

## Глава 3

### АСНОВЫ АЛГАРЫТМІЗАЦЫІ І ПРАГРАМІРАВАННЯ

#### § 13. Асноўныя алгарытмічныя канструкцыі

##### 13.1. Алгарытм і алгарытмічныя канструкцыі

У 7-м класе вы пазнаёміліся з асноўнымі алгарытмічнымі канструкцыямі. Для рашэння задач па праграміраванні былі вылучаны асноўныя этапы (прыклад 13.1).

**Алгарытм** — канечная паслядоўнасць дакладных дзеянняў, фармальнае выкананне якіх дазваляе атрымаць рашэнне задачы для любога дапушчальнага набору зыходных даных.

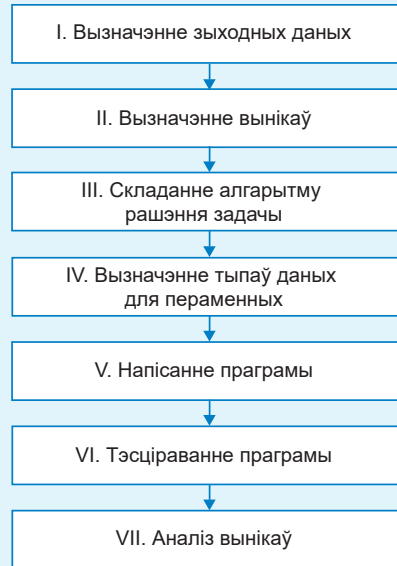
**Выканаўца** — чалавек, група людзей або тэхнічнае ўстройства, здольныя правільна выконваць каманды алгарытмаў. У далейшым будзем разглядаць толькі выканаўцу-ўстройства з абмежаваным наборам каманд. Набор каманд аднаго выканаўцы называюць **сістэмай каманд выканаўцы**. Каманды камп'ютарнага выканаўцы могуць быць рэалізаваны ў выглядзе працэдур і функцый.

Усе каманды выканаўцы дзеляць на групы:

1. Каманды, якія непасрэдна выконвае выканаўца.
2. Каманды, якія змяняюць парадак выканання іншых каманд выканаўцы.

Любы алгарытм можа быць запісаны з выкарыстаннем трох базавых алгарытмічных канструкцый:

Прыклад 13.1. Этапы рашэння задачы па праграміраванню:



У працэсе рашэння задачы некаторыя этапы даводзіцца паўтараць да таго часу, пакуль аналіз вынікаў не пакажа, што задача рэшана правільна.

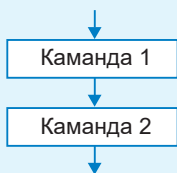
Для пошуку памылак можна выкарыстоўваць сродкі наладкі праграм (гл. *Дадатак 3*, с. 161—162).

У 1966 г. італьянскія матэматыкі Карада Бём (1923—2017) і Джузэпэ Джакапіні (1936—2001) сфармулявалі і даказалі становішча структурнага праграміравання, згодна з якім любы выконваемы алгарытм можа быць пераўтвораны да структурнага выгляду, калі ход выканання алгарытму вызначаецца пры дапамозе трох структур кіравання: паслядоўнасцей, галінаванняў і цыклаў.

Цалкам канцэпцыя структурнага праграмавання была распрацавана ў сярэдзіне 70-х гг. пры ўдзеле Э. Дэйкстры.

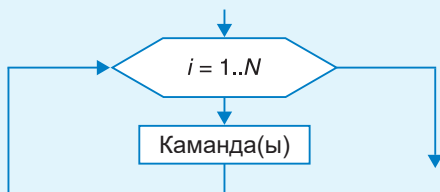
**Прыклад 13.2** Блок-схемы алгарытмічных канструкцый:

1. Паслядоўнасць:

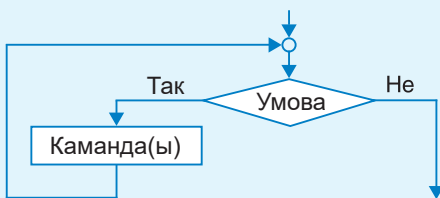


2. Цыкл:

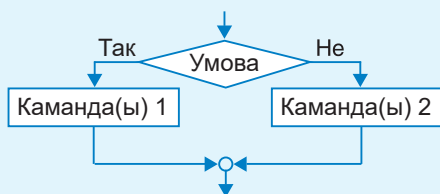
1) цыкл з параметрам (значэнне параметра змяняецца ад 1 да  $N$ )



2) цыкл з перадумовай



3. Каманда галінавання



**паслядоўнасць, цыкл і галінаванне** (прыклад 13.2).

Каманды цыкла і галінавання кіруюць парадкам выканання іншых каманд у праграме і належаць да **каманд кіравання (кіруючым канструкцыям)**.

Паслядоўнасць каманд, выканаўцам якой з'яўляецца камп'ютар, называецца **праграмай**. Праграма ўяўляе сабой запіс на некаторай фармальнай мове — мове праграмавання. Камандамі ў мове праграмавання лічаць:

- аператары (аператар прысвойвання, аператар галінавання, аператар цыкла і інш.);
- выклікі дапаможных алгарытмаў (як убудаваных у бібліятэкі, так і створаных карыстальнікам).

### 13.2. Алгарытмічная канструкцыя паслядоўнасць

Алгарытмічная канструкцыя **паслядоўнасць** — паслядоўнасць каманд алгарытму, што выконваюцца ў тым парадку, у якім яны запісаны. Сярод каманд, якія ўтвараюць алгарытмічную канструкцыю **паслядоўнасць**, адсутнічаюць каманды, што змяняюць парадак выканання іншых каманд.

У 7-м класе, вывучаючы мову Pascal, вы выкарыстоўвалі наступныя каманды (прыклад 13.3):

- працэдуры для ўводу і вываду даных;
- аператар прысвойвання.

Для ўводу даных прызначана каманда `read()`. У дужках праз коску пералічваюцца імёны пераменных, значэнні якіх неабходна ўвесці.

Для **вываду даных** выкарыстоўваюць каманду `write()`. Яна дазваляе выводзіць тэкставыя паведамленні і лікавыя значэнні. Тэкставыя паведамленні запісваюцца ў двукоссі, выводзяцца ў выглядзе паслядоўнасці сімвалаў так, як запісаны, і не аналізуюцца пры выкананні.

Пры выкарыстанні каманды `writeln()`; пасля вываду паведамлення або ліку адбываецца перавод курсора на наступны радок.

**Аператар прысвойвання** прызначаны для таго, каб:

- задаваць значэнні пераменным;
- вылічаць значэнне выразу (вынік будзе запісаны як значэнне пераменнай).

Фармат запісу аператара прысвойвання:

```
<імя пераменнай> := <выраз>;
```

У запісе арыфметычнага выразу выкарыстоўваюцца знакі матэматычных дзеянняў: складання (+), аднімання (-), множання (\*), дзялення (/), а таксама цэлалікавага дзялення (`div`) і знаходжання астачы (`mod`). Неабходна памятаць, што аперацыя дзялення (/) выкарыстоўваецца пры вылічэннях з данымі тыпу `real`. Для даных тыпу `integer` выкарыстоўваюцца аперацыі `div` і `mod`.

Нярэдка ў адной праграме даводзіцца выконваць адну і тую ж паслядоўнасць каманд некалькі разоў.

**Прыклад 13.3.** Дадзены  $x, y$ . Напісаць праграму для вылічэння значэння выразу

$$a = \frac{2x}{7 + y^2}(x - y).$$

Этапы выканання задання

I. Вызначэнне зыходных даных: пераменныя  $x, y$ .

II. Вызначэнне вынікаў: пераменная  $a$ .

III. Алгарытм рашэння задачы.

1. Увод зыходных даных.

2. Вылічэнне значэння выразу.

3. Вывад выніку.

IV. Апісанне пераменных: усе пераменныя, вызначаныя для рашэння задачы, маюць тып `real`.

V. Праграма:

```
var x,y,a: real;
```

```
begin
```

```
  write('Увядзіце x =');
```

```
  read(x);
```

```
  write('Увядзіце y =');
```

```
  read(y);
```

```
  a:= 2 * x * (x-y)/(7 + y * y);
```

```
  writeln('a =',a);
```

```
end.
```

VI. Тэсціраванне праграмы:

Запусціць праграму і ўвесці значэнні  $x = 3.8, y = 2.7$ . Вынік:

Окно вывода

```
Увядзіце x = 3.8
```

```
Увядзіце y = 2.7
```

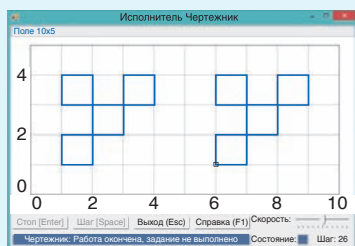
```
a = 0.585024492652204
```

VII. Правільнасць вылучэнняў праверыць на калькулятары.

У прыкладзе алгарытмічная канструкцыя *паслядоўнасць* утворана камандамі:

- вывад паведамленняў;
- увод значэнняў пераменных;
- каманда прысвойвання;
- вывад выніку.

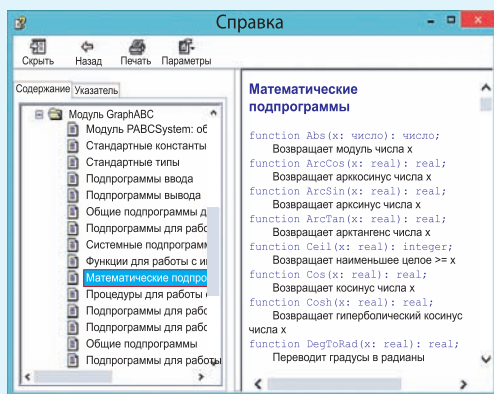
**Прыклад 13.4.** Напісаць праграму для вываду наступнага відарыса:



Відарыс складаецца з двух адволькавых фігур. Аформім дапаможны алгарытм `Figura` для адлюстравання адной фігуры. Праграма:

```
uses Drawman;
procedure Figura;
begin
  PenDown; OnVector(1, 0);
  OnVector(0, 3); OnVector(-1, 0);
  OnVector(0, -1); OnVector(3, 0);
  OnVector(0, 1); OnVector(-1, 0);
  OnVector(0, -2); OnVector(-2, 0);
  OnVector(0, -1); PenUp;
end;
begin
  Field(10, 5);
  ToPoint(1, 1); Figura;
  ToPoint(6, 1); Figura;
end.
```

**Прыклад 13.5.** Пералік матэматычных функцый можна паглядзець у даведцы `PascalABC`:



У гэтым выпадку зручна выкарыстоўваць дапаможны алгарытм, які можна выконваць патрэбную колькасць разоў, звяртаючыся да яго назвы.

**Дапаможны алгарытм** — алгарытм, які можна выкарыстоўваць у іншых алгарытмах, запісаўшы яго імя і, калі неабходна, значэнні параметраў.

Дапаможны алгарытм вырашае некаторую частку асноўнай задачы. Выклік дапаможнага алгарытму з’яўляецца камандай, якая можа замяняць некалькі каманд.

Дапаможныя алгарытмы вы выкарыстоўвалі пры напісанні праграм для вучэбных камп’ютарных выканаўцаў Чарцэжнік і Робат (прыклад 13.4). Каманды `read` і `write` таксама рэалізаваны як дапаможныя алгарытмы.

Пры вылічэннях часта выкарыстоўваюцца розныя матэматычныя функцыі (прыклад 13.5). Гэтыя функцыі рэалізаваны як убудаваныя дапаможныя алгарытмы і могуць ужывацца пры запісе арыфметычных выразаў. Аргументы функцый заўсёды запісваюцца ў дужках. Некаторыя з функцый прыведзены ў табліцы (іншыя можна паглядзець у *Дадатку 3*, с. 158).

Запіс на мове Pascal	Апісанне
<code>abs(x)</code>	Знаходзіць модуль ліку $x$
<code>sqr(x)</code>	Узводзіць лік $x$ у квадрат
<code>sqrt(x)</code>	Знаходзіць карань квадратнага ліку $x$ . Вынік — заўсёды лік тыпу <code>real</code>

Запіс на мове Pascal	Апісанне
<code>trunc(x)</code>	Знаходзіць цэлую частку рэчаіснага ліку $x$ (real). Вынік — лік тыпу <code>integer</code>
<code>frac(x)</code>	Знаходзіць дробавую частку рэчаіснага ліку $x$ (real). Вынік — лік тыпу <code>real</code>
<code>sin(x)</code>	Вылічае сінус ліку $x$ . Лік $x$ задаецца ў радыянах <sup>1</sup>
<code>cos(x)</code>	Вылічае косінус ліку $x$ . Лік $x$ задаецца ў радыянах
<code>RadToDeg(x)</code>	Пераводзіць радыяны ў градусы
<code>DegToRad(x)</code>	Пераводзіць градусы ў радыяны

Аргументам функцыі можа быць лік, пераменная, выраз або іншая функцыя: `sin(DegToRad(45))`, `sqrt(abs(-16))`.

У прыкладзе 13.6 выкарыстоўваюцца матэматычныя функцыі для ўзвядзення ліку ў квадрат і вылічэння квадратнага караня.



1. Што такое алгарытм?
2. Пералічыце асноўныя алгарытмічныя канструкцыі.
3. Якая каманда выкарыстоўваецца ў мове Pascal для ўводу даных?
4. Якія каманды выкарыстоўваюцца ў мове Pascal для вываду даных?
5. Для чаго патрэбна каманда прысвойвання?
6. У якіх выпадках зручна выкарыстоўваць дапаможны алгарытм?
7. Якія матэматычныя функцыі могуць выкарыстоўвацца пры запісе арыфметычных выразаў?

<sup>1</sup> Радыян — адзінка вымярэння вуглоў у Міжнароднай сістэме адзінак (π радыян адпавядае велічыні разгорнутага вугла 180°).

**Прыклад 13.6.** Зададзена даўжыня стараны квадрата  $a$ . Напісаць праграму знаходжання плошчы квадрата і даўжыні яго дыяганалі.

Этапы выканання задання

I. Зыходныя даныя: даўжыня стараны, пераменная  $a$ .

II. Вынік: пераменныя  $S$  (плошча) і  $d$  (даўжыня дыяганалі).

III. Алгарытм рашэння задачы.

1. Увод зыходных даных.

2. Вылічэнне плошчы па формуле  $S = a^2$ .

3. Вылічэнне даўжыні дыяганалі па формуле  $d = a\sqrt{2}$ .

4. Вывад выніку.

IV. Апісанне пераменных: усе пераменныя, вызначаныя для рашэння задачы, маюць тып `real`.

V. Праграма:

```
var a, S, d: real;
```

```
begin
```

```
  write('Увядзіце a = '); read(a);
```

```
  S:= sqr(a); d:= a*sqrt(2);
```

```
  writeln('S =',s); writeln('d =',d);
```

```
end.
```

VI. Тэсціраванне праграмы.

Запусціць праграму і ўвесці значэнне  $a = 5.6$ . Вынік:

Окно вывода

```
Увядзіце a = 5.6
S = 31.36
d = 7.91959594928933
```

Правільнасць вылічэнняў можна праверыць на калькулятары.



## Практыкаванні

1 Расстаўце каманды праграмы ў правільным парадку так, каб можна было вылічыць значэнне выразу  $a = \frac{2x}{x^2 + 4}$ .

1. `writeln('a = ', a);`

2. `write('Увядзіце значэнне x =');`

3. **End.**

4. **Var** x, y, a: real;

5. **Begin**

6. `a := 2 * x / (x * x + 4);`

7. `read(x);`

2 Вызначыце тыпы даных для кожнай пераменнай, выкарыстанай у апэратары прысвойвання.

1. `y := sqrt(a - 4) / 16;`

2. `z := sqr(3 * a + 2);`

3. `a := abs(a - 4.2);`

4. `d := x mod 2;`

5. `y := int(a);`

6. `y := trunc(a);`

7. `y := frac(a);`

8. `s := sin(3.14 * r).`

3 Знайдзіце і выпраўце памылкі ў праграмах.

1. **var** x, y, z1, z2:integer;

**begin**

`write('Увядзіце x =');`

`read(x);`

`write('Увядзіце y =');`

`read(y);`

`z1 := int(x/y);`

`z2 := frac(x/y);`

`write('Цэлая частка =', z1);`

`write('Дробная частка =', z2);`

**end.**

2. **var** x, y, z1, z2:real;

**begin**

`write('Увядзіце x =');`

`read(x);`

`write('Увядзіце y =');`

`read(y);`

`z1 := x div y;`

`z2 := x mod y;`

`write('Цэлая частка =', z1);`

`write('Астача =', z2);`

**end.**

4 Дадзены x і z. Змяніце праграму з прыкладу 13.4 так, каб вылічалася значэнне выразу  $a = \frac{2x}{\sqrt{z^2 + 9}}$ .

5 Зададзены тры лікі. Напішыце праграму для знаходжання сярэдняга арыфметычнага гэтых лікаў.

6 Зададзены два лікі. Напішыце праграму для знаходжання дзелі ад дзялення першага ліку на другі і акругліце вынік да найбліжэйшага цэлага.

7 Дадзены гіпатэнуза і катэт прамавугольнага трохвугольніка. Напішыце праграму для знаходжання другога катэта і плошчы трохвугольніка.

8\* Зададзены два лікі. Напішыце праграму для знаходжання дзелі гэтых лікаў. Акругліце вынік да дзясятых, пакінуўшы ў дробавай частцы адну лічбу.

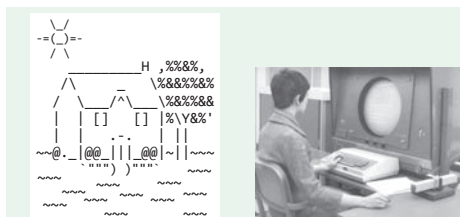
## § 14. Графічныя магчымасці асяроддзя праграмавання PascalABC

### 14.1. Асновы работы з графікай

Вы ўжо знаёмыя з графічнымі рэдактарамі, у якіх для пабудовы відарысаў на камп'ютары выкарыстоўваюцца графічныя прымітывы — простыя геаметрычныя фігуры: прамавугольнік, акружнасць, эліпс, адрэзак і г. д. Графічны рэдактар — праграма, напісаная на якой-небудзь мове праграмавання.

Для работы з графікай мовы праграмавання выкарыстоўваюць спецыяльныя бібліятэкі (модулі), якія змяшчаюць наборы каманд для пабудовы відарысаў. У PascalABC для работы з графікай выкарыстоўваецца бібліятэка GraphABC. Для падключэння гэтай бібліятэкі ў праграме запісваецца каманда `uses GraphABC;`

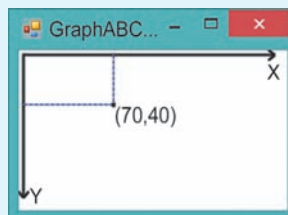
Становішча фігур задаецца каардынатамі ў графічным акне. Каардынатная плоскасць у ім адрозніваецца ад той, якую вы выкарыстоўваеце на ўроках матэматыкі. Пачаткам каардынат з'яўляецца верхні левы вугал графічнага акна — пункт  $(0; 0)$  (прыклад 14.1). Каардынаты задаюць парадкавы нумар пікселя па гарызанталі і па вертыкалі, таму яны могуць быць толькі цэлымі лікамі. Адлік значэнняў каардынат  $x$  адбываецца злева направа, а каардынаты  $y$  — зверху ўніз. Па змоўчанні ствараецца графічнае акно памерам  $640 \times 480$  пікселяў.



Першыя камп'ютары не мелі магчымасцей работы з графікай. На друкавальных устатках выводзіліся «карцінкі», што складаліся з сімвалаў.

У 1958 г. быў запушчаны камп'ютар Lincoln TX-2, у якім упершыню выкарыстоўваўся графічны экран. У 1981 г. пачалі выкарыстоўваць колеры. У графічным рэжыме пры разрэшэнні  $320 \times 200$  пікселяў выкарыстоўваліся 4 колеры са стандартных палітр: пурпурны, сіне-зялёны, белы, чорны або чырвоны, зялёны, жоўты, чорны.

**Прыклад 14.1.** Графічнае акно асяроддзя праграмавання PascalABC з адлюстраваннем каардынатных восяў і пункта з каардынатамі  $(70; 40)$ .

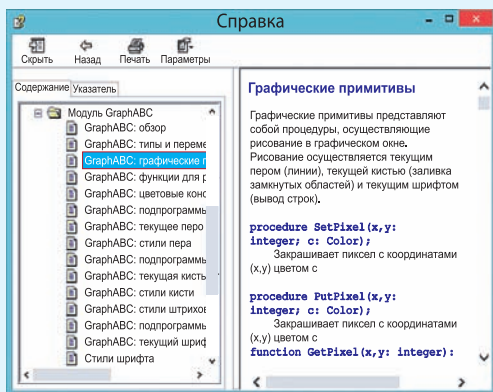


Пункт размешчаны на адлегласці 70 пікселяў ад левага краю акна і на адлегласці 40 пікселяў ад верхняга краю.

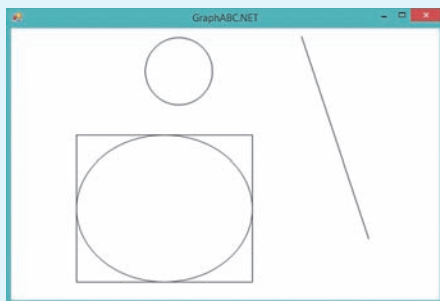
Памеры графічнага акна можна задаць камандай `SetWindowSize(n,m);` У дужках пададзены памеры акна па гарызанталі і па вертыкалі.

<sup>1</sup> <https://www.asciiart.eu/buildings-and-places/houses> (дата доступу: 26.07.2018).

**Прыклад 14.2.** Работа з даведчай сістэмай. Для пераходу ў даведнік неабходна націснуць клавішу F1 або выканаць каманду меню: **Помощь** → → **Справка**. У акне, якое адкрыецца, перайці ў раздзел **Стандартные модули** і выбраць **Модуль GraphABC**.



**Прыклад 14.3.** Графічныя прымітывы:



Праграма для іх рысавання:

```
uses GraphABC;
begin
    //Круг
    Circle(250, 125, 30);
    //Прамавугольнік
    Rectangle(100,200,400,450);
    //Эліпс
    Ellipse(100,200,400,450);
    //Адрэзак
    Line(450, 50, 550, 350);
end.
```

## 14.2. Работа з даведчай сістэмай асяроддзя праграмавання PascalABC

У бібліятэцы GraphABC змяшчаецца вялікая колькасць каманд. Гэтыя каманды апісаны ў даведчай сістэме асяроддзя PascalABC (прыклад 14.2). Тут ёсць апісанне графічных прымітываў, назвы колеравых канстант, апісанне работы з пярэм і пэндзлем, каманды работы з графічным акном.

Каманды бібліятэкі GraphABC — дапаможныя алгарытмы, запісаныя як асобныя працэдуры. Выкарыстанне каманды ў праграме азначае выклік адпаведнага алгарытму.

## 14.3. Асноўныя графічныя прымітывы

Разгледзім каманды для рысавання графічных прымітываў:

- $\text{Line}(x_1, y_1, x_2, y_2)$  — адрэзак, які злучае пункты з каардынатамі  $(x_1, y_1)$  і  $(x_2, y_2)$ ;
- $\text{MoveTo}(x, y)$  — устанаўлівае бягучую пазіцыю рысавання ў пункт  $(x, y)$ ;
- $\text{LineTo}(x, y)$  — адрэзак ад бягучай пазіцыі да пункта  $(x, y)$ ;
- $\text{Rectangle}(x_1, y_1, x_2, y_2)$  — прамавугольнік, зададзены каардынатамі супрацьлеглых вяршынь  $(x_1, y_1)$  і  $(x_2, y_2)$ ;
- $\text{Circle}(x_1, y_1, r)$  — круг з цэнтрам у пункце  $(x_1, y_1)$  і радыусам  $r$ ;
- $\text{Ellipse}(x_1, y_1, x_2, y_2)$  — авал (эліпс), упісаны ў прамавугольнік з каардынатамі супрацьлеглых вяршынь  $(x_1, y_1)$  і  $(x_2, y_2)$ .

(Разгледзьце прыклад 14.3.)

Каманды для рысавання іншых графічных прымітываў маюцца ў даведчай сістэме і ў *Дадатку 3* (с. 159).

**Прыклад 14.4.** Напісаць праграму, якая будзе відарыс доміка, вы-

карыстоўваючы працэдуры Line, Lineto, Rectangle, Circle.

Этапы выканання задання

I. Зыходныя даныя: вынік работы праграмы не залежыць ад зыходных даных.

II. Вынік: гатовы рысунак.

III. Алгарытм рашэння задачы.

Рысунак складаецца з: прамавугольнікаў, адрэзкаў, круга. Для разліку каардынат рэкамендуецца спачатку зрабіць рысунак на лісце паперы ў клетачку.

IV. Апісанне пераменных. Пераменныя не выкарыстоўваюцца.

#### 14.4. Работа з пярэм і пэндзлем

У графічных рэдактарах, перш чым рысаваць якія-небудзь фігуры, устаўляюць іх колер. Звычайна выбіраюць два колеры. Колер 1 вызначае колер ліній і контураў фігур, Колер 2 выкарыстоўваецца для заліўкі фігур. Акрамя таго, можна змяняць стыль ліній і заліўкі, а таксама вызначаць таўшчыню ліній.

У графічным рэжыме PascalABC настройкі ліній вызначае пярэ (Pen), а настройкі ўнутранай вобласці фігур — пэндзаль (Brush). Каманды для работы з пэндзлем і пярэм прыведзены ў табліцы.

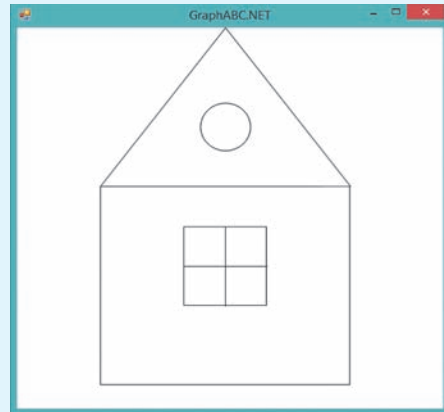
Каманда	Апісанне
SetPenColor	Колер ліній
SetPenWidth	Таўшчыня ліній
SetPenStyle	Стыль ліній
SetBrushColor	Колер заліўкі
SetBrushStyle	Стыль заліўкі
SetBrushHatch	Від штрыхоўкі для заліўкі

#### Прыклад 14.4.

V. Праграма:

```
uses GraphABC;
begin
  //Дом
  Rectangle(100,200,400,450);
  //Акно
  Rectangle(200,250,300,350);
  Line(250, 250, 250, 350);
  Line(200, 300, 300, 300);
  //Дах
  MoveTo(100,200);
  LineTo(250, 0);
  LineTo(400, 200);
  Circle(250, 125, 30);
end.
```

VI. Тэсціраванне праграмы: Запусіць праграму. Вынік:



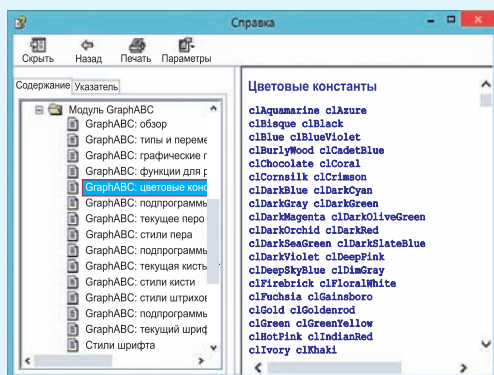
Звярніце ўвагу, што пры напісанні каманд у вас з'яўляюцца падказкі:

```
Circle(
  procedure GraphABC.Circle(x: integer; y: integer; r: integer);
  Описание:
  Рисует заполненную окружность с центром (x,y) и радиусом r
```

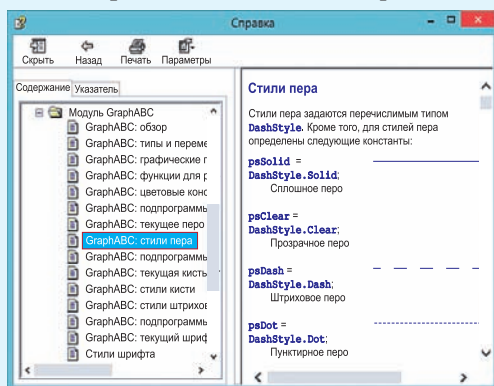
Падказка з'яўляецца таксама пры наведзенні мышы на ўжо напісаную каманду.

Калі ўстанавіць тэкставы курсор на каманду і націснуць F1, то адкрыецца старонка з даведніка з апісаннем гэтай каманды.

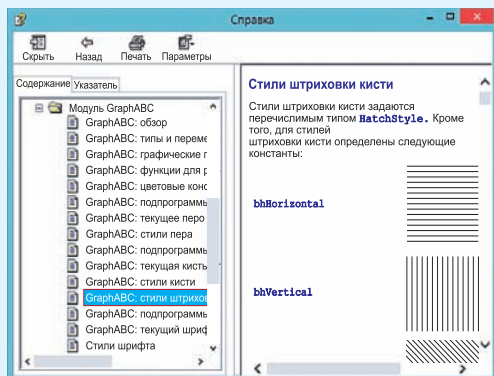
Прыклад 14.5. Колеравыя канстанты.



Прыклад 14.6. Стылі пярэ.



Прыклад 14.7. Стылі штрыхоўкі пэндзля.



Значэнне, якое трэба ўстанавіць для кожнай з каманд, запісваецца ў дужках. Напрыклад:

- `SetPenColor(clRed)` — чырвоны колер рысавання ліній;
- `SetBrushColor(clBlue)` — сіні колер заліўкі фігур;
- `SetPenWidth(3)` — таўшчыня пярэ ў 3 пікселі;
- `SetPenStyle(psDot)` — штрыховаая лінія;
- `SetBrushStyle(bsHatch)` — штрыховаая заліўка;
- `SetBrushHatch(bhCross)` — штрыхоўка ў клетачку.

Значэнні колеравых канстант, стыляў ліній і залівак можна знайсці ў даведкавай сістэме асяроддзя PascalABC (прыклады 14.5—14.7) і ў *Дадатку 3* (с. 160).

Каманды для ўстаноўкі колеру і стылю запісваюць перад камандай рысавання фігуры. Гэтыя каманды дзейнічаюць да таго часу, пакуль колер або стыль не будзе зменены. Калі, напрыклад, для пярэ была ўстаноўлена таўшчыня ў 3 пікселі, то адрэзкі і межы фігур будуць мець таўшчыню ў тры пікселі да новай змены таўшчыні інструмента.

Для фігур па змоўчанні ўстаноўлена заліўка белым колерам. Калі колер пэндзля выбраць да рысавання фігуры, то фігура будзе зафарбавана ўстаноўленым колерам. Колер ужо нарысаванай фігуры можна змяніць з дапамогай каманды заліўкі:

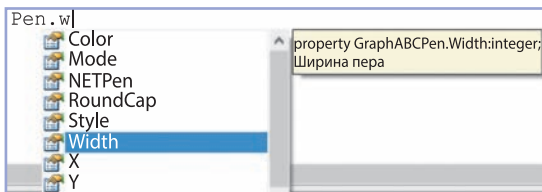
`FloodFill(x,y,c);` – залівае абмежаваную вобласць аднаго колеру колерам `c`, пачынаючы з пункта ўнутры вобла-

сці  $(x,y)$  (у прыкладзе 14.8 паказана, як расфарбаваны домік з прыкладу 14.4).

Акрамя таго, асяроддзе праграмавання PascalABC дазваляе звяртацца да ўласцівасцей пэндзля і пяра па-іншаму. Так, напрыклад, для змянення колеру (стылю, таўшчыні) можна запісаць

```
Pen.Color := clRed;
Pen.Style := psDot;
Pen.Width := 3;
```

Падказка асяроддзя выглядае наступным чынам:



Другую частку каманды можна выбраць з выпадаючага спіса.

Пры вывучэнні вектарнай графікі вы пазнаёміліся з колеравай мадэллю RGB, якая дазваляе запісаць любы колер трыма складнікамі: чырвоным, зялёным і сінім. Функцыя  $RGB(r, g, b)$  дазваляе вызначыць колер па трох складніках. Так, каманда `SetPenColor(clRed);` аналагічная камандзе `SetPenColor( RGB(255,0,0) );` Такі спосаб задання колеру дазваляе задаваць колеры, значэнні якіх не апісаны колеравымі канстантамі. Значэнні складнікаў колеру можна паглядзець у графічным рэдактары Paint (прыклад 14.9).

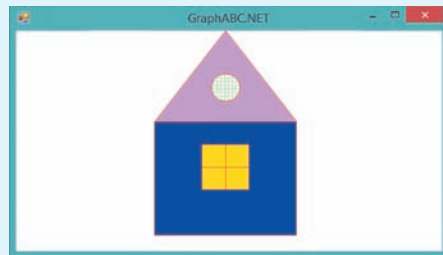
У графічным рэжыме PascalABC можна выводзіць у графічнае акно тэксты і лікі.

`TextOut(x,y,z);` — выводзіць радок або лік  $z$  у прамавугольнік з каардынатамі левага верхняга вугла  $(x,y)$ .

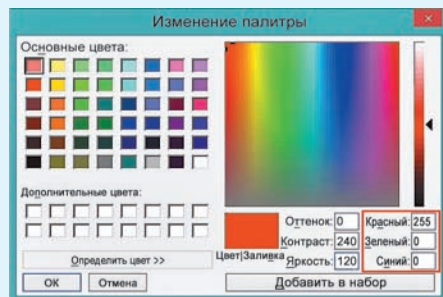
**Прыклад 14.8.** Праграма:

```
uses GraphABC;
begin
  //Дом
  SetPenColor( RGB(255,0,0) );
  SetBrushColor( clBlue );
  Rectangle(100,200,400,450);
  //Акно
  SetBrushColor( clYellow );
  Rectangle(200,250,300,350);
  SetPenColor( clRed );
  SetPenStyle( psDot );
  SetPenWidth( 2 );
  Line(250,250,250,350);
  Line(200,300,300,300);
  //Дах
  SetPenStyle( psSolid );
  SetPenWidth( 1 );
  Line(100, 200, 250, 0);
  Line(250, 0, 400, 200);
  SetBrushStyle( bsHatch );
  SetBrushColor( clLightGreen );
  SetBrushHatch( bhCross );
  Circle(250, 125, 30);
  FloodFill(250,70, clPlum);
end.
```

Вынік работы праграмы:



**Прыклад 14.9.** Складнікі колеру:



**Прыклад 14.10.** Выведзем у графічнае акно прывітанне свету, выкарыстоўваючы розныя ўласцівасці тэксту.

Праграма:

```
uses GraphABC;
begin
  //Колер фону для тэксту
  SetBrushColor(clYellow);
  SetFontName('Comic Sans MS');
  //Колер літар
  SetFontColor(clBlue);
  //Памер шрыфту
  SetFontSize(25);
  //Паўтлусты шрыфт
  SetFontStyle(fsBold);
  TextOut(20,30,'Прывітанне,');
  SetBrushColor(clPink);
  SetFontName('PaladiumC
    Bold Italic');
  SetFontColor(clSalmon);
  SetFontSize(50);
  SetFontStyle(fsUnderline);
  TextOut(120,130,'СВЕТ!');
  //Празрысты фон
  SetBrushStyle(bsClear);
  SetFontName('Tahoma');
  SetFontColor(clViolet);
  SetFontSize(20);
  SetFontStyle(fsItalic);
  TextOut(300,130,'PascalABC');
end.
```

Вынік выканання праграмы:



Калі трэба вывесці радок, то яго сімвалы змяшчаюць у двукоссе, для вываду ліку можна выкарыстоўваць пераменныя або значэнні лікаў. Калі ў якасці  $z$  запісаць арыфметычны выраз, то будзе выведзена яго значэнне.

Для змянення параметраў тэксту ўжываюць наступныя каманды:

Каманда	Апісанне
SetFontColor	Колер сімвалаў
SetFontSize	Памер сімвалаў
SetFontName	Імя бягучага шрыфту
SetFontStyle	Стыль тэксту

Каманда SetBrushColor устанаўлівае колер прамавугольніка, унутры якога будзе знаходзіцца тэкст (прыклад 14.10). Каманда дзейнічае да таго часу, пакуль колер не будзе зменены. Для запісу тэксту без фону трэба ўстанавіць празрысты фон пэндзля: SetBrushStyle(bsClear).

Усе параметры для тэксту задаюцца да каманды вываду яго ў графічнае акно. Імя шрыфта, якім будзе выведзены тэкст, змяшчаюць у двукоссе. Магчымыя варыянты можна паглядзець у Word. Стыль тэксту можа мець наступныя значэнні: fsNormal (звычайны), fsBold (тлусты), fsItalic (нахільны), fsUnderline (падкрэслены) або іх камбінацыі: fsBoldUnderline (тлусты падкрэслены).

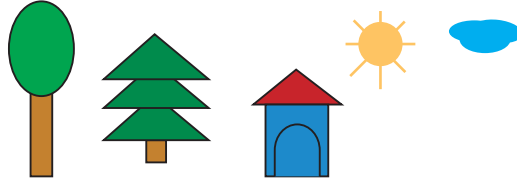


1. Якая бібліятэка выкарыстоўваецца для падключэння графікі ў PascalABC?
2. Як вызначана сістэма каардынат у графічным акне?
3. Як задаць памеры графічнага акна?
4. Як знайсці апісанне графічных прымітываў у даведнік?
5. Як можна змяніць колер ліній, заліўкі?
6. Як вывесці тэкст у графічным акне?



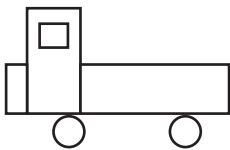
### Практыкаванні

- 1 Падпішыце відарыс з прыкладу 14.8.
- 2 Дапоўніце відарыс доміка з прыкладу 14.8 відарысамі трубы і дыму з трубы ў выглядзе некалькіх авалаў:
- 3 Дапоўніце вынік, атрыманы пры выкананні задання 2, якімі-небудзь з прапанаваных відарысаў або прыдумайце свае.

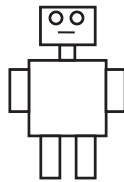


- 4 Напішыце праграму для стварэння відарыса. Размалюйце дадзены відарыс па сваім вырашэнні. Дадатковыя каманды для пабудовы графічных прымітываў можна знайсці ў даведачнай сістэме.

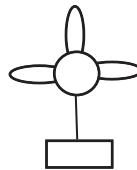
Грузавік  
1



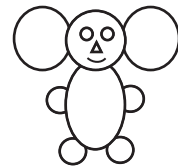
Робат  
2



Кветка  
3



Чабурашка  
4



## § 15. Простыя і састаўныя ўмовы

### 15.1. Лагічны тып даных

Нагадаем вывучаныя ў 7-м класе паняцці *выказванне* і *ўмова для выканаўцы*.

**Выказванне** — апавядальны сказ (сцверджанне), пра які можна сказаць, праўдзiвы ён ці непраўдзiвы.

**Умовай для выканаўцы** з'яўляецца вядомае яму выказванне, якое можа выконвацца (быць праўдзiвым) або не выконвацца (быць непраўдзiвым).

Тып Boolean названы ў гонар англійскага матэматыка і логіка Джорджа Буля, які займаўся пытаннямі матэматычнай логікі ў XIX ст.

Дадзены тып прысутнічае ў пераважнай большасці моў праграмавання. У некаторых мовах рэалізуецца праз лікавы тып даных. Тады за значэнне «няпраўда» прымаецца 0, а за значэнне «праўда» — 1.