

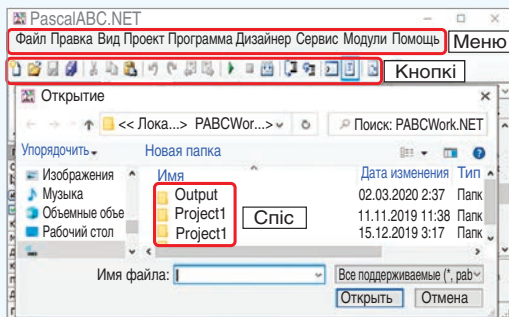
Глава 1 УВОДЗІНЫ Ў АБ'ЕКТНА-ПАДЗЕЙНАЕ ПРАГРАМАВАННЕ

§ 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).

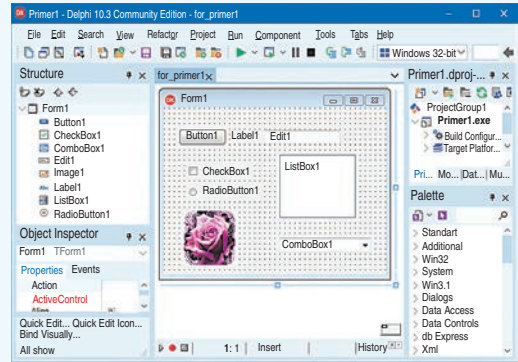
Праектаванне інтэрфейса акна праграмы можна выконваць з выкарыстаннем RAD-тэхналогіі (Rapid Application Development — **хуткая распрацоўка дадаткаў**). Тэхналогія RAD характэрная шмат для якіх сістэм праграмавання. Хуткая распрацоўка стала магчымай за кошт таго, што элементы кіравання былі візуалізаваны 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.

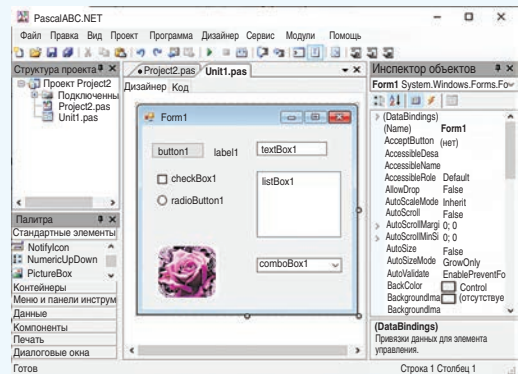
Візуальнае праграмаванне падтрымліваецца ў PascalABC і Delphi (код пішацца на мове Pascal), VisualBasic, C# і інш. (прыклад 1.2). Для навучання вучняў малодшых класаў выкарыстоўваецца візуальнае праграмаванне ў асяроддзі Скрэтч (Scratch).

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

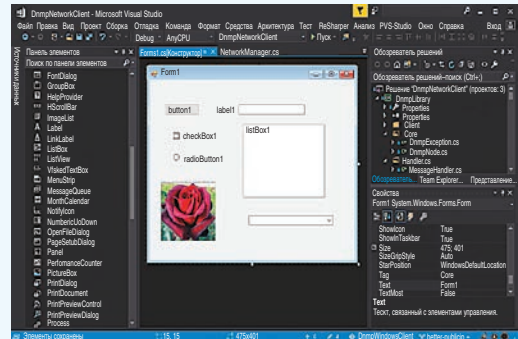
Delphi:



PascalABC:



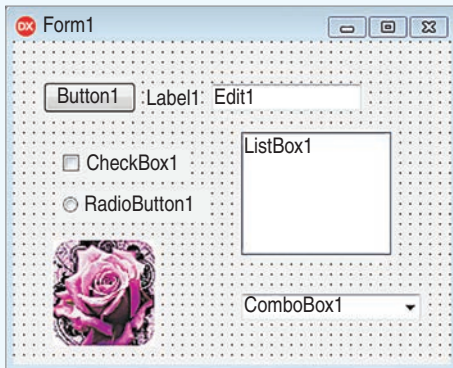
Visual Studio для мовы C#:



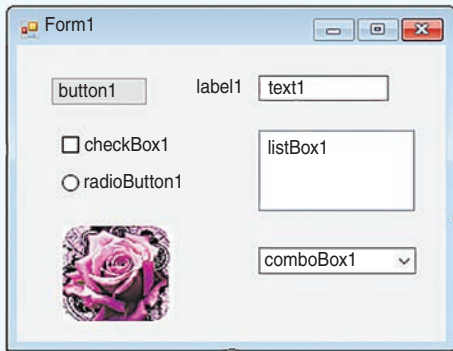
Прыклад 1.3. Асноўныя элементы інтэрфейса:

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 клавiшушы 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	Дваяная 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?



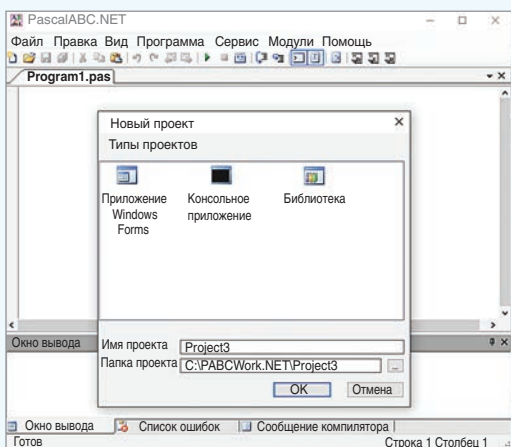
§ 2. Візуальнае асяроддзе распрацоўкі праграм

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.

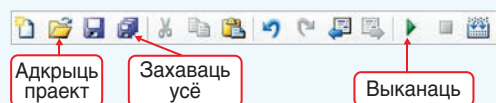
Інспектар аб'ектаў (прыклад 2.6) адлюстроўвае ўласцівасці (ці падзеі) выбранага аб'екта.

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.

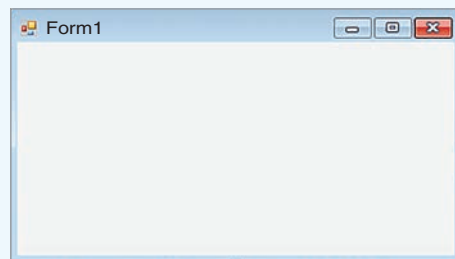
Укладка **События** змяшчае спіс падзей для аб'екта. Для кожнай падзеі можа быць вызначаны свой апрацоўшчык. Калі апрацоўшчык для падзеі вызначаны, насупраць падзеі будзе прапісана імя працэдуры (функцыі) апрацоўшчыка.

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

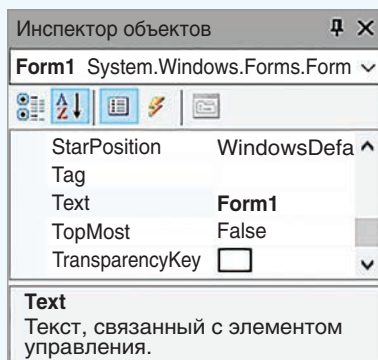
Прыклад 2.4. Меню і панэль хуткага доступу:



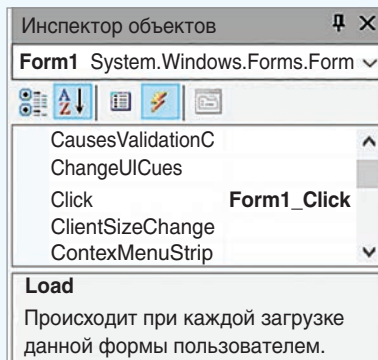
Прыклад 2.5. Форма:



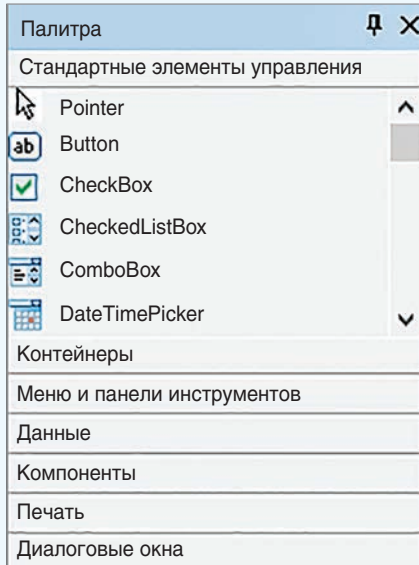
Прыклад 2.6. Інспектар аб'ектаў. Адлюстроўваюцца ўласцівасці формы:



Адлюстроўваюцца падзеі формы:



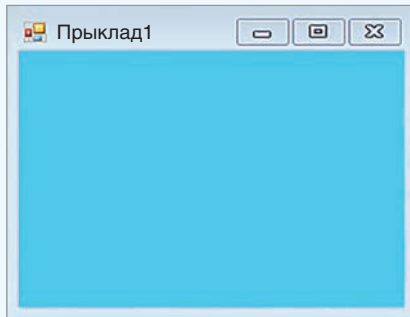
Прыклад 2.7. Палітра кампанентаў:



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: .

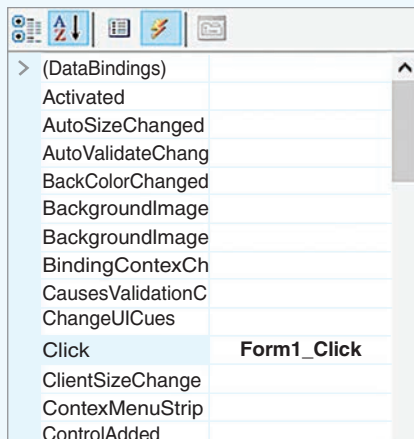
Пры стварэнні працэдуры-апрацоўшчыка ўласцівасці аб'ектаў можна змяняць праграма. Для гэтага трэба звярнуцца да ўласцівасці па яе імені і прысвоіць новае значэнне. Напрыклад, для змянення колеру формы трэба запісаць наступную каманду:

```
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.

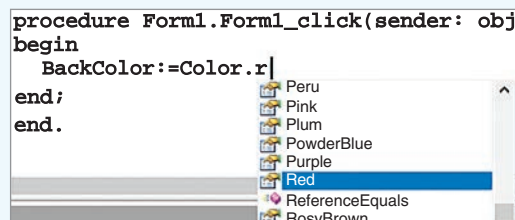
1. Выбар падзеі ў інспектары аб'ектаў:



2. Акно праграмнага кода з устаўленым апрацоўшчыкам:



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



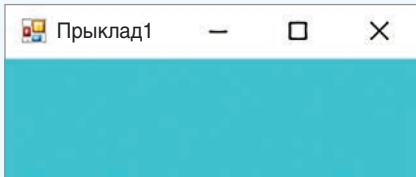
Калі вы ўведзяце першыя літары назвы ўласцівасці (метаду), то курсор перамясціцца ў спісе да тых ўласцівасцей і метадаў, назвы якіх пачынаюцца на гэтыя літары. Пасля гэтага патрэбную ўласцівасць можна ўставіць у праграму пстрычкай мышы ці націсканнем клявішы Enter. Калі спіс не з'явіўся, яго можна выклікаць 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. Устаноўце жоўты колер формы.
3. Стварыце апрацоўшчыкі для дзвюх падзей мышы: `MouseEnter` і `MouseLeave`.
4. У кодзе падзеі `MouseEnter` устаноўце 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.