыніцы відарыса:

```
,
дзе src – абавязковы атрыбут тэга img.
```

Тэг `` прызначаны для адлюстравання на вэб-старонцы відарысаў графічных фарматаў GIF, JPEG, PNG або SVG.

Асноўныя ўласцівасці CSS, якія выкарыстоўваюцца ў дачыненні да тэга ``:

- `width` — шырыня відарыса;
- `height` — вышыня відарыса;
- `padding` — водступы ад відарыса;
- `float` — выраўноўванне відарыса;
- `border` — рамка вакол відарыса.

Уласцівасці `width` і `height` выкарыстоўваюцца для змянення памераў відарыса (прыклад 9.3). Значэнні гэтых уласцівасцей можна задаваць у пікселях ці працэнтах. Даданне толькі адной з гэтых уласцівасцей дазволіць захаваць прапорцыі відарыса.

Па змоўчанні тэкст на вэб-старонцы змяшчаецца шчыльна да відарыса, што парушае эстэтычнае ўспрыманне кантэнту. Гэтага можна пазбегнуць, задаўшы для відарыса нябачныя водступы па гарызанталі (злева і справа) і вертыкалі (зверху і знізу). Водступы для відарысаў задаюцца гэтак жа, як для блокаў, — значэннямі ў пікселях для ўласцівасці `padding` (прыклад 9.4).

Дабавіць на старонку відарысы так, каб яны цалкам абцякаліся тэкстам, можна з дапамогай стылявой уласцівасці `float`. Значэнне `right` будзе выраўноўваць відарыс па правым краі акна браўзера ці блока, а `left` — па левым краі. Абцяканне пры гэтым адбываецца па іншых, свабодных ба-

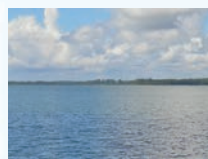
Прыклад 9.4. Водступы для відарыса¹.

```
img {
  padding: 10px;
  /* Аднолькавыя водступы з усіх бакоў*/
}
```



Прыклад 9.5. Выраўноўванне відарыса ў адносінах да тэксту.

```
img {
  float: left;
  /* Відарыс злева ад тэксту */
  padding: 5px;
  /* Водступ */
}
```



Нарач – самае вялікае возера ў Беларусі. Плошча яго люстэрка – 79,6 кв. км. Найбольшая глыбіня – 24,8 км. Даўжыня берагавой лініі – 40 км (чацвёртае

месца сярод азёр Беларусі). Аб'ём вады – 710 млн кв. м.

```
img {
  float: right; /* Відарыс справа ад тэксту */
}
```

Нарач – самае вялікае возера ў Беларусі. Плошча яго люстэрка — 79,6 кв. км. Найбольшая глыбіня – 24,8 км. Даўжыня берагавой лініі – 40 км

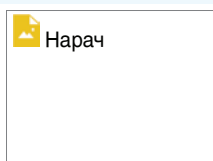


(чацвёртае месца сярод азёр Беларусі). Аб'ём вады – 710 млн кв. м.

¹ Тэкст у прыкладзе цытуецца па: <https://gp.by/category/news/culture/news21091.html> (дата доступу: 28.08.2019).

Прыклад 9.6. Альтэрнатыўны тэкст.

```
<img src = "ozero.jpg" alt = "Нарач">
```



Максім Танк

Люблю твае, Нарач, затокі і тоні,
Як вецер густыя туманы развесіць.
Ці снежная пена на вхвалях зазвоніць,
Цалуючы зоры, калышучы месяц.

Прыклад 9.7. Выкарыстанне відарыса ў якасці спасылкі.

Стылявое апісанне:

```
img {border: none}
```

Html-код:

```
<a href = "sample.html">  
<img src = "link.png" alt = "Прыклад">  
</a>
```

Атрыбутаў, якія могуць быць выкарыстаны ў тэгу , і ўласцівасцей, што выкарыстоўваюцца пры стылявым апісанні, вялікая колькасць. З імі можна пазнаёміцца на рэсурсе <http://htmlbook.ru/content>.

Пры ўключэнні мультымедыйных элементаў на старонку неабходна добра ўяўляць, ці будуць апраўданы затраты часу, якія ідуць на загрузку. Мультымедыйныя файлы займаюць шмат памяці. Самыя вялікія з іх — відэа-файлы, мінутны фрагмент можа займаць больш за 10 Мбайт памяці. Гукавыя файлы патрабуюць крыху менш месца — троххвілінны запіс высокай якасці можа займаць да 3,5 Мбайт.

Для дабаўлення гуку на вэб-старонкі лепш выкарыстоўваць файлы з расшырэннем .mp3. Даныя ў такіх файлах упакаваны з улікам асаблівасцей успрымання чалавечым вухам.

ках. У прыкладзе 9.5 відарыс і тэкст змяшчаюцца ў блоку.

У дачыненні да тэга можна ўжываць атрыбут alt, які ўстанаўлівае альтэрнатыўны тэкст для відарысаў. Такі тэкст дазваляе атрымаць тэкставую інфармацыю пра малюнак пры адключанай у браўзеры загрузцы відарысаў (прыклад 9.6).

Відарыс на вэб-старонцы можна зрабіць спасылкай на іншы файл, змясціўшы тэг у кантэйнер <a>. Калі пры гэтым вакол відарыса адлюстроўваецца рамка, то зняць яе можна, вызначыўшы значэнне none для стылявой уласцівасці border (прыклад 9.7).

9.2. Гук і відэа на вэб-старонках

У цяперашні час, выкарыстоўваючы Інтэрнэт, можна перадаваць не толькі тэкставыя даныя і відарысы, але і відэа, гук і ўсё тое, што называецца мультымедыя.

Пад тэрмінам **мультымедыя** разумеюць камп'ютарнае ўяўленне інфармацыі, што складаецца з больш чым аднаго тыпу даных, да якіх можна залічыць тэкст і гук, відэа і гук.

Гук на вэб-старонцы можна выкарыстоўваць як адзін з дэкаратыўных кампанентаў. Аднак сустракаюцца вэб-сайты, на якіх ён ператвараецца з дэкаратыўнага кампанента ў асноўны (сайты музычнай тэматыкі, музычныя архівы).

Кантэйнер <audio> дабаўляе, прайграе і кіруе настройкамі аўдыязапісу

на веб-страницы. Шлях да файла задаецца праз атрыбут `src` (прыклад 9.8).

Атрыбуты тэга `<audio>`:

- `src` — паказвае шлях да файла, які гучыць;
- `autoplay` — гук прайграецца адразу пасля загрузкі старонкі;
- `controls` — дабаўляе панэль кіравання гукам;
- `loop` — паўтарае прайграванне гучу пасля яго завяршэння;
- `preload` — выкарыстоўваецца для загрузкі файла разам з загрузкай веб-старонкі.

Кіраванне прайграваннем гучу адрозніваецца паміж браўзерамі, але асноўныя элементы супадаюць. Унутры кантэйнера `<audio>` можа змяшчацца тлумачальны тэкст. Ён будзе выводзіцца ў браўзерах, якія не працуюць з гэтым тэгам.

Кантэйнер `<video>` дабаўляе, прайграе відэаролік і кіруе яго настройкамі на веб-страницы. Шлях да файла задаецца гэтак жа, як і пры выкарыстанні гучу (прыклад 9.9).

Атрыбуты тэга `<video>`:

- `src` — паказвае шлях да файла, які прайграецца;
- `autoplay` — відэа пачынае прайгравацца аўтаматычна пасля загрузкі старонкі;
- `controls` — дабаўляе панэль кіравання да відэароліка;
- `loop` — паўтарае прайграванне відэа пасля яго завяршэння;
- `height` — задае вышыню вобласці для прайгравання;

Пры дабаўленні відэа на веб-старонкі трэба звярнуць увагу на файлы з расшырэннямі `.webm` і `.mp4`.

MP4 — фармат, які дазваляе захоўваць відэа высокай якасці і аптымальнай партатыўнасці з пункту гледжання захоўвання файлаў.

WebM — адкрыты фармат для мультимедыя-файлаў, выдадзены кампаніяй Google. Не патрабуе ліцэнзійных пагадненняў.

Прыклад 9.8. Дабаўленне гучу.

```
<audio controls width = "190px" src =
"muz.mp3">
Тег audio не падтрымліваецца вашым
браўзерам.
</audio>
```



Прыклад 9.9. Дабаўленне відэароліка.

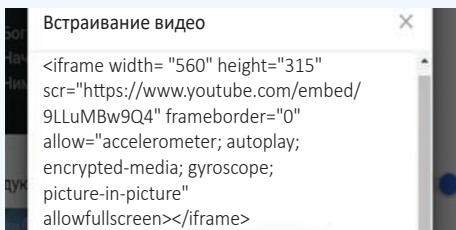
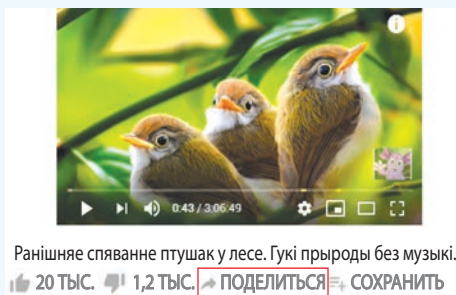
```
<video controls width = "190px" src =
="vid.mp4">
</video>
```



Для ўніверсальнага прайгравання ў розных браўзерах аўдыя і відэа кадзіруюць з дапамогай розных кодэкаў і адначасова ва ўкладзеным тэгу `<source>` дабаўляюць спасылкі на файлы розных фарматаў:

```
<video width = "400" controls =
"controls">
<source src = "duel.mp4" type = 'video/
mp4; codecs = "avc1.42E01E, mp4a.40.2">
<source src = "video/duel.webm" type =
'video/webm; codecs = "vp8, vorbis"'>
</video>
```

Прыклад 9.10. Дабаўленне відэароліка з сэрвіса YouTube.



•width — задае шырыню вобласці для прайгравання;

•preload — выкарыстоўваецца для загрузкі відэа разам з загрузкай вэб-старонкі.

Трэба заўсёды ўключаць у тэг <video> атрыбуты шырыні і/ці вышыні. Калі значэнні гэтых атрыбутаў не зададзены, браўзеру не вядомы памер відэа. У гэтым выпадку старонка пачне абнаўляцца пры загрузцы відэа, і будзе стварацца эфект мігання.

Кожнае відэа, змешчанае на вэб-сэрвісе YouTube, таксама можна дабавіць на створаную вэб-старонку. Для гэтага неабходна націснуць кнопку **Поделиться** і выбраць **Встроить**, а затым скапіраваць атрыманы html-код у адпаведнае месца html-дакумента (прыклад 9.10).


1. У якой якасці можна выкарыстоўваць графіку на вэб-старонках?
2. Якім павінен быць відарыс, што выкарыстоўваецца як фон вэб-старонкі?
3. У якім тэгу і з дапамогай якога атрыбута на вэб-старонку дабаўляецца фонавы відарыс?
4. Які тэг дазваляе дабавіць на вэб-старонку ілюстрацыю?
5. Якім чынам вызначаецца становішча ілюстрацыі на вэб-старонцы?
6. Для чаго прызначаны атрыбут alt?
7. Як зрабіць відарыс спасылкай?
8. З дапамогай якіх тэгаў на вэб-старонку дабаўляецца гук і відэа?
9. Як на вэб-старонку дабавіць відэа з вэб-сэрвіса?

Практыкаванні

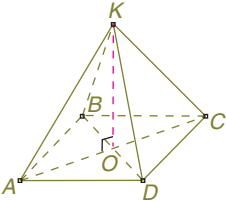
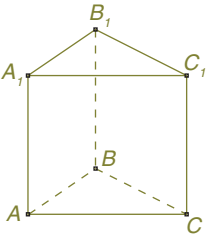
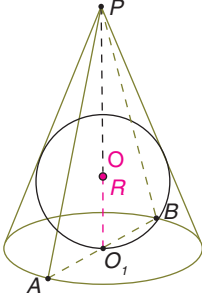
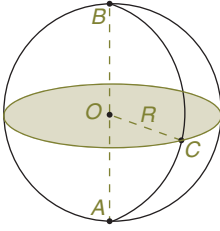
- 1 Стварыце вэб-старонку наступнай структуры:



Блок 1:

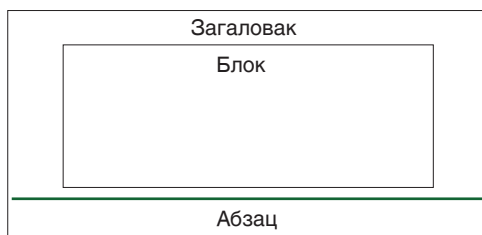
| Адлюстраванне ў браўзеры | Стылявое апісанне |
|---|---|
|  | <pre>div.bl1 { width: 300px; float: left; padding: 15px; margin: 30px; font: 30pt Tahoma; color: #808000; }</pre> |

Блок 2:

| Адлюстраванне ў браўзеры | Стылявое апісанне |
|---|---|
| <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Піраміда</p>  </div> <div style="text-align: center;"> <p>Прызма</p>  </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 20px;"> <div style="text-align: center;"> <p>Конус</p>  </div> <div style="text-align: center;"> <p>Сфера</p>  </div> </div> | <pre>div.bl2 { width: 600px; background: #ffffff; padding: 15px; margin: 30px; border: solid 1px black; font: 14pt Tahoma; float: left; }</pre> |


Захавайце вынік як файл index.html. Усе малюнкi, якія выкарыстоўваюцца на вэб-старонцы, захоўвайце ў асобнай папцы.

- 2 Для кожнай фігуры з практыкавання 1 стварыце вэб-старонку наступнай структуры:



Элементы старонак:

| Элемент | Прыклад адлюстравання ў браўзеры | Стылявое апісанне |
|-----------|---|--|
| Загалолак | | <pre>h1{ text-align: center; font-family: Comic Sans MS; color: #DC143C; }</pre> |
| Блок | <p>Піраміда — шматграннік, адна з граней якога (яна называецца асновай) — адвольны многавугольнік, а іншыя грані (бакавыя межы) — трохвугольнікі, што маюць агульную вяршыню.</p> | <pre>div { width: 620px; height: 300px; background: #ffffff; border: solid 1px black; padding: 15px; margin: 0 auto; font: 14pt Arial; line-height: 1.5; /*Міжрадковы інтэрвал*/ }</pre> |
| Абзац | | <pre>p { font: 14pt Tahoma; text-align: center; }</pre> |

| Элемент | Прыклад адлюстравання ў браўзеры | Стылявое апісанне |
|---------|---|--|
| Лінія |  | <pre>hr { color: #DC143C; background-color: #DC143C; /*Колер лініі для розных браўзераў*/ height: 2px; width: 700px; }</pre> |

Стылявое апісанне выканайце ў асобным файле. Падключыце стылявы файл да кожнай старонкі. Захавайце старонкі як 1.html, 2.html, 3.html, 4.html.

3 Звяжыце старонку index.html і старонкі, створаныя ў практыкаванні 2, з дапамогай перакрываваемых гіперспасылак. У якасці спасылак на старонцы index.html выкарыстайце відарысы фігур, а на старонках другога ўзроўню — тэкставыя элементы абзаца.

4 Дабаўце на старонку index.html гук (прайграванне пры загрузцы).

5 На адну са старонак практыкавання 2 дабаўце відэаролік адпаведнай тэматыкі. Для пошуку відэароліка выкарыстайце вэб-сэрвіс YouTube.

§ 10. Візуальнае вэб-канструяванне

Візуальны метада дазваляе аўтаматызаваць працэс канструявання вэб-сайтаў. Гэты метада узнік з неабходнасці паменшыць працаёмістасць і час стварэння сайтаў.

Сутнасць метада адлюстравана ў прынцыпе WYSIWYG (ад англ. *What you see is what you get* — «Што бачыш, тое і атрымаеш»). Пры стварэнні сайтаў з выкарыстаннем візуальнага метада трэба выбраць інструменты: рэдактары візуальнага канструявання, ці анлайн-канструктары сайтаў. У § 6 згадваліся некаторыя з іх. Выкарыстанне рэдактараў забяспечваюць карыстальніку магчымасць працаваць без непасрэднага падключэння да Інтэрнэту (прыклад 10.1).

Прыклад 10.1. Візуальны вэб-рэдактар Adobe Dreamweaver.

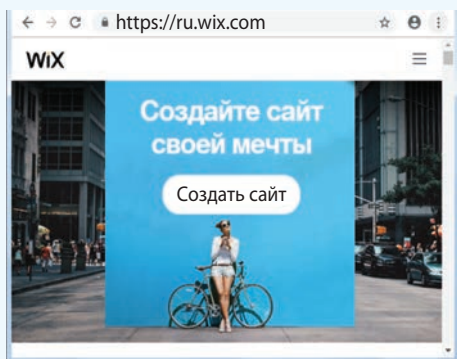
З магчымасцямі гэтага рэдактара можна пазнаёміцца, калі перайсці па спасылцы:

<https://www.adobe.com/products/dreamweaver.html>

Dreamweaver не толькі генеруе html-код, але таксама мае розныя карысныя інструменты, такія як бібліятэкі фрагментаў кода, кіраванне файламі, сервер наладкі і інш. Выкарыстоўваючы гэты рэдактар, можна праглядаць CSS інфармацыю ў адной уніфікаванай CSS-панэлі, што дазваляе лёгка ўбачыць стылі ў адносінах да пэўных элементаў і рэдагаваць існуючыя стылі.

Прыклад 10.2. Канструктары сайтаў.

1. Wix — <https://ru.wix.com/>

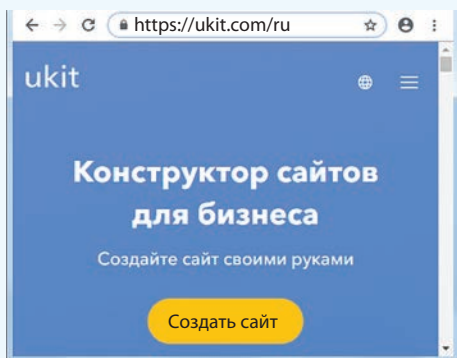


Адзін з самых папулярных канструктараў сайтаў. Ён дазваляе ствараць рэсурсы любога фармату, пачынаючы з благаў і візітак і заканчваючы анлайн-магазінамі. На выбар карыстальнікаў прапануецца больш за 500 шаблонаў.

Сэрвіс адрозніваецца інтуітыўна зразумелым інтэрфейсам, але мае і недахопы:

- два найбольш даступныя тарыфныя планы з пяці прапануюць абмежаваную функцыянальнасць;
- самы танны тарыфны план, як і бясплатны, не дазваляе выдаліць рэкламу Wix;
- нельга выбраць кароткі і запамінальны дамен.

2. Ukit — <https://ukit.com/ru>



У далейшым сайт можна змясціць у Інтэрнэце. Пры выкарыстанні рэдактараў для стварэння вэб-сайтаў html-код генеруецца ў адпаведнасці з візуальным уяўленнем вэб-старонкі.

У сучасным свеце кожная арганізацыя павінна мець свой сайт, інакш яна робіцца неканкурэнтаздольнай. У сувязі з гэтым кожны дзень па ўсім свеце з'яўляюцца тысячы новых сайтаў. І, як вынік, актыўна развіваецца індустрыя шаблонных сайтаў на канструктарах.

Канструктары вэб-сайтаў уяўляюць сабой скончаныя рашэнні для стварэння і наступнага кіравання вэб-сайтамі любой складанасці і прызначэння. Вэб-канструктары дазваляюць аўтаматызаваць працэсы стварэння не толькі гіпертэксту, але і структуры, і дызайну вэб-сайта.

Некаторыя канструктары сайтаў прапануюць комплексныя рашэнні для пабудовы сайтаў рознай накіраванасці «пад ключ», пры гэтым не патрабуюцца ўстанаўліваць ніякае праграмае забеспячэнне, патрэбен толькі доступ у Інтэрнэт (прыклад 10.2).

Перавагі канструктараў сайтаў:

1. Цана. Пры выкарыстанні некаторых канструктараў можна стварыць сайт бясплатна, а потым загрузіць яго ў Інтэрнэт на хостынг.

2. Хуткасць стварэння. Сайт можна стварыць за некалькі гадзін.

3. Хуткасць работы. Сайты ўжо аптымізаваны пад пляцоўкі, на якіх яны знаходзяцца, і будуць хутка працаваць.

4. Адсутнасць праграмавання. Для стварэння сайта не трэба ведаць ні моў вэб-праграмавання, ні моў разметкі.

5. Гатовая структура сайта. Пры стварэнні сайта падаецца ўжо гатовая структура з прыкладамі.

Выходзячы з усяго вышэйсказанага, канструктар сайтаў можа быць эфектыўным інструментам для тых, хто абмежаваны ў часе, ці для тых, каму патрэбен прасты сайт, а таксама ў выпадку, калі няма ўпэўненасці ў дызайнерскіх здольнасцях.

Прыклад 10.2. Працяг.

Канструктар Ukit арыентаваны на невялікія фірмы, якія без лішніх затрат і намаганняў хочуць атрымаць прыгожы і зручны сайт. Сэрвіс прапануе некалькі сотняў шаблонаў.

Бесплатна карыстацца канструктарам можна толькі ў межах пробнага перыяду, пасля якога неабходна падключыць адзін з тарыфаў. Самы даступны з іх ужо ўключае ў сябе дамен выгляду site.ru, неабмежаваную колькасць старонак і месца на серверы.

Пробны перыяд складае 14 дзён.

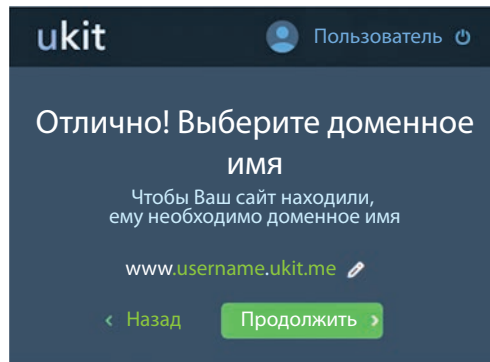


1. Што ўяўляе сабой метада візуальнага вэб-канструявання?
2. Якія інструменты стварэння сайтаў выкарыстоўваюцца пры візуальным метада вэб-канструявання?
3. Якімі перавагамі валодаюць канструктары сайтаў?



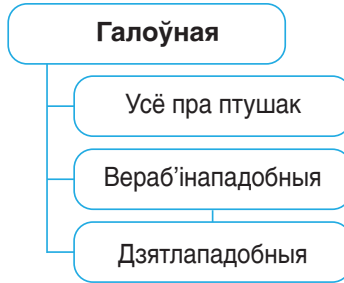
Практыкаванні

- 1 Зарэгіструйцеся на сайце <http://ukit.com>¹ і пацвердзіце рэгістрацыю праз спасылку ў электронным пісьме.
- 2 Выкарыстоўваючы папярэдні прагляд, разгледзьце прапанаваныя шаблоны. Выберыце шаблон для будучага сайта з катэгорыі «Універсальныя».
- 3 Увядзіце даменнае імя сайта, змяніўшы username (напрыклад, *birds*).



¹ Усе практыкаванні пасля § 22 можна выканаць з дапамогай іншага канструктара сайтаў, напрыклад <https://ru.wix.com/>

4 На аснове выбранага шаблону стварыце сайт наступнай структуры:



Для гэтага:

1. Перайдзіце на панэль **Страницы сайта**
2. У раздзеле **Основные** пераймянуіце старонкі. Лішнія старонкі выда-ліце. Пры неабходнасці стварыце новыя.

| Перайменаванне | Выдаленне |
|----------------|-----------|
| | |

3. Памяняйце парадак старонак (калі трэба, старонкі можна перацягваць).

4. Памяняйце назвы старонак у меню (для адкрыцця панэлі **Меню** проста вылучыце меню на старонцы).

5 У шапцы сайта змясціце лагатып з іконкай, якая адпавядае тэматыцы сайта. Для замены іконкі выканайце пералічаныя дзеянні.

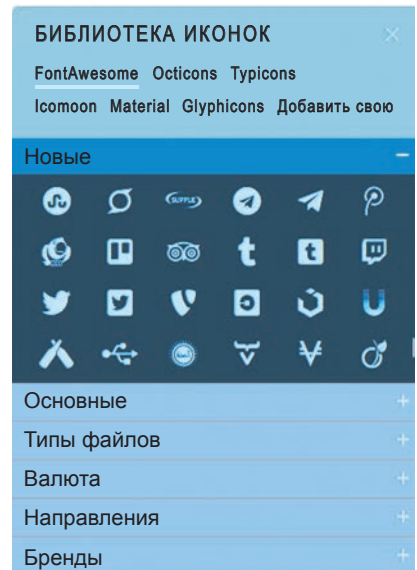
5.1. Выканайце пстрычку па наяўнай у шаблоне іконцы.

5.2. Націсніце **Заменить**.

5.3. У акне **Библиотека** пстрыкніце па неабходнай іконцы.

5.4. Настройце колер і памер іконкі.

6 У ніжні калантытул сайта дабайце персанальныя даныя (прозвішча, імя, клас).



- 7 Змяніце змест старонак у адпаведнасці з тэматыкай сайта (блокі можна выдаляць/дабаўляць, змяшчаць у іх свой тэкст, малюнкi і іншыя аб'екты). Матэрыялы для афармлення старонак можна ўзяць з сайта <http://e-vedy.adu.by>.
- 8 На старонцы «Пра птушак» змясціце:
 1. Загаловак 1 <Назва старонкі>.
 2. Тэкст пра птушак.
 3. Слайдэр <Фота птушак>.
- 9 На старонках «Вераб'інападобныя» і «Дзятлападобныя» змясціце:
 1. Загаловак 1 <Назва старонкі>.
 2. Кантэнт <Назва, фота і апісанне птушкі> — у 3 экзэмплярах.
- 10 Знайдзіце ў Інтэрнэце інфармацыю для старонкі «Гусепадобныя». Аформіце яе аналагічна старонкам «Вераб'інападобныя» і «Дзятлападобныя».
- 11 Старонку «Галоўная» аформіце самастойна (прадумайце змест і дызайн).



§ 11. Распрацоўка фрагментаў тэматычных сайтаў

Выкарыстоўваючы Інтэрнэт для пошуку інфармацыі, вы звычайна наведваеце дзясяткі разнастайных сайтаў. Шмат якія з іх прыцягваюць увагу сваім знешнім афармленнем, зручна змешчанай інфармацыяй, добра прадуманымі сродкамі навігацыі. Але ёсць і такія, што выклікаюць раздражненне доўгай загрузкай ці няўдала выбраным дызайнам, які перашкаджае ўспрымаць інфармацыю, а часам проста не змяшчаюць ніякай карыснай інфармацыі. Наведваць яшчэ раз такія сайты жадання не ўзнікае. Таму, прыступаючы да распрацоўкі ўласнага сайта, трэба старанна прадумваць шмат якіх пытанні.

Разгледзім этапы распрацоўкі вэб-сайта (прыклад 11.1).

1. Праектаванне сайта — пастаноўка мэт і задач сайта.

Вызначыце, інфармацыю якога тыпу вы хочаце змясціць на старонках

Прыклад 11.1. Стварэнне сайта «Паэтычныя вобразы ў рускай літаратуры».

Этапы распрацоўкі

1. Распрацаваць сайт па вучэбным прадмеце «Руская літаратура», 11-ы клас.
2. Структура сайта.



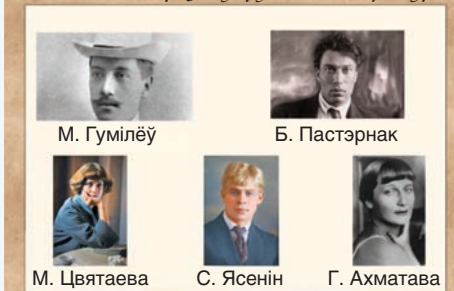
Відарысы на галоўнай старонцы будуць з'яўляцца гіперспасылкамі на старонкі другога ўзроўню. На кожнай старонцы другога ўзроўню будуць спасылкі на суседнюю і галоўную старонку.

3. Дызайн сайта.

Галоўная старонка. Змяшчае заглавак і блок з відарысамі і персанальнымі данымі рускіх паэтаў пачатку ХХ ст.

Прыклад 11.1. Працяг.

Фонавы відарыс старонкі — імітацыя паперы. Фон блока — суцэльны колер пастэльнага адцення.

Наэтычныя вобразы ў рускай літаратуры

Старонкі другога ўзроўню. Змяшчаюць заглавак, блок з ілюстрацыяй і тэкстам верша, гарызантальную лінію і блок з элементамі навігацыі. Колеравае рашэнне імкнецца да мінімалізму. Фонавы відарыс і ілюстрацыя адрозніваюцца для кожнай з пяці старонак.



сайта; для якой аўдыторыі і ўзроставай групы прызначаны сайт. Ад гэтага будзе залежаць і знешняе афармленне, і форма падачы матэрыялу, і арганізацыя навігацыі.

2. Распрацоўка структуры сайта.

Не трэба змяшчаць на адной старонцы вельмі вялікі аб'ём інфармацыі, бо гэта перашкаджае яе ўспрымання. Лепш стварыць некалькі звязаных старонак, змясціўшы на іх інфармацыю, згрупаваную па тэматыцы.

Навігацыю сайта неабходна старанна прадумаць. Асноўныя раздзелы сайта павінны быць даступныя з галоўнай старонкі. На кожнай старонцы неабходна спасылка, з дапамогай якой наведвальнік зможа вярнуцца на галоўную старонку.

3. Стварэнне дызайну сайта (знешняе афармленне сайта мае немалаважнае значэнне для яго наведвальнасці).

Перавагу маюць светлыя і кантрасныя колеравыя рашэнні (пры выбары колеравага афармлення можна выкарыстаць рэкамендацыі прафесіяналаў).

Не трэба перагружаць сайт вялікай колькасцю графічнай інфармацыі, што запавольвае яго загрузку.

Дызайн галоўнай старонкі ствараецца асобна і адрозніваецца ад дызайну тыпавых старонак.

На галоўнай старонцы не павінна быць тэксту вялікага аб'ёму.

4. Стварэнне мультымедыя-элементаў.

5. Вёрстка старонак і шаблонаў (стварэнне html-кода).

6. Праграмаванне (у выпадку распрацоўкі складанага, шматфункцыянальнага сайта).

7. Напаўненне кантэнтам.

8. Тэсціраванне і занясенне карэкціровак.

9. Публікацыя сайта на хостынг.

10. Абслугоўванне сайта, які працуе, ці яго праграмнай асновы.

У залежнасці ад пастаўленай задачы некаторыя этапы могуць адсутнічаць.

Стварэнне сайта з'яўляецца даволі складанай работай. Ад кожнага этапу і беспамылковай яго рэалізацыі залежыць якасць усяго сайта.

Прыклад 11.1. Працяг.

4. На кожнай старонцы другога ўзроўню прысутнічае адпаведнае гукавое суправаджэнне.

5. Пры вёрстцы старонак выкарыстоўваецца ручны метад і рэдактар html-кода. Падключаем табліцу стыляў для галоўнай старонкі. Для старонак другога ўзроўню ствараем і падключаем файл з апісаннем стыляў.

6. Сайт змяшчае толькі статычныя старонкі. Для іх стварэння не патрабуецца выкарыстоўваць мовы вэб-праграмавання.

7. Для пошуку тэкстаў вершаў і відарысаў паэтаў выкарыстоўваем Інтэрнэт.

8. Праводзім паэтапае (для кожнай старонкі) і канчатковае (правяраем функцыянальнасць навігацыі) тэсціраванне.

9—10. Мэта стварэння сайта — навучанне. Хостынг і абслугоўванне не ажыццяўляюцца.



1. Якія этапы можна вылучыць у працэсе стварэння сайта?
2. Чаму пры распрацоўцы сайта трэба прытрымлівацца пэўных этапаў?
3. У чым сутнасць кожнага этапу стварэння сайта?



Практыкаванні

1. Дапоўніце сайт з прыкладу 11.1 патрэбнымі старонкамі.
2. Стварыце сайт (можна выкарыстоўваць узор з прыкладу). Прапанаваная тэматыка:
 1. Найбуйнейшыя транскантынентальныя кампаніі.
 2. Асноўныя прамысловыя прадпрыемствы нашай краіны.
 3. Мадэлі крышталічных рашотак.
 4. Электрычны ток у розных асяроддзях.
 5. Міфалагічныя асновы назвы сузор'яў.
 6. Трыганаметрычныя функцыі.
 7. Сістэма ўтварэння Рэспублікі Беларусь.

Глава 3

ІНФАРМАЦЫЙНЫЯ ТЭХНАЛОГІІ Ў ГРАМАДСТВЕ

§ 12. Інформаційныя сістэмы, тэхналогіі і рэсурсы

Прыклад 12.1. Разнастайнасць сістэм.

Сістэмы ў прыродзе: сонечная сістэма, жывы арганізм, расліна і інш.

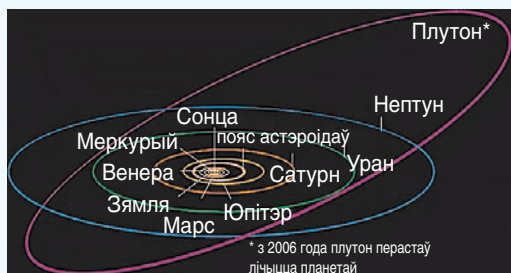
Тэхнічныя сістэмы: камп'ютар, камп'ютарная сетка, электрастанцыя, аўтамабіль і інш.

Сацыяльныя сістэмы: школа, сацыяльная сетка, горад і інш.

Нематэрыяльныя сістэмы: гутарковая мова, мова хімічных формул, нотны запіс, сістэма праграмавання і інш.

Прыклад 12.2. Структура сістэмы.

Структура сонечнай сістэмы:



Структура камп'ютарнай сеткі:



12.1. Інформаційныя сістэмы

Рэальныя аб'екты навокал чалавека вельмі складаныя, таму іх прынята разглядаць як сістэму. Сістэма складаецца з элементаў, якія звязаны паміж сабой. Усе разам яны выконваюць агульныя функцыі, што дазваляе разглядаць іх як адно цэлае (прыклад 12.1).

Сукупнасць вылучаных адносін (узаемасувязей) паміж элементамі сістэмы прынята называць **структурай** сістэмы. Адметнай асаблівасцю сістэмы з'яўляецца наяўнасць у яе такіх якасцей ці функцый, якія не ўласцівыя ніводнаму яе элементу, ніводнай яе падсістэме, узятым паасобку. Структура сістэмы можа змяняцца ў выніку працякання якіх-небудзь працэсаў: элементы сістэмы могуць дабаўляцца ці выдаляцца. Выдаленне элемента сістэмы ці з'яўленне новага заўсёды прыводзяць да змянення ўзаемасувязей паміж элементамі (прыклад 12.2).

Сістэма характарызуецца функцыямі, прызначэннем, уваходамі і выходамі, унутраным станам. Сістэма ацэньваецца пэўным наборам якасных і колькасных паказчыкаў — параметраў сістэмы.

Паняцці «інфармацыя», «інфармацыйны працэс», «інфармацыйная сістэма» ўзаемазвязаны. Інфармацыя выяўляецца ў інфармацыйных працэсах, якія працякаюць толькі ў рамках якой-небудзь інфармацыйнай сістэмы.

Інфармацыйная сістэма — гэта сістэма, элементамі якой з'яўляюцца даныя, тэхнічныя сродкі, метады і спецыялісты, а сувязі ўтвараюцца дзякуючы працяканню інфармацыйных працэсаў. Наяўнасць інфармацыі (даных) і інфармацыйных працэсаў дазваляе гаварыць пра інфармацыйную сістэму (прыклад 12.3).

Інфарматыка вывучае заканамернасці працякання інфармацыйных працэсаў у сістэмах рознай прыроды, але ў найбольшай ступені прадметам яе даследаванняў з'яўляюцца інфармацыйныя працэсы ў тэхнічных і сацыял-тэхнічных сістэмах. Прычым гэтыя заканамернасці важныя з пункту гледжання магчымасці аўтаматызацыі гэтых працэсаў. Пад **аўтаматызаванай інфармацыйнай сістэмай (АІС)** разумеюцца сукупнасць інфармацыйных масіваў, тэхнічных, праграмных і моўных сродкаў і персаналу, прызначаных для збору, захоўвання, пошуку, апрацоўкі і выдачы даных па запытах карыстальнікаў.

Аўтаматызаваныя інфармацыйныя сістэмы ўжываюцца практычна ва ўсіх сферах чалавечай дзейнасці: у кіраванні вытворчасцю ці прадпрыемствам; пры арганізацыі навуковых даследаванняў і выкананні канструктарскіх і праектных работ; у бібліятэчнай справе; у навучанні і г. д.

У прыкладзе 12.4 пералічаны аўтаматызаваныя інфармацыйныя сістэмы розных відаў.

Дзяленне аўтаматызаваных інфармацыйных сістэм на віды дастаткова

Прыклад 12.3. Інфармацыйная сістэма.



Прыклад 12.4. Віды АІС:

- *інфармацыйна-даведчыя* (ІДС) — разнастайныя электронныя слоўнікі, электронныя энцыклапедыі, электронныя запісныя кніжкі і г. д.;
- *інфармацыйна-пошукавыя сістэмы* (ІПС) — найбольш вядомымі з'яўляюцца сусветная павуціна (WWW) з адапаведнымі пошукавымі сістэмамі (Google, Yandex і інш.) і юрыдычныя ІПС, прызначаныя пераважна для захоўвання дакументаў афіцыйнага характару (законаў, палажэнняў, інструкцыйных лістоў і інш.), выдадзеных дзяржаўнымі органамі;
- *геаінфармацыйныя сістэмы* (ГІС). У іх інфармацыя пра аб'екты ўпарадкавана ў адпаведнасці з прасторавым размяшчэннем аб'ектаў, паказаных часцей за ўсё на геаграфічных картах;
- *вымяральныя ІС* выкарыстоўваюцца для аўтаматычнага (з дапамогай спецыяльных датчыкаў) збору інфармацыі пра стан і параметры цікавага аб'екта. Ужываюцца вымяральныя АІС у медыцыне, метэаралогіі, сейсмалогіі, пры арганізацыі касмічных палётаў, на атамных электрастанцыях, на вытворчасцях, шкодных для здароўя чалавека, і г. д.;
- *сістэмы аўтаматызацыі навуковых даследаванняў* забяспечаны сродкамі для пабудовы інфармацыйных мадэлей самага рознага віду;

Прыклад 12.4. Працяг.

• *экспертныя сістэмы (ЭС) і сістэмы падтрымкі прыняцця рашэнняў (СППР).* Их аснову складаюць базы ведаў (БВ) па пэўнай прадметнай галіне. Дадзеныя сістэмы актыўна выкарыстоўваюцца пры планаванні і складанні доўгатэрміновых прагнозаў у прамысловасці, для пастаноўкі дыягназу ў медыцыне, для выбару найбольш імавернай версіі ў юрыспрудэнцыі і г. д.;

• *навучальныя АІС* — разнастайныя электронныя падручнікі, камп'ютарныя тэсты, навучальныя праграмы, а таксама трэнажоры, якія імітуюць работу якога-небудзь устройства (самалёта, аўтамабіля і г. д.).

• *сістэмы аўтаматызаванага праектавання (САПР);*

• *аўтаматызаваныя сістэмы кіравання (АСК);*

• ІС, якія забяспечваюць *аўтаматызацыю дакументазвароту і ўліку.*

Паняцце «штучны інтэлект» было дадзена Джонам Макараці ў 1956 г. на канферэнцыі ў Дартмуцкім універсітэце.



Джон Макараці (1927—2011) — амерыканскі інфарматык, аўтар тэрміна «штучны інтэлект», вынаходнік мовы Лісп, заснавальнік функцыянальнага праграміравання, лаўрэат прэміі Цьорынга.

Па словах Д. Макараці: «Праблема складаецца ў тым, што пакуль мы не можам у цэлым вызначыць, якія вылічальныя працэдуры мы хочам называць інтэлектуальнымі».

ўмоўнае, рэальная АІС можа спалучаць у сабе магчымасці сістэм рознага віду. Большасць сучасных аўтаматызаваных інфармацыйных сістэм у сваёй рабоце выкарыстоўваюць магчымасці камп'ютара і камп'ютарных сетак. Важнымі кампанентамі аўтаматызаваных інфармацыйных сістэм з'яўляюцца базы і банкі даных.

Важнымі напрамкамі развіцця сучасных інфармацыйных сістэм з'яўляюцца сістэмы штучнага інтэлекту (ШІ). **Штучны інтэлект** (ШІ; англ. artificial intelligence, AI) — уласцівасць інтэлектуальных сістэм выконваць творчыя функцыі, якія традыцыйна лічацца прэрагатывай чалавека. У гэтым напрамку развіваецца робата-тэхніка, пошукавыя сістэмы, сістэмы аўтаматызаванага кіравання і г. д. Вынікі, атрыманыя пры стварэнні і эксплуатацыі сістэм штучнага інтэлекту, выкарыстоўваюцца зараз у сістэмах распазнавання маўлення і відарысаў, перакладах тэкстаў з адной мовы на другую, пастаноўцы дыягназаў, у сферы фінансаў і сферы камп'ютарных гульняў.

Інтэлектуальныя сістэмы, якія існуюць на сёняшні дзень, маюць вельмі вузкія галіны ўжывання. Напрыклад, праграмы, здольныя распазнаць твар чалавека, не могуць гуляць у шахматы. Развіццё сістэм штучнага інтэлекту, набліжанага да чалавечага, — задача будучыні.

12.2. Інфармацыйныя тэхналогіі

Азначэнне інфармацыйных тэхналогій цесна звязана з паняццем «тэхналогія». Тэрмін **тэхналогія** паходзіць ад грэчаскага слова *technē*, якое азначае «навука пра ўменне, майстэрства, мастацтва». З дапамогай тэхналогій апісваюцца шмат якія эканамічныя, сацыяльныя, культурныя і іншыя працэсы, што адбываюцца ў грамадстве (прыклад 12.5).

Інфармацыйная тэхналогія — сукупнасць спосабаў, прыёмаў і метадаў збору, апрацоўкі і перадачы даных (першасная інфармацыя) для атрымання інфармацыі новай якасці пра стан аб'екта, з'явы ці працэсу (інфармацыйны прадукт). Інфармацыйныя тэхналогіі ажыццяўляюцца ва ўсіх галінах чалавечай дзейнасці з выкарыстаннем сучасных сродкаў сувязі, вылічальнай тэхнікі і праграмнага забеспячэння.

Інструментамі інфармацыйных тэхналогій з'яўляюцца апаратнае, праграмнае і матэматычнае забеспячэнне. З іх дапамогай выконваецца ператварэнне першасных даных у інфармацыйны прадукт.

Тэхналагічны працэс па апрацоўцы інфармацыі звычайна разбіваюць на этапы. Кожны этап складаецца з аперацый і дзеянняў. У выніку выканання аперацый атрымліваюць пэўны інфармацыйны прадукт, які адпавядае мэтам бягучага этапу. Пад дзеяннямі разумеюць сукупнасць прыёмаў работы з пэўным праграмным забеспячэннем, што прыводзяць да дасягнення пастаўленай мэты (прыклад 12.6).

Прыклад 12.5. Параўнанне тэхналогіі матэрыяльнай вытворчасці і інфармацыйнай тэхналогіі:



Прыклад 12.6. Разгледзім тэхналогію стварэння стылю ў тэкставым дакуменце.

У якасці этапаў можна вылучыць заданне параметраў абзаца, сімвала, ужыванне тэкставых эфектаў і інш.

Аперацыі на этапе задання параметраў сімвала — гэта выбар шрыфту, колеру, памеру, абрысу сімвала.

Дзеянні — выбар пэўных інструментаў тэкставага рэдактара для выканання аперацый: кнопкі, спіса, меню і інш.

Злачынствы ў сферы інфармацыйных тэхналогій, ці кіберзлачыннасць, — злачынствы, што здзяйснююцца людзьмі, якія выкарыстоўваюць інфармацыйныя тэхналогіі для злачынных мэт. Злачынствы ў сферы інфармацыйных тэхналогій уключаюць як распаўсюджванне шкодных вірусаў, узлом пароляў, крадзеж нумароў крэдытных картак і іншых банкаўскіх рэквізітаў (фішынг), гэтак і распаўсюджванне супрацьпраўнай інфармацыі (паклёпу, матэрыялаў парнаграфічнага характару, матэрыялаў, якія выклікаюць міжнацыянальную і міжрэлігійную варожасць і г. д.) праз Інтэрнэт.

Прыклад 12.7. Спосабы класіфікацыі інфармацыйных тэхналогіяў па розных класіфікацыйных прыметах:

Інфармацыйныя тэхналогіі

- паводле прызначэння і характару выкарыстання
- паводле карыстальніцкага інтэрфейса
- паводле тыпу апрацоўваемых даных
- паводле прынцыпу даных
- паводле ступені ахопу задач кіравання
- паводле спосабу кіравання вытворчай тэхналогіяй
- паводле характару ўдзелу тэхнічных сродкаў у дыялогу з карыстальнікам
- паводле віду прадметнай дзейнасці

Прыклад 12.8. Класіфікацыя інфармацыйных тэхналогіяў па тыпе даных, якія апрацоўваюцца:

Інфармацыйныя тэхналогіі

- тэхналогіі апрацоўкі тэкставай інфармацыі
- тэхналогіі апрацоўкі графічнай інфармацыі
- тэхналогіі работы з электроннымі табліцамі
- тэхналогіі работы з базамі даных
- мультымедычныя тэхналогіі
- сеткавыя тэхналогіі
- тэхналогіі праграмавання
- іншыя тэхналогіі

Інфармацыйная тэхналогія, як і кожная іншая, павінна:

- забяспечваць магчымасць разбіцця ўсяго працэсу апрацоўкі на этапы, аперацыі і дзеянні;
- мець рэгулярны характар, г. зн. этапы, дзеянні і аперацыі тэхналагічнага працэсу павінны быць стандартызаваны, каб пры кожным ужыванні тэхналогіі атрымліваць адна тыпны вынік, што дазволіць ажыццяўляць мэтанакіраванае кіраванне інфармацыйнымі працэсамі.

Для правільнага выкарыстання інфармацыйных тэхналогіяў у розных сферах жыцця грамадства неабходна іх папярэдняя класіфікацыя. У залежнасці ад выбару класіфікацыйных прымет існуюць розныя класіфікацыі інфармацыйных тэхналогіяў (прыклад 12.7).

У залежнасці ад выбару прыметы класіфікацыі інфармацыйных тэхналогіяў могуць падзяляцца на розныя групы (прыклад 12.8).

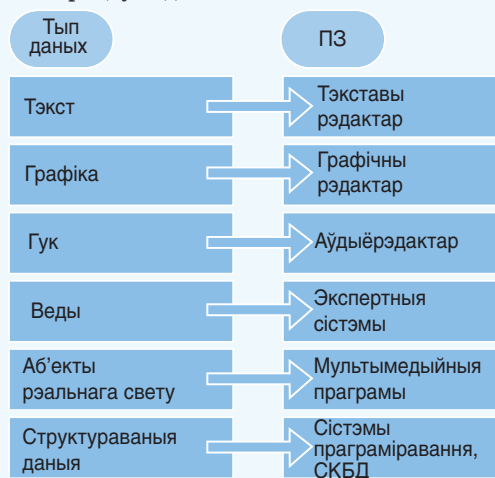
Для кожнага віду інфармацыйных тэхналогіяў вызначаны адзін ці некалькі праграмных прадуктаў для пэўнага тыпу камп'ютараў, тэхналогіі работы з якімі дазваляюць дасягнуць пастаўленай карыстальнікам мэты па апрацоўцы даных (прыклад 12.9).

У цяперашні час шырокае развіццё атрымалі **воблачныя інфармацыйныя тэхналогіі** — тэхналогіі размеркаванай апрацоўкі даных, у якой камп'ютарныя рэсурсы і магутнасці даюцца карыстальніку як інтэрнэт-сэрвіс. Воблачныя тэхналогіі забяс-

печваюць карыстальніку магчымасці для захоўвання даных і праграмныя сродкі для апрацоўкі даных. На сённяшні дзень у воблаку рэалізаваны практычна ўсе інфармацыйныя тэхналогіі, якія раней выкарыстоўваліся як лакальныя. Часткай воблачных тэхналогій з'яўляюцца так званыя **грыд-вылічэнні** (англ. *grid* — рашотка, сетка) — гэта форма размеркаваных вылічэнняў, у якой «віртуальны суперкамп'ютар» паказаны ў выглядзе кластараў, што складаюцца з асобных камп'ютараў, злучаных з дапамогай сеткі, і працуюць разам для выканання вялізнай колькасці задач (аперацый, работ). Гэта тэхналогія ўжываецца для рашэння навуковых, матэматычных задач, якія патрабуюць значных вылічальных рэсурсаў. Грыд-вылічэнні выкарыстоўваюцца таксама ў камерцыйнай інфраструктуры для рашэння такіх працаёмстых задач, як эканамічнае прагназаванне, сейсмааналіз, распрацоўка і вывучэнне ўласцівасцей новых лекаў і інш.

Тэхналогіі грыд-вылічэнняў ужываюцца сёння і для апрацоўкі вялікіх даных. Пад **вялікімі данымі** (англ. *big data*) разумеюць структураваныя і неструктураваныя даныя вялізных аб'ёмаў, якія эфектыўна апрацоўваюцца сучаснымі праграмнымі інструментамі, што з'явіліся ў канцы 2000-х гг. Вызначальнай характарыстыкай для вялікіх даных з'яўляецца набор прымет VVV («тры V») — *volume* (аб'ём), *velocity* (скорасць), *variety* (разнастайнасць)

Прыклад 12.9. Сувязь інфармацыйных тэхналогій з праграмнымі сродкамі апрацоўкі даных:



Тэрмін «грыд-вылічэнні» з'явіўся ў пачатку 1990-х гадоў як метафара, якая дэманструе магчымасць простага доступу да вылічальных рэсурсаў як да рэсурсаў электрычнай сеткі (англ. *power grid*). Выкарыстанне свабоднага часу працэсараў для рашэння даследчых задач, якія патрабуюць вялікіх вылічальных магчымасцей і самаахвотнага грыд-камп'ютынга (вылічэнне на камп'ютары), стала папулярным у канцы 1990-х гг. пасля запуску праектаў самаахвотных вылічэнняў GIMPS у 1996 г., *distributed.net* у 1997 г. і SETI@home ў 1999 г. Ідэі грыд-сістэмы (уключаючы ідэі з абласцей размеркаваных вылічэнняў, аб'ектна-арыентаванага праграмавання, выкарыстання камп'ютарных кластараў, вэб-сэрвісаў і інш.) былі сабраны і аб'яднаны Іэнам Фостэрам, Карлам Кесельманам і Стывам Цікі, якіх часта называюць бацькамі грыд-тэхналогіі. Яны пачалі стварэнне набору інструментаў для грыд-камп'ютынга Globus Toolkit.

Прыклад 12.10. Характарыстыкі вялікіх даных:

- аб'ём (англ. *volume*) характарызуе велічыню фізічнага аб'ёму даных;
- скорасць (англ. *velocity*) вызначае як імкліваць прыросту даных, так і неабходнасць высакаскораснай апрацоўкі даных для атрымання вынікаў;
- разнастайнасць (англ. *variety*) дазваляе адначасова апрацоўваць розныя тыпы структураваных і паўструктураваных даных.

Часам характарыстыкі пашыраюць («чатыры V», «пяць V» і г. д.). Акрамя традыцыйных трох, яшчэ дабаўляюць: *veracity* — пэўнасць, *viability* — жыццяздольнасць, *value* — каштоўнасць, *variability* — пераменлівасць, *visualization* — візуалізацыю.

Прыклад 12.11. Устройства віртуальнай рэальнасці:



Акуляры віртуальнай рэальнасці



Шлем віртуальнай рэальнасці з аксэсуарамі



Пакой віртуальнай рэальнасці

(прыклад 12.10). Тэхналогіі апрацоўкі вялікіх даных з'яўляюцца альтэрнатывай традыцыйным сістэмам кіравання данымі з дапамогай СКБД.

Неад'емнай часткай сучаснага свету становяцца тэхналогіі віртуальнай і дапоўненай рэальнасці.

Віртуальная рэальнасць — створаны тэхнічнымі сродкамі свет, які перадаецца чалавеку праз яго адчуванні: зрок, слых, нюх, пачуццё дотыку і інш. Да ўстройстваў віртуальнай рэальнасці залічваюць шлемы, акуляры, пакоі віртуальнай рэальнасці і інш. (прыклад 12.11).

Дапоўненая рэальнасць (англ. *augmented reality*, AR — «пашыраная рэальнасць») — тэхналогіі, якія дапаўняюць рэальны свет, дабаўляючы любыя сэнсарныя даныя. Дапоўненая рэальнасць рэалізуецца з дапамогай тэхналогій камп'ютарнага зроку, якія могуць працаваць у «разумных» акулярах і шлемах, мабільных устравках, інтэрактыўных стэндах і інш. (прыклад 12.12).

Тэхналогіі дапоўненай і віртуальнай рэальнасці выкарыстоўваюцца ў адукацыі і медыцыне, на іх базе распрацоўваюцца вучэбныя праграмы і трэнажоры. Яны знайшлі ўжыванне ў праграмных прадуктах для інжынераў, архітэктараў, дызайнераў, рыэлтараў і інш. Самы вялікі працэнт продажу праграмага забеспячэння ў галіне віртуальнай і дапоўненай рэальнасці прыпадае на сферу забаў:

відэагульні, кіно, анімацыя, серыялы і інш.

Грунтуючыся на навуковых прагнозах, можна гаварыць пра тое, што тэхналогіі віртуальнай і дапоўненай рэальнасці разам з вялікімі данымі, воблачнымі тэхналогіямі, штучным інтэлектам і некаторымі іншымі стануць ключавымі тэхналогіямі чарговай прамысловай рэвалюцыі.

Вялікая колькасць інфармацыйных тэхналогій распрацоўваецца ў сістэмах штучнага інтэлекту (прыклад 12.13). **Машыннае мысленне** (англ. *machine reasoning*) ахоплівае працэсы планавання, перадачы ведаў і разважанняў, пошук і аптымізацыю. Грунтам тэхналогіі **машыннага навучання** (англ. *machine learning, ML*) з'яўляецца не прамое рашэнне задачы, а навучанне ў працэсе ўжывання рашэнняў мноства падобных задач. **Робататэхніка** ўключае ў сябе кіраванне, успрыманне, датчыкі і прыводы, а таксама інтэграцыю ўсіх метадаў у кіберфізічныя сістэмы. **Біялагічнае мадэліраванне** штучнага інтэлекту ўключае ў сябе такія напрамкі, як нейронавыя сеткі, генетычныя алгарытмы, агентны падыход.

Інфармацыйныя тэхналогіі цесна звязаны з інфармацыйнымі сістэмамі. Рэалізацыя функцый інфармацыйнай сістэмы немагчыма без ужывання арыентаванай на яе інфармацыйнай тэхналогіі, хоць сама тэхналогія можа існаваць і па-за сферай інфармацыйнай сістэмы.

Прыклад 12.12. Устройства дапоўненай рэальнасці:



Акуляры дапоўненай рэальнасці

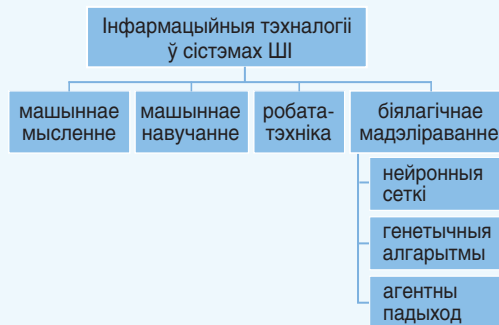


Дапоўненая рэальнасць на мабільным устройстве



Стэнды дапоўненай рэальнасці

Прыклад 12.13. Інфармацыйныя тэхналогіі ў сістэмах штучнага інтэлекту:



Нейронныя сеткі выкарыстоўваюцца для рашэння недакладных і складаных задач, такіх як распазнаванне выяў і класіфікацыя.

Генетычны падыход заснаваны на ідэі, што нейкі алгарыт можа стаць больш эфектыўным, калі пазычыць лепшыя характарыстыкі ў іншых алгарытмаў («бацькоў»).

Пры агентным падыходзе ствараецца праграма — інтэлектуальны агент — для ўзаемадзеяння з навакольным асяроддзем. Прыкладамі такіх задач з'яўляюцца анлайн-гандаль, ліквідацыя надзвычайных сітуацый, мадэліраванне сацыяльных структур і інш.

Прыклад 12.14. Нацыянальныя інфармацыйныя рэсурсы:

Нацыянальныя інфармацыйныя рэсурсы

- бібліятэчныя рэсурсы
- архіўныя рэсурсы
- навукова-тэхнічная інфармацыя
- прававая інфармацыя
- інфармацыя дзяржаўных структур
- галіновыя інфармацыя
- фінансавая і эканамічная інфармацыя
- інфармацыя пра прыродныя рэсурсы
- інфармацыя прадпрыемстваў і ўстаноў
- іншая інфармацыя

Прыклад 12.15. Інфармацыйныя рэсурсы грамадства адмежаваны ад людзей, якія іх аналізавалі і стваралі. Яны матэрыялізаваліся ў наступных відах:

- дакументы;
- кнігі;
- лічбавыя даныя;
- алгарытмы;
- творы мастацтва і навукі.

12.3. Інфармацыйныя рэсурсы

Традыцыйна ў якасці рэсурсаў разглядаюцца: матэрыяльныя, прыродныя, працоўныя, фінансавыя, энергетычныя і інш.

У сучасным грамадстве разглядаюць інфармацыйныя рэсурсы — рэсурсы, якія заўсёды існавалі, але не разглядаліся з пункту гледжання эканомікі.

Азначэнне інфармацыйнага рэсурсу дадзена ў законе РБ «Аб інфармацыі, інфарматызацыі і ахове інфармацыі» (2008 г). **Інфармацыйны рэсурс** — арганізаваная сукупнасць дакументаванай інфармацыі, якая ўключае базы даных, іншыя сукупнасці ўзаемазвязанай інфармацыі ў інфармацыйных сістэмах (прыклад 12.14).

Дакументы не існуюць самі па сабе. У іх у рознай форме пададзены веды, якімі валодалі людзі, што стварылі гэтыя дакументы. Таму інфармацыйнымі рэсурсамі з'яўляюцца веды, зафіксаваныя на матэрыяльным носьбіце і падрыхтаваныя людзьмі для сацыяльнага выкарыстання (прыклад 12.15).

Матэрыяльныя, фінансавыя і іншыя рэсурсы вымяраюць і ацэньваюць. У адносінах да інфармацыйных рэсурсаў крытэрыі ацэнкі іх якасці і колькасці пакуль не выпрацаваны канчаткова.

Развіццё інфармацыйных рэсурсаў фарміруе рынак інфармацыйных паслуг, дазваляе ператварыць дзейнасць

па аказанні інфармацыйных паслуг у глабальную агульназначальную дзейнасць (прыклад 12.16).

Інфармацыйныя рэсурсы з'яўляюцца асновай для стварэння **інфармацыйных прадуктаў**, якія могуць распаўсюджвацца з дапамогай **інфармацыйных паслуг**. Інфармацыйныя прадукты і паслугі фарміруюць **інфармацыйны рынак** — сістэму эканамічных, прававых і арганізацыйных адносін па гандлі прадуктамі інтэлектуальнай работы.

Асноўнай крыніцай інфармацыі для інфармацыйнага забеспячэння ў сучасным грамадстве з'яўляюцца інфармацыйныя сістэмы, якія на аснове баз даных інтэгруюць у сабе пастаўшчыкоў і спажывцоў інфармацыйных паслуг. Інфармацыйныя тэхналогіі, выкарыстоўваючы інфармацыйныя рэсурсы і працэсы, забяспечваюць парадак і ўмовы аказання інфармацыйных паслуг, а таксама сувязі паміж імі (прыклад 12.17).

У цяперашні час ва ўсіх развітых краінах адным з галоўных напрамкаў удасканалення сістэмы дзяржаўнага кіравання з'яўляецца пераход на аказанне дзяржаўных паслуг у электронным выглядзе. У Беларусі створана і паспяхова развіваецца сістэма аказання такіх паслуг грамадзянам і бізнесу¹.

Прыклад 12.16. Рынак інфармацыйных паслуг:

| | |
|--|--|
| Інфармацыя для спецыялістаў | навукова-тэхнічная прафесійная першакрыніца |
| Дзелавае інфармацыя | статыстычная камерцыйная біржавая фінансавая |
| Спажывецкая інфармацыя | навіны літаратура забавы мас-медыя |
| Паслугі адукацыі | дашкольная, школьная спецыяльная вышэйшая павышэнне кваліфікацыі і перападрыхтоўка |
| Інфармацыйныя сістэмы і сродкі па забеспячэнні | распрацоўка і суправаджэнне кансультаванне тэхнічныя сродкі крыніцы інфармацыі |

Прыклад 12.17. Узаемасувязь інфармацыйных тэхналогій, інфармацыйных рэсурсаў, інфармацыйных працэсаў і інфармацыйнага забеспячэння:



¹ Нацыянальны цэнтр электронных паслуг — <https://nces.by/e-government/>



1. Што такое інфармацыйная сістэма?
2. Якія віды аўтаматызаваных інфармацыйных сістэм вы можаце назваць?
3. Што такое інфармацыйныя тэхналогіі?
4. Па якіх прыметах можна класіфікаваць інфармацыйныя тэхналогіі?
5. Якія інфармацыйныя тэхналогіі актуальныя ў сучасным свеце?
6. Што такое інфармацыйныя рэсурсы?
7. Што такое інфармацыйныя паслугі?



Практыкаванні

- 1 Падрыхтуйце паведамленне на адну з тэм:
 1. Сістэмы штучнага інтэлекту.
 2. Тэхналогіі віртуальнай і дапоўненай рэальнасці.
 3. Тэхналогіі машыннага навучання.
 4. Рынак інфармацыйных паслуг у Беларусі.
- 2 Знайдзіце ў Інтэрнэце значэнні наступных тэрмінаў: «высокія тэхналогіі», «біяметрыя», «ІТ-кансалтынг», «метаданыя», «чат-бот», «Data Science», «Інтэрнэт рэчаў», «разумны горад», «квантавая сувязь», «аўтсорсінг і інсорсінг», «Мовазнаўчая беларусістыка», «інфармацыйнае абслугоўванне», «блокчэйн». Вызначыце, да якога з паняццяў — «інфармацыйная сістэма», «інфармацыйная тэхналогія», «інфармацыйны рэсурс», «інфармацыйная паслуга» — іх можна залічыць.
- 3 Азнаёмцеся з інфармацыйнымі рэсурсамі ўніверсітэтаў Беларусі.

§ 13. Інфарматызацыя грамадства

Прыклад 13.1. Характарыстыкі інфармацыйнага грамадства:

- стварэнне навукаёмістых вытворчасцей і тэхналогій з высокай інтэнсіфікацыяй камп'ютарнага апаратнага, праграмнага і інфармацыйнага забеспячэння;
- інтэнсіўнасць інфармацыйных працэсаў;
- шырокае выкарыстанне тэлекамунікацый;
- станоўчая дынаміка развіцця эканомікі;
- поспехі ў галіне экалагічнай аховы чалавека.

Адна з мэт развіцця інфармацыйнага грамадства — стварэнне шырокіх магчымасцей для свабоднага развіцця асобы і грамадства.

Асновай **інфармацыйнага грамадства** з'яўляецца шырокае выкарыстанне інфармацыйных тэхналогій ва ўсіх сферах чалавечай дзейнасці (прыклад 13.1).

Інфармацыйнае грамадства адраўняецца ад традыцыйнага індустрыяльнага грамадства высокім развіццём інфарматыкі, сувязі і мікраэлектроннай прамысловасці, з'яўляецца крыніцай новых рабочых месцаў, якія дамінуюць у эканамічным развіцці.

Пераход ад індустрыяльнага грамадства да інфармацыйнага супраджаецца інфарматызацыяй.

Пад інфарматызацыяй разумеецца арганізацыйны сацыяльна-экана-

мічны і навукова-тэхнічны працэс стварэння аптымальных умоў для задавальнення інфармацыйных патрэбнасцей грамадства на аснове фарміравання і выкарыстання інфармацыйных рэсурсаў.

Працэс інфарматызацыі грунтуецца на дасягненнях такіх навук, як кібернетыка, інфарматыка, мікраэлектроніка, тэорыя сістэм і інш.

У інфармацыйным грамадстве вялізнае значэнне набывае фарміраванне інфармацыйнай культуры сучаснага чалавека. Інфармацыйная культура з'яўляецца часткай агульначалавечай культуры як сукупнасці матэрыяльных і духоўных каштоўнасцей, якія створаны і ствараюцца чалавецтвам у працэсе грамадска-гістарычнага развіцця.

Пад інфармацыйнай культурай разумеецца сукупнасць ведаў і ўменняў чалавека, пададзеных у выглядзе сістэмы правілаў яго паводзін у інфармацыйным грамадстве.

Сучаснаму чалавеку неабходна валодаць тэарэтычнымі і практычнымі асновамі інфарматыкі, камп'ютарнымі ўстройствамі, разумець і выкарыстоўваць інфармацыйныя тэхналогіі ў розных сферах чалавечай дзейнасці: навукавай, вытворчай, сацыяльнай і інш.

Працэс фарміравання інфармацыйнай культуры чалавека ўключае ў сябе розныя кампаненты, такія як метадалагічны, тэхналагічны, камунікатыўны, моўны, алгарытмічны і інш. (прыклад 13.2).

Інфармацыйная культура мае свой этычны і эстэтычны змест і выяўляецца ў правілах, нормах паводзін

Тэрмін «інфарматызацыя» (англ. *informatization*) набыў шырокае распаўсюджванне толькі ў Расіі і Кітаі. Гэта было звязана, па-першае, з недастатковай распрацаванасцю ў 1980—1990-х гг. гласарыя па тэматыцы «інфармацыйныя тэхналогіі» і «інфармацыйнае грамадства», па-другое, з некаторымі спецыфічнымі асаблівасцямі развіцця інфармацыйна-камунікацыйных тэхналогій у гэтых краінах. Яны характарызаваліся высокім узроўнем развіцця прыкладных, спецыялізаваных апаратна-праграмных комплексаў і вельмі слабай тэлекамунікацыйнай інфраструктурай, якая станавілася тормазам гарманічнага развіцця інфармацыйнага грамадства.

Прыклад 13.2. Кампаненты інфармацыйнай культуры.

Метадалагічны кампанент дапускае авалоданне чалавекам ведамі пра ролю і значэнне камп'ютарнай тэхнікі, камп'ютарных інфармацыйных тэхналогій і камунікацый у жыцці грамадства; разуменне ім эканамічных, сацыяльных і псіхалагічных праблем, якія ўзнікаюць у інфармацыйным грамадстве і інш.

Тэхналагічны кампанент выяўляецца ва ўменні чалавекам выкарыстоўваць сучасныя інфармацыйныя тэхналогіі ў сваёй дзейнасці.

Камунікатыўны кампанент уключае ў сябе веды ў галіне выкарыстання камп'ютарных сетак рознага прызначэння, іх тэхнічнага і праграмнага забеспячэння, авалодання чалавекам метадамі эфектыўнага пошуку інфармацыі ў сетцы Інтэрнэт і інш.

Моўны і алгарытмічны кампаненты прадугледжваюць веданне і выкарыстанне чалавекам адной або некалькіх моў праграмавання, асноўных паняццяў тэорыі алгарытмаў і г. д.

Прыклад 13.3. Асноўныя рысы інфармацыйнай цывілізацыі¹.

1. Інфармацыйная эканоміка

Інфармацыйная сфера на сённяшні дзень з'яўляецца адной з самых эфектыўных сфер укладання капіталу. Агульны аб'ём сусветнага рынку інфармацыйных тэхналогій ацэньваецца велічынёй каля 4 трыльёнаў долараў² (па стане на 2020 г.), і гэты аб'ём увесь час расце.

2. Глобальная лічбавізацыя

Адной з асноўных тэндэнцый развіцця сучаснай тэхнікі з'яўляецца шырокае выкарыстанне даных у лічбавай форме. Апрацоўка вялікіх аб'ёмаў даных і выкарыстанне вынікаў іх аналізу дазваляць істотна павялічыць эфектыўнасць розных відаў вытворчасці, тэхналогій, абсталявання, захоўвання, продажу, дастаўкі тавараў і паслуг у параўнанні з традыцыйнымі формамі гаспадарання.

3. Развіццё інтэлектуальных камп'ютарных сістэм

Шырокае распаўсюджванне атрымліваюць шэраг камп'ютарных сістэм, экспертных сістэм з інтэлектуальным інтэрфейсам, саманавучальныя сістэм. Узрастае роля інтэлектуальных баз даных і машыннага навучання.

4. Віртуалізацыя эканомікі, палітыкі і культуры

З'яўленне рынку каштоўных папер прывяло да таго, што асноўныя здзелкі сталі ажыццяўляцца не з рэальнымі прадметамі, а з акцыямі, якія замяняюць гэтыя прадметы (г. зн. па сутнасці — з інфармацыйнымі мадэлямі прадметаў). Замена рэальных рэчаў іх інфармацыйнымі мадэлямі ў інфармацыйнай цывілізацыі становіцца ўсеагульнай з'явай.

чалавека ў інфармацыйным грамадстве, узорах дзейнасці, каштоўнасцямі арыенцірамі і інш.

Працэс пабудовы інфармацыйнага грамадства ў кожнай краіне працякае па-рознаму. У розных краінах прымаюцца розныя законы пра рэгуляванне інфармацыйнай дзейнасці чалавека, развіццё індустрыі інфармацыйных тэхналогій (ІТ-індустрыі).

У Беларусі ІТ-індустрыя — галіна, якая развіваецца вельмі дынамічна і з кожным годам ператвараецца ва ўсё больш магутны сектар эканомікі краіны, сферу стратэгічнага значэння. У сакавіку 2018 г. пачаў дзейнічаць падпісаны Прэзідэнтам Рэспублікі Беларусь дакумент, што вызначае асноўныя напрамкі стварэння лічбавай эканомікі XXI ст. і развіцця Беларусі як ІТ-краіны — дэкрэт «Аб развіцці лічбавай эканомікі». Дэкрэт адкрывае новыя магчымасці для развіцця інавацый у Беларусі, дасягненняў у самых перадавых сферах ад штучнага інтэлекту да віртуальнай рэальнасці³.

Працэсы інфарматызацыі, якія праходзяць у розных краінах, у канчатковым выніку прывядуць да стварэння інфармацыйнай цывілізацыі (прыклад 13.3).

Інфармацыйная цывілізацыя — простая «наследніца» індустрыяльнай цывілізацыі, якая змяніла аграрную цывілізацыю.

¹ Колин К. К. Информационная цивилизация. — Институт проблем информатики РАН Москва, 2002. — 112 с.

² [http://www.tadviser.ru/index.php/Статья:ИТ_\(мировой_рынок\)](http://www.tadviser.ru/index.php/Статья:ИТ_(мировой_рынок))

³ http://president.gov.by/ru/official_documents_ru/view/dekret-8-ot-21-dekabrya-2017-g-17716/

Мяркуецца, што паўнаватрасная інфармацыйная цывілізацыя можа быць толькі глабальнай, якая ахоплівае ўсё чалавецтва. Яна павінна грунтавацца на маральнай аснове ўзаемнай згоды, уступак і кампрамісаў пры захаванні нацыянальна-культурных традыцый, без навязвання чых бы там ні было догмаў і светапоглядаў, без войнаў і гвалту.

Лічбавізацыя і віртуалізацыя грамадства вядзе да некаторых негатыўных наступстваў. У віртуальнай палітыцы важныя не дзелавыя якасці кандыдата, а яго «імідж» (інфармацыйная мадэль). У мастацтве таленту мастака, пісьменніка ці выканаўцы, як правіла, аказваецца недастаткова. Патрэбна значная «раскрутка», што патрабуе ўжывання ўсё тых жа тэхналогій.



1. Што такое інфармацыйнае грамадства?
2. Якія рысы інфармацыйнага грамадства вы можаце назваць?
3. Што такое інфарматызацыя?
4. Што разумеюць пад інфармацыйнай культурай?
5. Якія кампаненты ўваходзяць у інфармацыйную культуру?
6. Што такое інфармацыйная цывілізацыя?



Практыкаванні

Падрыхтуйце паведамленне на адну з тэм:

1. Інфармацыйныя рэвалюцыі ў жыцці грамадства.
2. Інфарматызацыя ў Беларусі.
3. Перспектывы развіцця інфармацыйнай цывілізацыі.
4. Магчымыя небяспекі інфармацыйнай цывілізацыі.

§ 14. Адукацыя і прафесійная дзейнасць у інфармацыйным грамадстве

14.1. Адукацыя ў інфармацыйным грамадстве

У сучасным свеце ключавое значэнне маюць веды. Для абазначэння гэтага феномена выкарыстоўваецца тэрмін «грамадства ведаў» (прыклад 14.1).

У «грамадстве ведаў» важней за ўсё «навучыцца вучыцца», а інфармацыйныя тэхналогіі спрыяюць пастаяннаму абнаўленню асабістай і прафесійнай кампетэнтнасці. Новыя тэхналогіі паўсюдна паскараюць стварэнне і распаўсюджванне ведаў. Навучанне становіцца

Прыклад 14.1. Характэрныя рысы грамадства ведаў:

- усведамленне ролі ведаў як умовы поспеху ў кожнай сферы дзейнасці;
- наяўнасць пастаяннай патрэбнасці чалавека ў новых ведах, неабходных для рашэння новых задач, стварэння новых відаў прадукцыі і паслуг;
- эфектыўнае функцыянаванне сістэм вытворчасці і перадачы ведаў;
- узаемнае стымуляванне прапанавання ведаў і попыту на веды (прапанаванне імкнецца задавальняць наяўны попыт на веды і фарміраваць попыт).

Прыклад 14.2. Мэты інфарматызацыі сістэмы адукацыі Рэспублікі Беларусь (паводле канцэпцыі інфарматызацыі):

- стварэнне для насельніцтва роўных магчымасцей атрымання якасных адукацыйных паслуг на ўзроўні сучасных патрабаванняў нацыянальных і міжнародных стандартаў па-за залежнасцю ад месца жыцця і навучання з выкарыстаннем сучасных ІКТ;

- фарміраванне асобы, адаптаванай да жыцця ў інфармацыйным грамадстве з усімі яго магчымасцямі, пагрозамі, выклікамі і рызыкамі.

Прыклад 14.3. Асноўныя напрамкі мабільнага навучання:

Мабільнае навучанне

пашырэнне магчымасцей і забеспячэнне роўнага доступу да адукацыі

персаналізацыя навучання

навучанне ў любы час і ў любым месцы

імгненная зваротная сувязь і ацэнка вынікаў навучання

эфектыўнае выкарыстанне часу на ўроках і ў класах

фарміраванне новых згуртаванняў навучэнцаў

падтрымка сітуацыйнага навучання

дапамога навучэнцам з абмежаванымі магчымасцямі

Акрамя традыцыйнай вочнай і за-вочнай формы навучання некаторыя беларускія ўніверсітэты прапануюць вучыцца дыстанцыйна¹.

Дыстанцыйная форма адукацыі дапускае камунікаванне выкладчыка і студэнта на адлегласці. Абмен інфармацый адбываецца з дапамогай Інтэрнэту, анлайн-сэрвісаў. Навучэнец можа атрымаць кансультацыю, патрэбную літаратуру ад выкладчыка, паслухаць лекцыю аддалена.

ключавой каштоўнасцю грамадства ведаў. Паводле даследаванняў ЮНЕСКА важнае значэнне ў грамадстве набываюць здольнасць арыентавацца ў струмені інфармацыі, кагнітыўныя здольнасці, крытычны розум, што дазваляе адрозніваць карысную інфармацыю ад бескарыснай.

Для грамадства ведаў важным з'яўляецца ўзаемасувязь ведаў і сістэмы адукацыі, прытым адменную ролю набываюць працэсы інфарматызацыі ў сістэме адукацыі (прыклад 14.2).

Для сістэмы адукацыі актуальным становіцца лозунг: «Сучасны навучэнец — мабільны навучэнец!». Навучэнец: школьнік, гімназіст, ліцэіст, студэнт — павінен мець пастаянны доступ да электронных адукацыйных рэсурсаў і паслуг, у тым ліку ва ўстанове адукацыі, дома, у дарозе. Гэта датычыцца і іншых удзельнікаў адукацыйнага працэсу: бацькоў, педагогічных работнікаў, кіраўнікоў сістэмы адукацыі розных узроўняў. Мабільнасць кожнага ўдзельніка адукацыйнага працэсу будзе ляжаць у аснове мабільнай адукацыі ў новым інфармацыйным грамадстве (прыклад 14.3).

Забяспечыць актуальнасць і мабільнасць адукацыйных рэсурсаў дазволіць ужыванне «воблачных» тэхналогій, распрацоўка і ўкараненне электронных адукацыйных рэсурсаў.

Адной з форм электроннага навучання з'яўляецца дыстанцыйнае навучанне. Яно дае магчымасць вучыцца незалежна ад месца работы і жыцця, для яго характэрныя гнуткасць і эка-

¹ Адукар. <https://adukar.by/news/distancionnoe-obuchenie>

намічнасць. Да перспектыўных на-прамкаў дыстанцыйнага навучання можна залічыць дадатковую адукацыю дарослых (павышэнне кваліфікацыі і перарыхтоўку, вучэбныя курсы, падрыхтоўку да паступлення ва ўстановы адукацыі і інш.), дадатковую адукацыю дзяцей і моладзі, спецыяльную адукацыю.

14.2. Прафесійная дзейнасць у інфармацыйным грамадстве

У інфармацыйным грамадстве галоўным рэсурсам з'яўляецца інфармацыя. Менавіта на аснове валодання інфармацыяй пра самыя розныя працэсы і з'явы можна эфектыўна і аптымальна будаваць кожную дзейнасць. Пры гэтым павялічваецца не толькі якасць спажывання, але і якасць вытворчасці: чалавек, які выкарыстоўвае інфармацыйныя тэхналогіі, мае лепшыя ўмовы працы, праца становіцца інтэлектуальнай, творчай, крэатыўнай.

Асноўным крытэрыем развітасці інфармацыйнага грамадства можна лічыць колькасць насельніцтва, занятага ў інфармацыйнай сферы, а таксама насельніцтва, якое выкарыстоўвае інфармацыйныя і камунікацыйныя тэхналогіі ў сваёй штодзённай дзейнасці.

Усё больш жыхароў Беларусі выкарыстоўваюць Інтэрнэт у прафесійнай сферы і ў штодзённым жыцці. Па падліках аналітыкаў, да студзеня 2019 г. на 9,44 млн жыхароў

Кантроль ведаў таксама праводзіцца дыстанцыйна: па відэасувязі ці з дапамогай інтэрактыўных праграм тэсціравання. Аднак сесія здаецца вочна.

Галоўныя перавагі: навучанне ў любы час, у зручным месцы і псіхалагічна камфортных абставінах. Студэнт сам выбірае прымальны тэмп заняткаў, дзякуючы чаму лёгка сумяшчаць навучанне з іншымі справамі.

Паводле даных Нацыянальнага статыстычнага камітэта Рэспублікі Беларусь, доля насельніцтва, занятага ў ІТ-сферы, павялічылася з 1,7 % (2010 г.) да 2,5 % (2018 г.). Пры гэтым у 2019 г. доля ІТ-сектара ў УВП складае 5,5 % і па прагнозах можа вырасці да 10 % да 2023 г.

Прыклад 14.4. Агляд перспектыўных прафесій:

- Прагноз ад Адукара¹. Запатрабаванымі застаюцца спецыялісты па інфармацыйнай бяспецы, ацэнцы інтэлектуальнай уласнасці, рэкламшчыкі, маркетологі, лагісты. У будучым спатрэбяцца спецыялісты ў галіне штучнага інтэлекту, біяінжынеры, урбаністы, будаўнікі разумных гарадоў і дарог.

- Запатрабаваныя прафесіі 2020². Сярод агульнага спіса найбольш патрэбных прафесій 2015—2020 гг. называюць: ІТ-спецыялістаў (праграмісты, спецыялісты па адміністраванні баз даных ці сервернага абсталявання, вэб-дызайнеры), інжынераў і спецыялістаў у сферы маркетынгу, медыкаў, педагогаў, перакладчыкаў.

¹ <https://adukar.by/news/kakie-professii-budut-vostrebovany-v-budushchem>

² <http://www.proprof.ru/stati/careera/vybor-professii/statistika-i-reytingi/vostrebovannye-professii-2020-spisok>

Прыклад 14.4. Працяг.

• Самыя запатрабаваныя прафесіі будучага дзесяцігоддзя¹. Такімі прафесіямі сталі ІТ-спецыялісты, інжынеры, маркетологі, медыкі, спецыялісты ў галіне сэрвісу, спецыялісты ў галіне нанатэхналогій.

Адной з самых незвычайных прафесій будучыні называюць спецыяліста па захоўванні памяці. Мяркуецца, што асноўныя нейрабіёлагі пусцяць у масы рэвалюцыйныя мазгавыя імпланты, якія дазваляць чалавеку выкарыстоўваць мозг як флэшку. Спецыялісты будучы дапамагаць людзям рацыянальна працаваць са сваёй памяццю, не перагружаючы мозг².

У інфармацыйным грамадстве трэба чакаць запатрабаванасці прафесій інфармацыйнай арыентацыі:

• інжынеры ведаў — спецыялісты, якія свабодна арыентуюцца ў аўтаматызаваных сістэмах фарміравання, захоўвання і выкарыстання інфармацыйных рэсурсаў;

• сістэмныя аналітыкі, якія валодаюць вопытам камп'ютарнага мадэлявання і прагназавання ў розных галінах сацыяльнай практыкі;

• інфармацыйныя менеджары, якія прапануюць свае паслугі ў забеспячэнні інфармацыйнай розных напрамкаў дзейнасці.

Беларусі прыпадала 11,87 млн абанентаў мабільнай сувязі — гэта прыклад на 126 % насельніцтва. Паказчык павялічыўся на 3,1 % (357 тыс.) у параўнанні са статыстыкай на студзень 2018 г.³

У цяперашні час дастаткова шмат інтэрнэт-рэсурсаў прапануюць агляды прафесій, якія будуць актуальнымі ў найбліжэйшае дзесяцігоддзе (прыклад 14.4).

Ва ўсіх спісах на першым месцы — спецыялісты ІТ-сферы. Да прафесій гэтай сферы належаць: праграмісты, спецыялісты па адміністраванні баз даных ці сервернага абсталявання, вэб-дызайнеры, спецыялісты па інфармацыйнай бяспецы, спецыялісты па ацэнцы інтэлектуальнай уласнасці і інш.

У спісах таксама фігуруюць спецыяльнасці інжынера, медыка, хіміка, лагіста, нанатэхнолага — тыя спецыяльнасці, якія патрабуюць трывалых ведаў у галіне прыродазнаўчых навук і матэматыкі.



1. Якія характэрныя рысы грамадства ведаў вы можаце назваць?
2. Што разумеюць пад мабільным навучаннем?
3. У чым перавагі дыстанцыйнага навучання?
4. Што лічаць асноўным крытэрыем развітасці інфармацыйнага грамадства?
5. Якія прафесіі будучы запатрабаваны ў найбліжэйшым будучым?

**Практыкаванні**

Падрыхтуйце паведамленне на адну з тэм:

1. Дыстанцыйнае навучанне ў Беларусі.
2. Незвычайныя прафесіі будучага.

¹ <https://benefit.by/page/show/articles/2285>

² <https://gagadget.com/science/23845-10-neobyichnyih-professij-buduschego-kotoryie-rouavyatsya-uzhe-zavtra/>

³ <https://dev.by/news/digital-2019-belarus>

§ 15. Кібербяспека і кіберустойлівасць

15.1. Кібербяспека

Тэрмін «кібербяспека» аб'ядноўвае ў сабе словы «кібер» і «бяспека». Пачнём з бяспекі.

Бяспека — гэта стан ахаванасці збалансаваных інтарэсаў асобы, грамадства і дзяржавы ад знешніх і ўнутраных пагроз.

Пагроза бяспекі — гэта магчымасць уплыву на сістэму ці аб'ект, які можа парушыць бяспеку.

У залежнасці ад сфер інтарэсаў адрозніваюць некалькі відаў бяспекі. У сферы эканомікі — эканамічная бяспека, у сферы экалогіі — экалагічная бяспека і г. д.

Ключавое значэнне ў сучасным свеце набыла інфармацыйная сфера, якая аказвае ўсёабдымны ўплыў на эканамічныя, палітычныя і сацыяльныя працэсы.

Інфармацыйная бяспека — гэта стан ахаванасці збалансаваных інтарэсаў асобы, грамадства і дзяржавы ад знешніх і ўнутраных пагроз у інфармацыйнай сферы.

Паняцце інфармацыйнай бяспекі ўведзена ў нашай краіне на заканадаўчым узроўні (прыклад 15.1).

Асноўнымі аб'ектамі інфармацыйнай сферы з'яўляюцца *інфармацыйныя сістэмы*, якія ўключаюць банкі даных, інфармацыйныя тэхналогіі і комплекс праграмна-тэхнічных сродкаў (прыклад 15.2).

Прыклад 15.1. Тэрмін «інфармацыйная бяспека» ўведзены ў Канцэпцыі інфармацыйнай бяспекі Рэспублікі Беларусь, зацверджанай Саветам Бяспекі Рэспублікі Беларусь у 2019 г.

У Канцэпцыі мэтай забеспячэння інфармацыйнай бяспекі пастаўлена дасягненне і падтрыманне такога ўзроўню ахаванасці інфармацыйнай сферы, які забяспечвае рэалізацыю нацыянальных інтарэсаў Рэспублікі Беларусь і яе прагрэсіўнае развіццё.

Увага да інфармацыйнай бяспекі на дзяржаўным узроўні выклікана тым, што аб'екты прамысловасці, транспарту, энергетыкі, электрасувязі, аховы здароўя пабудаваны на базе камп'ютарных сістэм. Гэта ставіць жыццё і здароўе насельніцтва ў простую залежнасць ад надзейнасці гэтых камп'ютарных сістэм і ахаванасці інфармацыі ў іх.

Прыклад 15.2. Тэрмін «інфармацыйная сістэма» ўведзены ў Законе Рэспублікі Беларусь «Аб інфармацыі, інфарматызацыі і ахове інфармацыі», прынятым у 2008 г.

Інфармацыйныя сістэмы прызначаны для захоўвання, апрацоўкі, пошуку, распаўсюджвання, перадачы і ўяўлення інфармацыі. Па ступені размеркаванасці яны падзяляюцца на лакальныя (настольныя) і размеркаваныя.

Прыклады лакальных інфармацыйных сістэм — гэта пакеты офісных праграм (MS Office, OpenOffice і інш.), сістэмы праграмавання, сістэмы кіравання базами даных (СКБД).

У размеркаваных інфармацыйных сістэмах кампаненты размеркаваны па некалькіх камп'ютарах. Прыкладамі могуць быць інфармацыйныя сістэмы прадпрыемстваў, устаноў і арганізацый, аўтаматызаваныя сістэмы кіравання вытворчымі працэсамі, базы даных і інфармацыйна-пошукавыя сістэмы ў Інтэрнэце.

Прыклад 15.3. Слова «кібер» узнікла ў англійскай мове як частка слова «кібернетыка», якое з’яўляецца назвай навукі пра агульныя законы кіравання, сувязі і перапрацоўкі інфармацыі.

Слова «кібер» спачатку абазначала слова «кібернетычны» і ўпершыню стала выкарыстоўвацца пісьменнікамі-фантастамі. Менавіта ў фантастыцы ўзнікла слова «кіберпрастора».

У рускамоўнай фантастыцы кіберамі называлі разнавіднасць робатаў.

Прыклад 15.4. Кіберпрастора — панятка для абазначэння віртуальнай рэальнасці.

Кіберспорт (камп’ютарны спорт) — камандныя ці індывідуальныя спаборніцтвы па камп’ютарных відэагульнях, якія імітуюць рэальныя спартыўныя спаборніцтвы.

Кібербулінг — гэта ананімныя наўмысныя абразы, пагрозы, нападкі на чалавека з дапамогай Інтэрнэту.

Кібертэразызм — гэта ўздзеяння на інфармацыйныя сістэмы, якія нясуць пагрозу здароўю і жыццю людзей, ужываюцца для запалохвання насельніцтва або для дэстабілізацыі грамадскага парадку.

Прыклад 15.5. *Цэласнасць інфармацыі* азначае, што змяненне інфармацыі можа быць толькі кантралюемым.

Канфідэнцыяльнасць інфармацыі азначае, што асоба, якая атрымала да яе доступ, не мае права перадаваць яе іншым асобам.

Сапраўднасць інфармацыі азначае, што дакладна вызначана аўтарства інфармацыі.

Даступнасць інфармацыі азначае, што асоба, якая мае доступ да інфармацыі, рэалізуе гэты доступ бесперашкодна.

Захаванасць інфармацыі азначае, што пры любых акалічнасцях інфармацыя не будзе беззваротна страчана.

Тэхнічныя сродкі, сістэмы і тэхналогіі захоўвання, перадачы і апрацоўкі інфармацыі ўтвараюць *інфармацыйную інфраструктуру* прадпрыемства, арганізацыі ці дзяржавы.

Кібербяспека — гэта стан ахованасці інфармацыйнай інфраструктуры і змешчанай у ёй інфармацыі ад знешніх і ўнутраных пагроз.

Такім чынам, кібербяспека — гэта раздзел інфармацыйнай бяспекі.

«Кібер» — прыстаўка, якая азначае «звязаны з камп’ютарамі і Інтэрнэтам» і мае цікавую гісторыю (прыклад 15.3). Вядома нямала тэрмінаў, якія выкарыстоўваюць гэтыя слова-прыстаўкі (прыклад 15.4).

Кібербяспека прадугледжвае ахованасць інфармацыі ў інфармацыйнай інфраструктуры. У адпаведнасці з Законам Рэспублікі Беларусь «Аб інфармацыі, інфарматызацыі і ахове інфармацыі» ахова інфармацыі павінна забяспечваць *цэласнасць, канфідэнцыяльнасць, сапраўднасць, даступнасць і захаванасць інфармацыі*. У прыкладзе 15.5 прыведзены апісанні гэтых яе ўласцівасцей.

Ахова інфармацыі — гэта не простая задача, бо колькасць даных з часам расце лавінападобна. Толькі з 2010 па 2020 год аб’ём інфармацыі, якая захоўваецца ва ўсім свеце, вырас у 50 разоў. Лік сервераў амерыканскіх кампаній Google і Amazon вылічваецца мільёнамі.

15.2. Уразлівасць інфармацыйнай інфраструктуры

Кібербяспека прадугледжвае ахаванасць інфармацыйнай інфраструктуры і змешчана ў ёй інфармацыі ад знешніх і ўнутраных пагроз.

Пагрозы ўзнікаюць, калі інфармацыйная інфраструктура мае ўразлівасць.

Уразлівасць — уласцівасць інфармацыйнай інфраструктуры ці яе аб'ектаў, якая дазваляе рэалізаваць пагрозу (уплываць на інфраструктуру).

Уплыў на інфармацыйную інфраструктуру ці яе аб'екты можа быць *выпадковым* або *наўмысным* (прыклад 15.6).

Наўмысны ўплыў на інфармацыйную інфраструктуру праграмнымі ці праграма-тэхнічнымі сродкамі называецца *кібератакай*.

Асноўная мэта наўмысных уплываў — атрыманне фінансавай выгады. Для гэтага выкарыстоўваюць як вымаганне, гэтак і крадзеж каштоўнай інфармацыі (персанальныя даныя акаўнтаў, даныя доступу да плацежных сэрвісаў).

Уразлівасць могуць мець інфармацыйная інфраструктура ў цэлым, яе тэхнічныя і праграмныя сродкі, персанал і карыстальнікі.

Калі антывірусныя сродкі інфраструктуры маюць уразлівасць, то вірусная кібератака можа дасягнуць поспеху (прыклад 15.7).

Прыклад 15.6. Выпадковы ўплыў можа з'яўляцца вынікам памылковых дзеянняў персаналу, раптоўных аднаўленняў тэхнічных сродкаў і г. д.

Наўмысныя ўплывы дзеляцца на актыўныя і пасіўныя.

Пасіўныя ўплывы накіраваны на несанкцыянаванае выкарыстанне інфармацыйных рэсурсаў, напрыклад падслухоўванне, відэазапіс.

Актыўнымі наўмыснымі ўплывамі з'яўляюцца ўкараненне шкодных праграм у камп'ютары карыстальнікаў, ашуканскае выкарыстанне сайтаў інфармацыйных паслуг і інш.

Самы прасты і пашыраны спосаб наўмыснага ўплыву — гэта фішынг (згадваўся ў дапаможніку для 9-га класа). Уяўляе сабой рассыланне электронных пільмаў ці паведамленняў у сацсетках ад імя калег па рабоце, супрацоўнікаў банка ці прадстаўнікоў дзяржаўнай установы. Мэта фішынгавага пільма — прымусяць атрымальніка неадкладна перайсці на падраблены вэбсайт, вельмі падобны да вядомага, і там увесці прыватную інфармацыю, каб пазбегнуць нібыта раптам узнікшай небяспекі ці нейкіх сур'ёзных наступстваў бяздзейнасці. Наўмысныя ўплывы, якія рэалізуюць пагрозы кібербяспекі, у адпаведнасці з Крымінальным кодэксам Рэспублікі Беларусь могуць быць кваліфікаваныя як злачынства супраць інфармацыйнай бяспекі.

Прыклад 15.7. Вірусы заражаюць файлы спецыфічным кодам. Траянцы — шкодныя праграмы, якія хаваюцца пад маскай легальнага праграмага сродку. Шпіёнскія праграмы таемна сочаць за дзеяннямі карыстальніка і збіраюць фінансавую персанальную інфармацыю. Праграмы-вымагальнікі шыфруюць файлы і даныя, а злачынцы патрабуюць выкуп за аднаўленне.

Прыклад 15.8. Уразлівасць праграмнага сродку можа быць вынікам памылак праграмавання, але часцей гэта непрадугледжаная ўласцівасць, якую распрацоўшчыку выявіць вельмі складана. Праверкай працаздольнасці праграм займаюцца тэсціроўшчыкі, а пошукам уразлівасці — злачынцы.

Прыклад 15.9. У дадатках пакета MS Office ўразлівасць знаходзяць увесь час, і распрацоўшчыкі заўсёды ствараюць і рассылаюць абнаўленні, якія закрываюць знойдзеную ўразлівасць.

Гэта супрацьстаянне злачынцаў і распрацоўшчыкаў вядзецца ўжо даўно.

Са з'яўленнем новых інфармацыйных тэхналогій і новых праграмных сродкаў можна чакаць з'яўлення новых уразлівасцей і новых пагроз.

Прыклад 15.10. У жніўні 2013 г. былі ўзламаны акаўнты з персанальнымі данымі больш як 3 млрд карыстальнікаў кампаніі Yahoo!.

У 2018 г. аказаліся ўзламанымі акаўнты больш як 500 млн кліентаў гасцінчнай сеткі Marriott, 440 млн карыстальнікаў праграмнага забеспячэння Veeam, 300 млн кліентаў лагістычнай кампаніі SF Express.

У жніўні 2020 г. у Інтэрнэце з'явілася неахаваемая база даных, якая змяшчае персанальныя даныя каля 235 млн карыстальнікаў сэрвісаў Instagram, TikTok і YouTube.

Прыклад 15.11. Поспех размеркаваных сеткавых атак заснаваны на ўразлівасці сервераў камп'ютарнай сеткі, якія заўсёды маюць абмежаванні па колькасці адначасова апрацоўваемых запытаў. Калі колькасць запытаў перавышае магчымасці сервера, то вынікам можа быць поўнае спыненне яго нармальнай работы — адмова ў абслугоўванні.

Вялікай праблемай з'яўляецца ўразлівасць аперацыйных сістэм і дадаткаў (прыклад 15.8). Як толькі злачынцы знаходзяць слабае месца праграмы, яны ствараюць свой шкодны праграмны код, які называецца «эксплойт», і, атакуючы, спрабуюць укараніць яго ў праграму (прыклад 15.9).

Па ацэнках экспертаў, кампаніі ва ўсім свеце падпадаюць кібератакам эксплойтаў-вымагальнікаў кожныя 14 с. Маштабы крадзяжу персанальных даных карыстальнікаў з сервераў сеткі ўражваюць яшчэ больш (прыклад 15.10).

Іншы від кібератак — *размеркаваныя сеткавыя атакі*. Іх называюць таксама атакамі тыпу «адмова ў абслугоўванні» (па-англійску Distributed Denial of Service) ці DDoS-атакамі (прыклад 15.11).

Звычайная мэта DDoS-атакі — вымагальніцтва грошай за спыненне атакі, дыскрэдытацыя бізнес-канкурэнта ці нанясенне яму шкоды.

З кожным годам лік кібератак расце. Пры гэтым набор прыёмаў, якія выкарыстоўваюць злачынцы, папаўняецца амаль штомесячна.

Але самым уразлівым месцам кібербяспекі быў і будзе чалавечы фактар — паводзіны супрацоўнікаў і карыстальнікаў.

Паводле статыстыкі 91 % атак на банкі здзяйсняюць карумпаваныя супрацоўнікі саміх банкаў, 8 % — банкаўскія пасрэднікі і толькі 1 % прыпадае на хакераў.

Па выніках даследаванняў, страты расійскага карыстальніка ў ходзе кібератак складаюць у сярэднім 80 долараў, а кожная дзявятая ахвяра губляе больш за 1000 долараў (прыклад 15.12).

Для атрымання прыватнай інфармацыі злачынцы шырока выкарыстоўваюць сацыяльную інжынерную. Гэта псіхалагічнае маніпуляванне людзьмі дзеля здзяйснення пэўных дзеянняў ці разгалашэння прыватнай інфармацыі (прыклад 15.13).

15.3. Забеспячэнне кібербяспекі. Кіберустойлівасць

Правільны падыход да кібербяспекі дапускае рэалізацыю сістэмы мер аховы інфармацыйнай інфраструктуры.

Забеспячэнне кібербяспекі — гэта сістэма мер супрацьдзеяння пагрозам інфармацыйнай бяспекі.

Трэба заўважыць, што ў Інтэрнэце паняцце «забеспячэнне кібербяспекі» часта называюць кібербяспекай (у шырокім сэнсе).

Забеспячэнне кібербяспекі — гэта ўвасабленне ўсіх мер аховы сетак, устройстваў і дадаткаў, аховы канфідэнцыяльнасці і цэласнасці даных, а таксама мер захавання нармальнай работы інфармацыйнай інфраструктуры ў цэлым (прыклад 15.14).

У сістэме мер забеспячэння кібербяспекі асаблівае месца займае навучанне карыстальнікаў. Карыстальнік і яго лічбавае ўстройства — гэта папулярная мэта злачынцаў. Узламаўшы індывідуальнае лічбавае ўстройства,

Прыклад 15.12. Прычынай страты 21 % пацярпелых назвалі перадачу махлярам доступу да запісаў у плацежным сэрвісе, 19 % — увод сваіх прыватных даных на падрабленым вэб-сайце.

Прыклад 15.13. Махляр уступае ў кантакт з карыстальнікам, прыкідваючыся супрацоўнікам банка, прадаўцом ці пакупніком Інтэрнэт-сэрвісу, даверанай асобай сэрвісу і г. д. Мэта — псіхалагічна вымусіць карыстальніка паведаміць свае персанальныя даныя злачынцу ці на сайт, які злачынец называе.

Прыклад 15.14. Сістэма мер супрацьдзеяння пагрозам інфармацыйнай бяспекі ўтварае некалькі ўзроўняў аховы як унутры інфраструктуры, гэтак і на яе перыметры. На кожным узроўні рэалізуюцца пэўныя прыцыпы і сродкі кантролю.

Першую лінію аховы сетак забяспечваюць міжсеткавыя экраны, якія ажыццяўляюць маніторынг уваходнага і выходнага сеткавага трафіка. Міжсеткавы экран можа быць апаратным, праграмным ці змешаным тыпу.

Стратэгія DLP (data-loss prevention), якую рэалізуюць спецыялісты па інфармацыйнай бяспецы, дазваляе кантраляваць магчымыя шляхі збегу даных.

Ахову інфармацыі забяспечваюць сістэмы ідэнтыфікацыі па біяметрычных даных, сістэмы крыптаграфічнай аховы каналаў перадачы і носьбітаў даных, праграмныя рашэнні для кіравання ключамі шыфравання.

У апошнія гады шыфраванне даных стала нормай у рабоце з вэб-сайтамі.

Прыклад 15.15. У вучэбным дапаможніку «Інфарматыка» для 9-га класа быў прыведзены шэраг эфектыўных мер бяспекі карыстальніка ў сетцы Інтэрнэт. У дапаўненне да іх прывядзём наступныя меры:

1. Рэгулярна выконвайце рэзервовае капіраванне даных, якое дае гарантыі захаванасці ўнікальнай інфармацыі. У выпадку крадзяжу ці шыфравання такой інфармацыі вымагальніцтва не пройдзе. Аднавіць інфармацыю з рэзервовых копіяў можна ў кожны момант.

2. Абнаўляйце аперацыйную сістэму дадатку. Выкарыстоўваючы абнаўленні, вы атрымаеце свежыя выпраўленні ўразлівасці.

3. Уважліва стаўцеся да таго, што падключаеце да камп'ютара. Шкодныя праграмы могуць распаўсюджвацца праз заражаныя флэшкі, знешнія цвёрдыя дыскі і нават смартфонны.

4. Пазбягайце неахованых сетак Wi-Fi у грамадскіх месцах. Далучыўшыся да неахованай сеткі Wi-Fi, вы можаце папасці пад атаку Man-in-the-Middle («Чалавек пасярэдзіне»). Гэта атака, у ходзе якой злачынец перахоплівае даныя ў час іх перадачы. Ён як бы становіцца прамежкавым звяном у інфармацыйным ланцужку, а ахвяра пра гэта нават не падазрае і можа перадаць каштоўную інфармацыю.

5. Банкаўскія аперацыі ці пакупкі выконвайце толькі на вашым уласным устрыістве ў сетцы, якой вы давяраеце.

6. Сачыце за захаванасцю сваіх лічбавых устрыістваў (смартфона, планшэта і ноўтбука). Не дапускайце да іх выкарыстання незнаёмых.

7. Як правіла, сцерці даныя з Інтэрнэту немагчыма. Усё, што трапіла ў Інтэрнэт, застанеца там назаўжды. Адзіны спосаб пазбегнуць уцечкі інфармацыі — не дзяліцца ёй.

можна атрымаць доступ не толькі да інфармацыі ў ім, але і да прыватнай інфармацыі ў сетцы.

Меры забеспячэння кібербяспекі для карыстальнікаў часта фармулююць як меры ці правілы бяспекі карыстальніка (прыклад 15.15).

Эфектыўная сістэма мер супрацьдзеяння пагрозам інфармацыйнай бяспекі пераводзіць інфармацыйную інфраструктуру ў новы стан — у стан кіберустойлівасці.

Кіберустойлівасць — здольнасць інфармацыйнай інфраструктуры паспяхова прадухіляць рэалізацыю пагроз ці хутка аднаўляцца пасля іх рэалізацыі.

Кібератака на ўзроўні карыстальніка можа прывесці да самых розных наступстваў.

Тэхнічныя рашэнні не змогуць ахаваць ад пагроз, калі злачынец-прафесіянал завалодае вашым даверам. Шмат хто з іх сапраўды вывучае паводзіны людзей, збірае інфармацыю, вывучае метады, якімі людзі карыстаюцца, і на аснове гэтых даных здзяйсняе кіберзлачынствы. Таму галоўнае — не давярайце незнаёмым і будзьце пільнымі.

Зразумейце, што вы — прывабная мэта для злачынцаў, і кібератака можа здарыцца ў любы час, у кожным месцы, на кожным устрыістве. Вам дапаможа толькі ваша гатоўнасць адбіць гэту атаку.



1. Што такое бяспека?
2. Што такое пагроза бяспецы?
3. Што такое інфармацыйная бяспека?
4. У якім заканадаўчым дакуменце ўведзена паняцце «інфармацыйная сістэма»?
5. Што такое кібербяспека?
6. Што азначае слова-прыстаўка «кібер»?
7. Як вымаўляецца слова «cyber» у англійскай мове?
8. Чаму ахова інфармацыі ў сучасным свеце з'яўляецца складанай задачай?
9. Што такое ўразлівасць інфармацыйнай інфраструктуры?
10. Што такое кібератака?
11. Што хаваецца за абрэвіятурай DDoS?
12. Што такое сацыяльная інжынерыя?
13. У чым падобныя фішынг і сацыяльная інжынерыя?
14. Што такое забеспячэнне кібербяспекі?
15. Што такое кіберустойлівасць?
16. Чаму навучанне карыстальнікаў займае асаблівае месца ў сістэме мер забеспячэння кібербяспекі?
17. Да якіх наступстваў можа прывесці кібератака на ўзроўні карыстальніка?

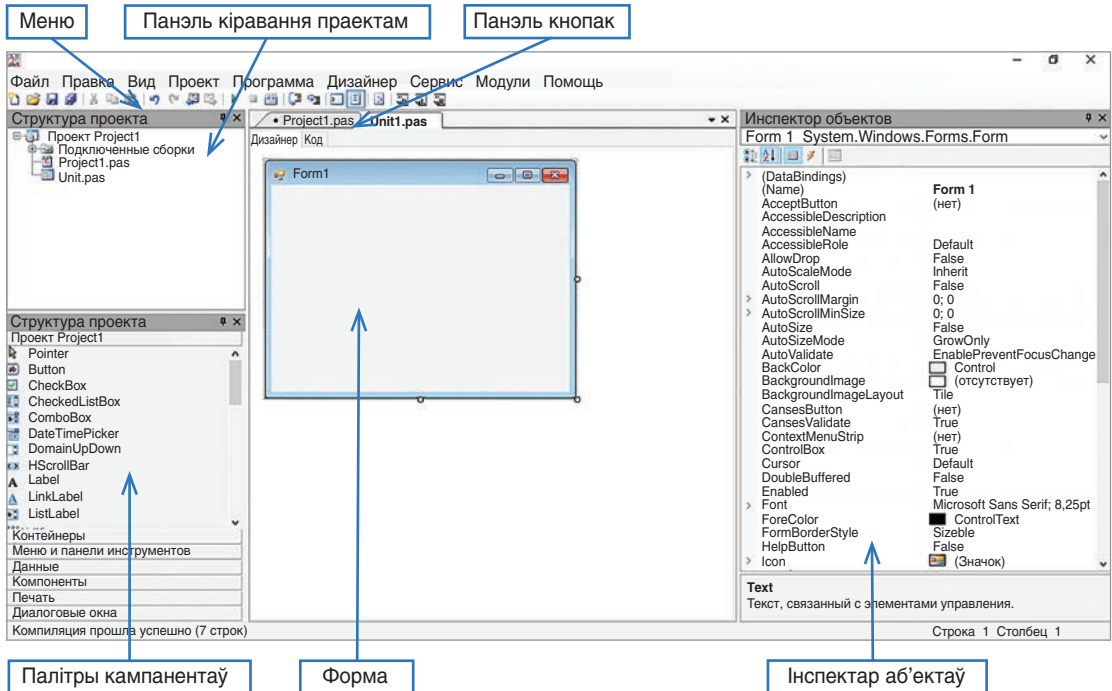


Практыкаванні

- 1 Прыведзіце прыклады наўмысных уплываў на інфармацыйныя інфраструктуры.
- 2 Прыведзіце прыклады кібератак.
- 3 Прыведзіце іншыя назвы DDoS-атак.
- 4 Назавіце магчымыя мэты DDoS-атак.
- 5 Знайдзіце ў Інтэрнэце азначэнне эксплойта.
- 6 Знайдзіце ў Інтэрнэце і прывядзіце прыклады відаў эксплойтаў.
- 7 Высветліце ў Інтэрнэце, што такое прэтэкстынг.
- 8 Высветліце ў Інтэрнэце азначэнне і паходжанне слова «хакер».

ДАДАТАК ДА ГЛАВЫ 1

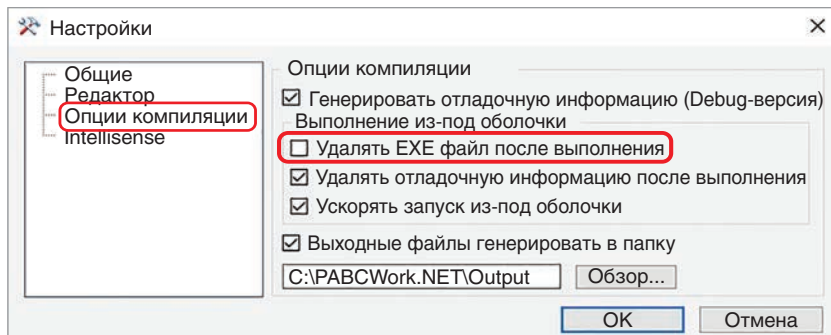
Акно асяроддзя PascalABC.Net (пры стварэнні праекта Windows Form)



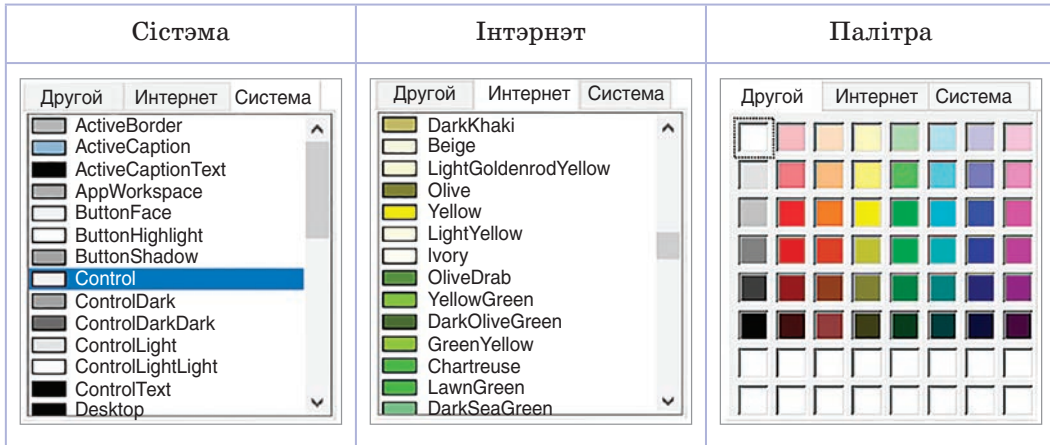
Опцыі кампілятара, пры якіх захоўваецца ехе файл

Адкрыць акно **Настройки** (каманда меню **Сервис** → **Настройки**).

У раздзеле **Опции компиляции** зняць птушку з опцыі **Удалять EXE файл после выполнения**.



Колеравыя канстанты



Работа з графікай

Уласцівасць **Pen** (аловак) дазваляе вызначыць колер, таўшчыню ліній.
Стварэнне:

`Pen p_c := new Pen(Color.Black, 1);` — аловак чорнага колеру, таўшчынёй 1 піксель.

Уласцівасць **Brush** (пэндзаль) прызначана для запаўнення абласцей зададзеным шаблонам. Стварэнне: `SolidBrush sb := new SolidBrush(Color.Blue);` — пэндзаль сіняга колеру.

Метады **Graphics** дазваляюць будаваць графічныя відарысы. Шмат у чым метады падобныя да аналагічных каманд пабудовы графічных відарысаў у асяроддзі PascalABC.Net.

| Кароткае апісанне некаторых метадаў graphics | |
|--|---|
| <code>DrawArc(Pen, Rectangle, Single, Single)</code> | Малюе дугу, якая з'яўляецца часткай эліпса, зададзенага структурай <code>Rectangle</code> і двума радыяльнымі лініямі |
| <code>DrawEllipse(Pen, Rectangle)</code> | Малюе эліпс |
| <code>DrawLine(Pen, Point, Point)</code> | Праводзіць лінію, якая злучае дзве структуры <code>Point</code> |

| Кароткае апісанне некаторых метадаў graphics | |
|--|--|
| DrawPie(Pen, Rectangle, Single, Single) | Малюе сектар, які вызначаецца эліпсам, зададзеным структурай Rectangle і дзвюма радыяльнымі лініямі |
| DrawRectangle(Pen, Rectangle) | Малюе прамавугольнік, які вызначаецца структурай Rectangle |
| DrawString(String, Font, Brush, PointF) | Стварае неабходны тэкставы радок у зададзеным месцы з дапамогай вызначаемых аб'ектаў Brush і Font |
| FillEllipse(Brush, Rectangle) | Запаўняе ўнутраную частку эліпса, які вызначаецца абмежавальным прамавугольнікам, зададзеным структурай Rectangle |
| FillPie(Brush, Rectangle, Single, Single) | Запаўняе ўнутраную частку сектара, які вызначаецца эліпсам, зададзеным структурай Rectangle і дзвюма радыяльнымі лініямі |
| FillPolygon(Brush, Point[]) | Запаўняе ўнутраную частку многавугольніка, які вызначаецца масівам пунктаў, зададзеных структурамі Point |
| FillRectangle(Brush, Rectangle) | Запаўняе ўнутраную частку прамавугольніка, які вызначаецца структурай Rectangle |

Структура **Rectangle** апісваецца дзвюма парамі каардынат, якія з'яўляюцца дыяганаллю прамавугольніка са старанамі, паралельнымі межам экрана.

Метады **DrawArc**, **DrawPie** і **FillPie** маюць аднолькавыя параметры. Радыяльныя лініі задаюць вугал (у градусах), які адмяраецца па гадзіннікавай стрэлцы, пачынаючы ад восі X і заканчваючы пунктам дугі.



Апрацоўшчыкі падзей дадатку «Графічны рэдактар»

```

procedure Form1.Form1_Load(sender: Object; e: EventArgs);
begin
  bm := new Bitmap(pictureBox1.Width, pictureBox1.Height);
  pictureBox1.Image := (Image) (bm);
  gr := Graphics.FromImage(pictureBox1.Image);
  gr.Clear(Color.White);
  gr := pictureBox1.CreateGraphics;
  c_f := Color.Black;
  p_c := new Pen(c_f, 1);
  c_b := Color.White;
  s_b := new SolidBrush(c_b);
end;

procedure Form1.numericUpDown1_ValueChanged(sender: Object;
e: EventArgs);
begin
  p_c.Dispose;
  w := numericUpDown1.Value;
  p_c := new Pen(c_f, (integer) (w));
end;

procedure Form1.button1_Click(sender: Object; e: EventArgs);
begin
  colorDialog1.ShowDialog();
  c_f := colorDialog1.color;
  w := numericUpDown1.Value;
  p_c.Dispose;
  p_c := new Pen(c_f, (integer) (w));
  panell1.BackColor := c_f;
end;

procedure Form1.button2_Click(sender: Object; e: EventArgs);
begin
  colorDialog1.ShowDialog();
  c_b := colorDialog1.color;
  s_b.Dispose;
  s_b := new SolidBrush(c_b);
  panel2.BackColor := c_b;
end;

procedure Form1.pictureBox1_MouseDown(sender: Object;
e: MouseEventArgs);
begin
  m_d := true;
  x1 := e.x; y1 := e.y;
end;

procedure Form1.pictureBox1_MouseMove(sender: Object;
e: MouseEventArgs);
begin
  if m_d then

```

```
begin
  gr.DrawLine(p_c, x1, y1, e.X, e.Y);
end;
//запомніць каардынаты для малявання наступнага адрэзка
x1 := e.X; y1 := e.Y
end;

procedure Form1.pictureBox1_MouseUp(sender: Object;
e: MouseEventArgs);
begin
  m_d := false;
end;

procedure Form1.toolStripMenuItem3_Click(sender: Object;
e: EventArgs);
begin
  gr := pictureBox1.CreateGraphics;
  gr.Clear(c_b);
end;




















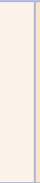




















procedure Form1.toolStripMenuItem4_Click(sender: Object;
e: EventArgs);
begin
  openFileDialog1.ShowDialog();
  F_N := openFileDialog1.FileName;
  Text := 'myPaint' + F_N;
  PictureBox1.Load(F_N);
end;





















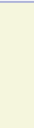





















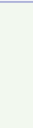



procedure Form1.toolStripMenuItem5_Click(sender: Object;
e: EventArgs);
begin
  saveFileDialog1.ShowDialog();
  F_N := saveFileDialog1.FileName;
  PictureBox1.Image.save(F_N);
  Text := 'myPaint' + F_N;
end;




procedure Form1.numericUpDown1_ValueChanged(sender: Object;
e: EventArgs);
begin
  p_c.Dispose;
  w := numericUpDown1.Value;
  p_c := new Pen(c_f, (integer)(w));
end;

procedure Form1.toolStripMenuItem7_Click(sender: Object;
e: EventArgs);
begin
  close;
end;
```

Кадзіраванне колеру

| Назва колеру | Узор | Код | Назва колеру | Узор | Код |
|--------------|---|---------|----------------|---|---------|
| black |  | #000000 | orangered |  | #ff4500 |
| dimgray |  | #696969 | lightsalmon |  | #ffa07a |
| gray |  | #808080 | sienna |  | #a0522d |
| darkgray |  | #a9a9a9 | seashell |  | #fff5ee |
| silver |  | #c0c0c0 | saddlebrown |  | #8b4513 |
| lightgrey |  | #d3d3d3 | chocolate |  | #d2691e |
| whitesmoke |  | #f5f5f5 | peachpuff |  | #ffdab9 |
| white |  | ffffff | sandybrown |  | #f4a460 |
| rosybrown |  | #bc8f8f | linen |  | #faf0e6 |
| lightcoral |  | #f08080 | bisque |  | #ffe4c4 |
| brown |  | #a52a2a | darkorange |  | #ff8c00 |
| firebrick |  | #b22222 | antiquewhite |  | #faebd7 |
| maroon |  | #800000 | tan |  | #d2b48c |
| darkred |  | #8b0000 | burlywood |  | #deb887 |
| red |  | #ff0000 | blanchedalmond |  | #ffebed |
| salmon |  | #fa8072 | papayawhip |  | #ffe5d5 |
| mistyrose |  | #ffe4e1 | oldlace |  | #fdf5e6 |
| tomato |  | #ff6347 | wheat |  | #f5deb3 |
| darksalmon |  | #e9967a | orange |  | #ffa500 |
| coral |  | #ff7f50 | floralwhite |  | #fffaf0 |

| Назва колеру | Узор | Код | Назва колеру | Узор | Код |
|----------------|---|---------|-------------------|---|---------|
| goldenrod |  | #daa520 | forestgreen |  | #228b22 |
| darkgoldenrod |  | #8860b | limegreen |  | #32cd32 |
| cornsilk |  | #fff8dc | darkgreen |  | #006400 |
| gold |  | #ffd700 | green |  | #008000 |
| lemonchiffon |  | #ffffac | lime |  | #90ee90 |
| khaki |  | #f0e68c | seagreen |  | #2e8b57 |
| palegoldenrod |  | #ee8aa | mediumseagreen |  | #3cb371 |
| darkkhaki |  | #bdb76b | mintcream |  | #f5ffa |
| ivory |  | #fffff0 | springgreen |  | #00ff7f |
| beige |  | #f5f5dc | mediumspringgreen |  | #00fa9a |
| lightyellow |  | #ffffe0 | mediumaquamarine |  | #66cdaa |
| olive |  | #808000 | aquamarine |  | #7fffd4 |
| yellow |  | #ffff00 | turquoise |  | #40e0d0 |
| olivedrab |  | #6b8e23 | lightseagreen |  | #20b2aa |
| yellowgreen |  | #9acd32 | mediumturquoise |  | #48d1cc |
| darkolivegreen |  | #556b2f | azure |  | #f0ffff |
| greenyellow |  | #adff2f | paleturquoise |  | #afeeee |
| lawngreen |  | #7cfc00 | darkslategray |  | #2f4f4f |
| chartreuse |  | #7fff00 | teal |  | #008080 |
| honeydew |  | #f0ffff | darkcyan |  | #008b8b |
| darkseagreen |  | #8fbc8f | cyan |  | #00ffff |
| lightgreen |  | #90ee90 | lightcyan |  | #e0ffff |
| palegreen |  | #98fb98 | darkturquoise |  | #00ced1 |

| Назва колеры | Узор | Код | Назва колеры | Узор | Код |
|----------------|--|---------|-----------------|--|---------|
| powderblue |  | #b0e0e6 | mediumslateblue |  | #7b68ee |
| lightblue |  | #add8e6 | mediumpurple |  | #9370db |
| deepskyblue |  | #00bfff | blueviolet |  | #8a2be2 |
| skyblue |  | #87ceeb | indigo |  | #4b0082 |
| lightskyblue |  | #87cefa | darkorchid |  | #9932cc |
| steelblue |  | #4682b4 | darkviolet |  | #9400d3 |
| aliceblue |  | #f0f8ff | mediumorchid |  | #ba55d3 |
| slategray |  | #708090 | thistle |  | #d8bfd8 |
| lightslategray |  | #778899 | plum |  | #dda0dd |
| dodgerblue |  | #1e90ff | violet |  | #ee82ee |
| lightsteelblue |  | #b0c4de | purple |  | #800080 |
| cornflowerblue |  | #6495ed | darkmagenta |  | #8b008b |
| royalblue |  | #4169e0 | magenta |  | #ff00ff |
| ghostwhite |  | #f8f8ff | orchid |  | #da70d6 |
| lavender |  | #e6e6fa | mediumvioletred |  | #c71585 |
| midnightblue |  | #191970 | deeppink |  | #ff1493 |
| navy |  | #000080 | hotpink |  | #ff69b4 |
| darkblue |  | #00008b | lavenderblush |  | #fff0f5 |
| mediumblue |  | #0000cd | palevioletred |  | #db7093 |
| blue |  | #0000ff | crimson |  | #dc143c |
| darkslateblue |  | #483d8b | pink |  | #ffc0cb |
| slateblue |  | #6a5acd | lightpink |  | #ffb6c1 |

Спецсімвалы HTML

| Імя | Выгляд | Апісанне |
|----------|---------------|-----------------------|
| | | Неразрыўны прабел |
| € | € | Знак еўра |
| § | § | Параграф |
| ° | ° | Градус |
| ± | ± | Плюс-мінус |
| ¼ | $\frac{1}{4}$ | Дроб — адна чацвёртая |
| ½ | $\frac{1}{2}$ | Дроб — адна другая |
| ¾ | $\frac{3}{4}$ | Дроб — тры чацвёртыя |
| × | × | Знак множання |
| < | < | Знак «менш» |
| > | > | Знак «больш» |

Уласцівасці CSS для кіравання шрыфтамі

| Уласцівасць | Значэнне | Апісанне | Прыклад |
|--------------|--|--|--|
| font-family | імя шрыфта | Задае спіс шрыфтоў | P {font-family: Arial, serif} |
| font-style | normal italic oblique | Нармальны шрыфт Курсіў Нахілены шрыфт | P {font-style: italic} |
| font-variant | normal small-caps | Капітэль (асаблівыя вялікія літары) | P {font-variant: small-caps} |
| font-weight | normal lighter bold bolder 100–900 | Нармальная тлустасць Светлае напісанне Паўтлусты Тлусты 100 — светлы шрыфт, 900 — самы тлусты | P {font-weight: bold} |
| font-size | normal pt px % | Нармальны памер Пункты Пікселі Працэнты | font-size: normal font-size: 12pt font-size: 12px font-size: 120% |

Уласцівасці CSS для кіравання выглядам тэксту

| Уласцівасць | Значэнне | Апісанне | Прыклад |
|-----------------|--|---|---|
| line-height | normal множнік значэнне ў % | Міжрадковы інтэрвал | line-height: normal line-height: 1.5 line-height: 12px line-height: 120% |
| text-decoration | none underline overline line-through blink | Зняць усё афармленне Падкрэсліванне Лінія над тэкстам Перакрэсліванне Мігценне тэксту | text-decoration: none |
| text-transform | none capitalize uppercase lowercase | Зняць усё эфекты Пачынаць з Вялікіх УСЕ ВЯЛІКІЯ усе маленькія | text-transform: capitalize |
| text-align | left right center justify | Выраўноўванне тэксту | text-align: justify |
| text-indent | значэнне % | Водступ першага радка | text-indent: 15px; text-indent: 10% |

(Назва ўстановы адукацыі)

| Вучэбны год | Імя і прозвішча навучэнца | Стан вучэбнага дапаможніка пры атрыманні | Адзнака навучэнцу за карыстанне вучэбным дапаможнікам |
|-------------|---------------------------|--|---|
| 20 / | | | |
| 20 / | | | |
| 20 / | | | |
| 20 / | | | |
| 20 / | | | |
| 20 / | | | |

Вучэбнае выданне

Котаў Уладзімір Міхайлавіч
Лапо Анжаліка Іванаўна
Быкадораў Юрый Аляксандравіч
Вайцеховіч Алена Мікалаеўна

ІНФАРМАТЫКА

Вучэбны дапаможнік для 11 класа
ўстаноў агульнай сярэдняй адукацыі з беларускай мовай навучання
(з электроннымі дадаткамі)

Заг. рэдакцыі *Г. А. Бабаева*. Рэдактар *Д. І. Сімановіч*. Мастацкі рэдактар *А. А. Праваловіч*.
Мастакі *А. А. Шайчэнка*, *А. М. Багушэвіч*, *В. Ч. Праваловіч*. Тэхнічнае рэдагаванне
і камп'ютарная вёрстка *І. І. Дуброўскай*. Карэктары *В. С. Казіцкая*, *Г. В. Алешка*.

Падпісана да друку 30.07.2021. Фармат 70 × 90^{1/16}. Папера афсетная. Гарнітура школьная.
Друк афсетны. Умоўн. друк. арк. 8,19. Ул.-выд. арк. 8,0 + 20,0 эл. дадат. Тыраж 10 205 экз.
Заказ 1205.

Выдавецкае рэспубліканскае ўнітарнае прадпрыемства «Народная асвета» Міністэрства інфармацыі Рэспублікі Беларусь. Пасведчанне аб дзяржаўнай рэгістрацыі выдаўца, вытворцы, распаўсюджвальніка друкаваных выданняў № 1/2 ад 08.07.2013. Пр. Пераможцаў, 11, 220004, Мінск, Рэспубліка Беларусь.

Рэспубліканскае ўнітарнае прадпрыемства «Выдавецтва «Беларускі Дом друку». Пасведчанне аб дзяржаўнай рэгістрацыі выдаўца, вытворцы, распаўсюджвальніка друкаваных выданняў № 2/102 ад 01.04.2014. Пр. Незалежнасці, 79/1, 220013, Мінск, Рэспубліка Беларусь.

Правообладатель Народная асвета